Batching of Tasks by Users of Pseudonymous Forums: Anonymity Compromise and Protection

Anonymous Author(s)
Affiliation
Address
`email`

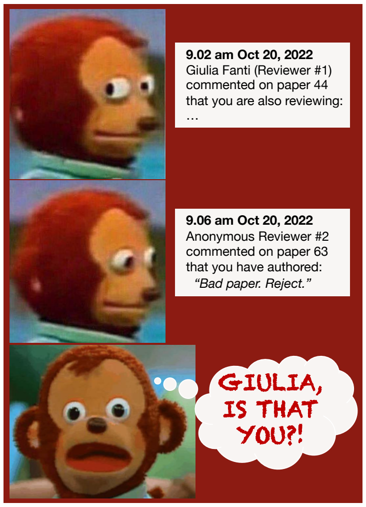
Abstract

There are a number of forums where people participate under pseudonyms. One example is peer review, where the identity of reviewers for any paper is confidential. When participating in these forums, people frequently engage in "batching": executing multiple related tasks (e.g., commenting on multiple papers) at nearly the same time. Our empirical analysis shows that batching is common in two applications we consider – peer review and Wikipedia edits. In this paper, we identify and address the risk of deanonymization arising from linking batched tasks. To protect against linkage attacks, we take the approach of adding delay to the posting time of batched tasks. We first show that under some natural assumptions, no delay mechanism can provide a meaningful differential privacy guarantee. We therefore propose a "one-sided" formulation of differential privacy for protecting against linkage attacks. We design a mechanism that adds zero-inflated uniform delay to events and show it can preserve privacy. We prove that this noise distribution is in fact optimal in minimizing expected delay among mechanisms adding independent noise to each event, thereby establishing the Pareto frontier of the trade-off between the expected delay for batched and unbatched events. Finally, we conduct a series of experiments on Wikipedia and Bitcoin data that corroborate the practical utility of our algorithm in obfuscating batching without introducing onerous delay to a system.

1 Introduction

In a number of applications where anonymity is critical, users act under pseudonyms to preserve their privacy. For instance, in scientific peer review using online forums like OpenReview.net, reviewers make comments on papers that are publicly viewable. Reviewers (and meta-reviewers) who have been assigned multiple papers operate under different pseudonyms across their papers to remain anonymous. Other examples of publicly visible tasks where users operate under pseudonyms include Wikipedia editing and cryptocurrency transactions.

In many settings, it is common for users to engage in *batching* — the completion of several similar tasks at the same time. Batching occurs both due to natural bursts in activity (e.g., a person visits a website and makes many comments at once) or as a productivity strategy used to streamline work. Indeed, both academic studies [23, 29, 4] and popular media [32, 33, 24] rec-



9.02 am Oct 20, 2022
Giulia Fanti (Reviewer #1)
commented on paper 44
that you are also reviewing:
...

9.06 am Oct 20, 2022
Anonymous Reviewer #2
commented on paper 63
that you have authored:
"Bad paper. Reject."

Figure 1: Cartoon illustration of reviewer de-anonymization due to batching.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

ommend performing tasks like responding to emails in batches in order to improve efficiency and reduce work-related stress.

In peer-review forums such as computer science conferences, reviewers and meta-reviewers are often assigned multiple papers. We find empirically that reviewers and meta-reviewers are highly likely to batch their comments and/or reviews. Specifically, we analyze data from a top Computer Science conference[1] with thousands of papers, reviewers, and discussion comments. We find that when reviewers and meta-reviewers comment on multiple papers, they have a 30.10% chance of batching their comments within 5 minutes of one other. In comparison, any randomly chosen pair of reviewers and meta-reviewers had only a 0.66% chance of making comments on different papers within 5 minutes of each other.

While batching is normal human behavior, it introduces a risk of deanonymization in peer-review settings.[2] For example, in many open peer-review settings, comments are publicly posted. Furthermore, many conferences have policies that (meta-)reviewers for any paper know the identities of other (meta-)reviewers on that paper. Now, when a (meta-)reviewer batches their comments, an author may observe that two comments are generated at nearly the same time on their own paper and on another paper. The author can then link the identity of this anonymous (meta-)reviewer on their own paper to a (meta-)reviewer on the other paper. If the author knows the identity of the (meta-)reviewers on the other paper—for instance, if the author is the meta-reviewer or another reviewer for that paper—this can uncover the identity of the (meta-)reviewer of their own paper. See Figure 1 for a cartoon illustration.

A back-of-the-envelope calculation based on our aforementioned measurements in peer review suggests that if an author has a uniform prior over 10 possible (meta-)reviewers of their paper, then after observing a comment posted on their own paper within 5 minutes of another comment from one of these (meta-)reviewers on another paper, their posterior probability that this (meta-)reviewer made the comment increases to $\frac{0.301}{0.301+9(0.0066)} = 83.51\%$ as compared to the prior of 10%. Thus, the linking of (meta-)reviewers across papers using batched comments can undermine the anonymity of the peer review process.

Similar privacy risks due to batching arise in many systems where users generate publicly logged events under pseudonyms. For instance:

- *Inferring the identity of editors on Wikipedia articles.* Wikipedia provides public edit histories of articles. While edit history is public, Wikipedia users are known to maintain their anonymity for a variety of important reasons. For instance, one study of Wikipedia editors who use the anonymity network Tor found that editors are concerned about their privacy due to risks like "*threats of surveillance, violence, harassment, opportunity loss, reputation loss, and fear for loved ones.*" [13] These risks are especially acute for already marginalized groups like women and ethnic minorities. Thus, the study concludes that "*open collaboration communities must go beyond attracting participants, to develop social and technical arrangements that support contributors' needs for privacy.*"

  In order to address these privacy concerns, Wikipedia's terms of service explicitly allow for the use of a pseudonymous alternate account: *"A person editing an article that is highly controversial within their family, social or professional circle, and whose Wikipedia identity is known within that circle, or traceable to their real-world identity, may wish to use an alternative account to avoid real-world consequences from their editing or other Wikipedia actions in that area.*" [43] However, as in the peer review example, batched timing of article revisions can enable linkage of the second account to a known primary account. In practice, the batching of edits is ubiquitous on Wikipedia; our analysis of publicly logged Wikipedia article revisions shows that over 50% of all edits are made within 5 minutes of an edit from the same user on a different article. This common editing behavior may undermine the privacy of users employing a second account to preserve their anonymity.

[1] Name redacted for privacy.

[2] This outcome is bad for a review system that needs a lot of interaction with the authors, but not for conferences where this is not expected nor allowed, like AAAI and IJCAI. The conference we analyzed was not on OpenReview.net but on a different conference management platform that does not make discussions public and has only a single-shot interaction between reviewers and authors (via a "rebuttal"). It is of interest to see an analogous analysis on conferences on OpenReview.net, but we do not have access to such data.

- *Clustering crypto-currency transactions on a public blockchain.* In cryptocurrencies like Bitcoin, users' transaction histories are recorded on a public blockchain where a person can send or receive currency to an associated public key, which acts as a pseudonym. Users can have multiple addresses, each containing its own funds and identified by a different public key [3]. A transaction can (and often does) draw funds from multiple input addresses, particularly if no single address contains sufficient funds for a given transaction [1]. However, a common heuristic used in practice is to link multiple input addresses to a single transaction to the same user [2, 31]. Hence, users who wish to preserve their privacy can separate inputs from different addresses into different transactions to obfuscate the linkage between transactions from the same person [2].[3] However, if a user batches these transactions in time across addresses, an adversary may use this timing (along with other signals) to still link together their multiple addresses. Linking pseudonyms together is a common first step in a full deanonymization attack. For instance, attacks on Bitcoin transactions begin by leveraging a user's "idioms of use" to cluster together addresses likely belonging to the same person [31]. The attacker then leverages a single known link to a real-world identity to de-anonymize the entire cluster.

These scenarios motivate the need for defenses against timing-based linkage attacks that exploit the batching of tasks by people. There is already extensive literature on privacy-preserving data release in various settings. However, there are a number of strict constraints in our setting that prevent these methods from being applicable. A common approach to preserving privacy is to introduce fake events to obscure patterns among the real events. However, in all three applications — peer review, Wikipedia, and cryptocurrency — generating fake events is highly undesirable or impractical, and withholding events indefinitely is also not possible. In the setting of commenting in peer review, it is undesirable to generate fake comments, as this would require giving made-up feedback to paper authors. Similarly, in Wikipedia, adding fake edits to articles can undermine the quality and legitimacy of the content. For cryptocurrencies, introducing dummy transactions would introduce additional financial cost, causing undesirable overhead. Furthermore, transactions include the amount of currency sent, so dummy transactions would require a sender to transfer actual funds just to preserve privacy. Instead, our approach is to design *delay mechanisms* that introduce random delays to the time at which events are posted on the platform (without the use of any dummy data) to preserve privacy. Thus, the mechanism will trade off privacy for additional delay in the system.

Our contributions. In this work, we introduce the problem of anonymity compromise due to the batching of tasks in pseudonymous forums and then propose defenses. Our primary contributions are:

- We identify the problem of deanonymization risk due to the batching of tasks by users of pseudonymous online forums. By analyzing data from an actual peer-reviewed conference, we demonstrate that a simple attack using the timing of comments on an online forum can link anonymous (meta)-reviewer's identities, increasing their certainty about a specific (meta)-reviewer's identity to 83% from a prior of 10%. In analysis of Wikipedia article revisions, we show that batching of tasks on Wikipedia makes it possible to link editors across articles with an accuracy of 85% based only on the timing of their revisions.
- We formulate the problem of trading off privacy and delay in pseudonymous forums where users engage in batching. We show that standard notions of differential privacy (DP) [10] cannot be satisfied in our problem setting without introducing fake events or withholding events indefinitely. Therefore, we consider a "one-sided" relaxation of traditional DP [22]. Our formulation aims to prevent an adversary from inferring when batching happened, but allows an adversary to learn that batching did not happen.
- We propose a general framework for designing mechanisms that guarantee one-sided DP by adding independent random delay to batched and unbatched comments. We show that we can instantiate this framework with a number of different distributions and guarantee privacy. Notably, it is possible to guarantee privacy with non-negative versions of typical distributions used for differential privacy like the Laplace distribution and the Staircase distribution. It is also possible to guarantee privacy at any setting of the privacy parameters by adding delay drawn from a uniform distribution with inflated probability mass at 0, which we call the Zero-Inflated Uniform Mechanism.

[3] There exist other cryptographic solutions (e.g., CoinJoin) that leak more information in exchange for cost benefits compared to generating multiple transactions [30].

- We establish the optimality of our Zero-Inflated Uniform Mechanism among mechanisms that add independent noise to each comment. In particular, we give a full characterization of the Pareto frontier of the expected delay added to batched and unbatched events by any mechanism that adds independent non-negative noise to comments, at any setting of privacy parameters, and show that our proposed mechanism achieves this frontier. This result may be of independent interest. While the uniform distribution is not typically used in the design of two-sided DP algorithms, our results show that for one-sided DP when only non-negative noise can be added (as is the case for streaming timing data) the Zero-Inflated Uniform Mechanism can optimally trade off privacy for utility.
- We conduct a series of experiments simulating linkage attacks using batched timing of tasks on Wikipedia article revision data and Bitcoin transaction data. These experiments reveal the applicability of our methods in preserving privacy in practice without exceedingly large delays.

All of our code is available online at https://github.com/akgoldberg/batching-privacy.

2 Related Work

There is a substantial body of work on anonymity when sending packets over a network. However, as we discuss below, the techniques developed therein are inapplicable to our setting. Specifically, prior work has described deanonymization attacks which leverage correlated timing of packet arrivals. The work gives various defenses against such attacks [37, 26, 19, 16, 38, 39]. Anonymous networking seeks to prevent an adversary from inferring the sender and recipient of a given message. Packets are routed through a sequence of "mix nodes" to obscure the path taken. The highly correlated arrival times of packets on the first mix node and the last mix node in one path can enable inferences that a specific sender and recipient are communicating with one another. Prior work [37, 26] demonstrates the practical viability of deanonymization attacks that take advantage of batching in anonymous networks.

The defenses proposed in these papers rely on the introduction of *dummy packets* or "cover traffic" to a network, obscuring any instance of batching amidst many instances of spurious batching. In contrast, a critical constraint in the settings we consider is the *infeasibility of generating fake data* as a means of preserving privacy. Therefore, our work will consider mechanisms that delay batched arrivals in order to preserve anonymity, trading off delay for privacy, without introducing any synthetic data.

Our work defines privacy based on a "one-sided" relaxation of the popular notion of differential privacy [10]. The definition of one-sided DP was introduced in the paper [22] in a setting where contributors of individual data-points to a database have different privacy constraints and hence data-points can be classified as "sensitive" and "non-sensitive." In our work, we argue that this classification of sensitive and non-sensitive data-points is applicable to batched and unbatched events. Interestingly, while the paper [22] shows that one-sided DP can improve utility compared to standard two-sided DP, we find that in our problem setting, one-sided DP admits useful privacy-preserving algorithms where two-sided DP does not admit any useful algorithms at all. We cannot readily apply algorithms from the paper [22] due to the constraint that we publish all data. Therefore, while they develop mechanisms that release a subset of non-sensitive data with no noise addition, while withholding all sensitive data entirely, we consider mechanisms that add noise to both sensitive and non-sensitive data-points and release all data-points.

Geng and Viswanath [15] address the question of optimal distributions for noise addition in standard differential privacy. They show that in order to minimize the magnitude of noise added to a query with known sensitivity, noise should be drawn from a "staircase" distribution, which has a probability density function that is roughly a piece-wise constant approximation of the Laplace distribution. Our work can be seen as an analogous result in the one-sided DP regime. Specifically, we prove that for the one-sided relaxation of differential privacy, adding staircase noise is no longer optimal, but rather adding uniform noise with a possibly inflated probability of sampling 0 minimizes the magnitude of noise addition.

Our running application in this paper is that of peer review. A few previous papers have considered certain issues of privacy in peer review, but with very different objectives and methods. The paper [6] considers the problem of miscalibration [12, 35, 14, 41] in peer review. They consider privacy leakage when correcting for such miscalibration and provide methods (for a simplified setting) to mitigate this leakage. The paper [7] provides privacy-preserving algorithms for releasing some peer-review data to allow researchers at large to analyze and address problems like subjectivity [25, 34] and miscalibration. The paper [18] considers the problem of coalition-based fraud [40, 27, 44, 17] in peer review, and provides a randomized algorithm to assign reviewers to papers to mitigate such fraud. They argue that such a randomized assignment algorithm has another benefit: it can allow for release of the data that underlies the automated assignment algorithm while still preserving some privacy about which paper was assigned to which reviewer. We refer the reader to [36] for an overview of research on peer review.

3 Problem Formulation

We now describe our problem formulation. For clarity of exposition, we use the running example of peer-review.

Comment Arrivals. We call the event when a reviewer makes a comment on a paper a *comment arrival*. Each comment arrival consists of 4 elements: the text of the comment, a timestamp t when the comment arrived, a paper p to which it responds, and the reviewer r who made the comment. We assume that comments arrive in continuous time over an infinite time horizon, as this is the most general setup, although our analysis extends to any finite time horizon (for example, in the case where a conference has an end time after which comments can no longer be posted). We consider settings where the comments are publicly observable, as is the case for many conferences run on popular platforms like OpenReview.net.

Batching. In our initial model, we consider comments to be "batched" if they arrive simultaneously. Specifically, a set of 2 or more comment arrivals is *batched* if all comments in the set come from the same reviewer at the same time, and furthermore, the comments are all on different papers. In Section 5.2 we discuss how to extend the model to allow for a short gap between batched comments.

Comment Posting Mechanism. A comment posting mechanism $\mathcal{M}$ receives comments as they arrive and can choose to delay when they are posted, with the comments only becoming publicly visible at the time they are posted. The mechanism receives a streaming set of comment arrivals A as input. It outputs a set of comments where each comment has identical content, paper, and reviewer to a comment in the input but with a potentially delayed timestamp. We place the following natural constraints on any *valid* comment posting mechanism:

1. (*Delay-Only*) If a comment arrives at time t it must be output at time t or later.
2. (*No Fake Data*) Any comment posted at time t must have arrived at or before time t.
3. (*Eventual Release of All Comments*) For any comment, letting d denote the potentially randomized delay introduced to the comment by the mechanism, it must be that $\lim_{D\to\infty} \Pr[d \leq D] = 1$.

Privacy. Our goal is to protect against an adversary who is trying to infer whether a specific pair of comment arrivals was batched. Following the widely-adopted framework of differential privacy, we consider a strong adversary who knows exactly when all comments arrived, except for one pair of comments that either arrived in a batch or at separate times. The adversary knows that comments arrive at the same time if batched and knows the exact inter-arrival time of the pair of comments if they arrive unbatched. In preserving privacy against such a strong adversary, we also provide privacy guarantees for general classes of weaker adversaries with less prior knowledge. For instance, in Section 5.1 we discuss an adversary who only has an estimate of the baseline distribution of inter-arrival times when comments are unbatched, rather than the exact inter-arrival time.

Ideally, we would like to provide a privacy guarantee with respect to the standard notion of differential privacy (DP). Such a DP guarantee would promise difficulty of distinguishing whether the mechanism was run on one of two neighboring inputs, where one input has an additional batched pair of comments compared to its neighbor. Unfortunately, as we prove in Section 4.4, it is impossible to guarantee standard ϵ-DP in this setting. There are two main reasons for this impossibility.

First, consider defining neighboring inputs to a DP mechanism where a pair of comments arrives simultaneously in one input when batched, but arbitrarily far apart when unbatched in the neighboring input. Then, to satisfy a traditional DP guarantee, batched comments must be delayed *indefinitely* to make these two inputs indistinguishable. Second, even with a bounded change in arrival time for any comment on neighboring inputs, we show that if the neighboring relation is symmetric (i.e., a pair of comments can be batched in one input and unbatched in the other, and it doesn't matter which input contains the batched comments), then to satisfy ϵ-DP the mechanism must delay a batched comment indefinitely.

In order to address the aforementioned roadblocks, we relax the definition of neighboring inputs in two ways. First, we introduce a real-valued parameter $g > 0$ into our formulation of neighbors that bounds how far in time a batched comment can move in a neighboring input where it arrives unbatched. Second, we define neighbors in a one-sided manner: a set of comment arrivals neighbors another set only if it contains one *additional* pair of batched comments as compared to its neighbor. In contrast, a set of comment arrivals does not neighbor another set if it contains one *fewer* pair of batched comments than its potential neighbor. Formally, we define neighboring comment arrival sets as follows:

Definition 3.1 (g-Neighboring Comment Arrival Sets)**.** A set of comment arrivals $A^{(B)}$ is g-*neighboring* to set of comment arrivals A, if $A^{(B)}$ can be obtained from A by batching together one pair of comments that arrive separately in A. The comments must arrive within g units of time of one another in A and the later comment moves to the earlier comment in $A^{(B)}$ to create a batch. Formally, $\exists (c, t, p, r), (c', t', p', r) \in A$ such that $p \neq p'$, $0 < t' - t \leq g$ and $A^{(B)} = (A \setminus \{c'\}) \cup \{(c', t, p', r)\}$.

Note that this definition of adjacency is asymmetric as a set of comment arrivals with no pairs of batched comments is not g-adjacent to any other sets of comment arrivals. As an example, consider the following pair of comment arrival sets A and $A^{(B)}$:

$$\begin{aligned} A =& \{(c_1, t = 1, p_1, r_1), (c_2, t = 2, p_1, r_2), \boldsymbol{(c_3, t = 3, p_2, r_1)}\}, \quad \text{and} \\ A^{(B)} =& \{(c_1, t = 1, p_1, r_1), \boldsymbol{(c_3, t = 1, p_2, r_1)}, (c_2, t = 2, p_1, r_2)\}. \end{aligned}$$

Then under our definition above, $A^{(B)}$ is 2-neighboring to A. However, A is *not* 2-neighboring to $A^{(B)}$.[4]

Now, we define privacy of a mechanism using a notion similar to the definition of one-sided differential privacy introduced in [22]. We note that apart from the one-sidedness of neighbors, our privacy formulation differs substantially from that of [22] as we focus on inputs differing in the timing of a pair of comments due to batching, while [22] considers databases where arbitrary entries are considered non-private.

For any finite time horizon T and set of comment arrivals A, we will let $\mathcal{M}_T(A)$ denote the output of the mechanism up to time T. Then, we define privacy as follows:

Definition 3.2 ((ϵ, g)-One-Sided Differential Privacy (OSDP))**.** For any $\epsilon \geq 0$ and $g > 0$, a comment posting mechanism $\mathcal{M}$ is (ϵ, g)-one-sided differentially private if for any $A, A^{(B)}$ such that $A^{(B)}$ is g-neighboring to A, for any time horizon T, and for any subset of possible outputs $S \subseteq \text{Range}(\mathcal{M}_T)$ of the mechanism:

$$\Pr[\mathcal{M}_T(A^{(B)}) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}_T(A) \in S].$$

This privacy definition guarantees that the likelihood of observing an outcome on an input with at least one instance of batching is never much larger than the likelihood of observing that outcome on an input with one *fewer* batched pair. Therefore, the mechanism obscures the fact that any pair of comments was batched. However, it is possible for the mechanism to reveal that a pair of

[4] The reader may have observed that the definition of neighboring comment arrival sets has a technical condition that a batched comment moves later in time in a neighboring input with one fewer instance of batching. It is possible to modify the formulation to let a batched pair of comments arrive at either one of the later or earlier arrival times of an unbatched pair in an adjacent input. This modified formulation would capture an even stronger adversary who knows the exact time-frame in which a batched pair arrives. However, ensuring privacy against this adversary would require even more delay added to the system, Hence, we do not pursue this formulation.

comments was unbatched; we allow for outputs that occur with non-zero probability given input A but zero probability given input $A^{(B)}$ (unlike in standard two-sided DP). We argue that the one-sided definition effectively captures privacy risk due to batching, as the presence of a batched pair of comments is sensitive information, while the absence of batching is non-sensitive. We further discuss the motivation for only treating batching as sensitive via the concrete example of reviewer deanonymization by a meta-reviewer.

The privacy definition requires two parameters: ϵ and g. The interpretation of ϵ is similar to two-sided DP as it quantifies the "level" of privacy: for smaller ϵ it is harder to distinguish neighboring inputs, whereas for larger ϵ it is easier to distinguish neighboring inputs. The g parameter captures domain knowledge about what types of inputs can be neighbors, similar to restricting the domain of inputs in two-sided DP. Roughly, g should capture how far apart consecutive comments would plausibly arrive if batching were not occurring. It is necessary for a practitioner to include this domain knowledge in the form of finite value g as we prove that batched comments must be delayed by at least g (in Section 4.3) and hence without this bound, comments must be withheld indefinitely. We give heuristics for how to set g based on a hypothesis testing interpretation of the privacy definition in Section 5.1.

Utility. We measure the cost of our mechanism in terms of expected delay added to comments. Because the privacy guarantee is asymmetric, the mechanism can behave differently on batched and unbatched comments. Therefore, we will consider measuring utility in terms of expected delay to batched comments denoted $\mathbb{E}[B]$, expected delay to unbatched comments denoted $\mathbb{E}[U]$ or more generally any weighted sum of the two expectations.

Goal. Our goal is to design comment posting mechanisms that guarantee (ϵ, g)-one-sided differential privacy for chosen privacy parameters ϵ and g while minimizing the expected delay added to comments. We may add random delay to batched and unbatched comments drawn from different distributions B and U respectively. Therefore, we wish to design (ϵ, g)-OSDP mechanisms that are Pareto optimal in trading off between $\mathbb{E}[B]$ and $\mathbb{E}[U]$ at any setting of ϵ and g. Moreover, we want to allow practitioners to choose a mechanism on this Pareto frontier that minimizes an appropriate cost function suiting the requirements of their system. For instance, a system with a higher rate of batching may wish to weight delay to batched comments higher in their cost function than a system with a lower rate of batching. To this end, we consider minimizing any cost function that is a convex combination of expected delay to batched and unbatched comments. We aim to provide the exact mechanism on the Pareto frontier that minimizes $w\mathbb{E}[B] + (1-w)\mathbb{E}[U]$ for any choice of weighting parameter $w \in [0, 1]$ and any privacy parameters ϵ and g. We note that this choice of utility function is without loss of generality. In particular, the feasible region of $\mathbb{E}[B]$ and $\mathbb{E}[U]$ is convex (as we prove in Appendix A.4, Lemma A.8.) Therefore, any mechanism that is Pareto optimal in trading off $\mathbb{E}[B]$ and $\mathbb{E}[U]$ minimizes the weighted cost function for some choice of w (since any point on the Pareto frontier of a convex feasible region optimizes some weighted sum objective per Boyd [5, Chapter 4.7]).

Example: De-anonymizing reviewers. We now discuss the one-sided nature of privacy risk inherent to batching using the running example of a meta-reviewer de-anonymizing a reviewer or meta-reviewer of a paper they have authored. Recall the introductory scenario where an meta-reviewer observes two comments c and c' that arrive consecutively on different papers and are made by (meta)-reviewers r_1 and r' respectively (where it is possible that $r' = r_1$). The meta-reviewer knows that the first comment was made by r_1 and has a uniform prior over K possible reviewers who could have made c' (including r_1). They wish to de-anonymize r' based on whether or not c' arrived in a batch with c. From our aforementioned analysis of a conference peer review where we define two comments as "arriving together" if they arrive within 5 minutes of one another, we estimate that: $\Pr[c, c' \text{ arrive together} \mid r' = r_1] \approx 0.3$, while $\Pr[c, c' \text{ arrive together} \mid r' \neq r_1] \approx 0.0066$. Therefore, after learning that c and c' arrived together, the meta-reviewer's posterior puts the most weight on $\Pr[r' = r_1 \mid c, c' \text{ arrived together}] = \frac{0.301}{0.301+0.0066(K-1)}$. On the other hand, after learning that c and c' did not arrive together, their posterior puts the most weight on: $\Pr[r' = r_k \mid c_1, c_2 \text{ did not arrive together}] = \frac{0.9934}{0.699+0.9934(K-1)}$ for $K \neq 1$. We give further detail on how these statistics were estimated in Appendix B.

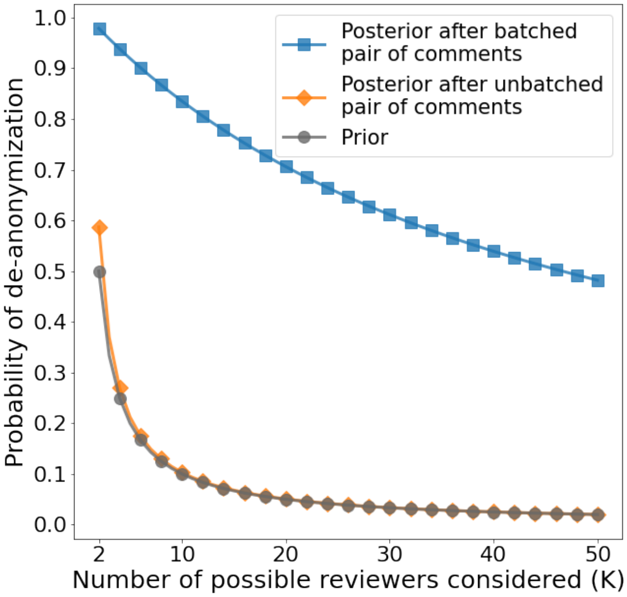


Figure 2: Success probability of de-anonymizing a (meta)-reviewer after learning that a pair of comments arrived together vs. learning that a pair of comments did not arrive together.

As shown in Figure 2, in learning that the pair of comments was batched, the meta-reviewer can identify the (meta)-reviewer of a paper they authored with much higher confidence than before observing the batched timing; on the other hand, by learning that the pair of comments was unbatched, the meta-reviewer's posterior hardly changes from the prior. Our one-sided privacy definition captures this asymmetric privacy risk. It ensures that an adversary does not learn much about the sensitive information of whether two comments are likely to be batched after observing the time that comments get posted, while allowing the adversary to potentially learn the insensitive information that two comments were unbatched.

4 Theoretical Results

In this section, we present our main theoretical results. First, in Section 4.1, we propose an algorithmic framework to design comment posting mechanisms that guarantee (ϵ, g)-one-sided differential privacy under batching. In this framework, we add random noise to the timestamps of batched and unbatched comments, drawing the noise from a pair of distributions (B, U) that depend on parameters ϵ and g.

Within this framework, there are many possible choices of the noise distributions (B, U), and we investigate them in Section 4.2. For instance, one could use one-sided analogues of distributions commonly used for two-sided differential privacy, like exponential noise, which is the absolute value of the Laplace distribution [10], or one-sided staircase noise [15] (whose two-sided version is known to be optimal for two-sided DP [15]). However, we show that perhaps surprisingly, these distributions are all sub-optimal for the privacy-delay trade-off.

In Section 4.3 we provide another distribution – a zero-inflated uniform distribution with carefully chosen parameters – that we show guarantees one-sided differential privacy in our setting and also achieves a Pareto-optimal privacy-delay trade-off.

Finally, in Section 4.4, we motivate the usefulness of our one-sided DP formulation as a means of capturing the privacy-delay trade-off by showing that the popular two-sided definition of DP does not yield a useful privacy-delay trade-off for valid comment posting mechanisms.

4.1 Algorithmic Framework

In Algorithm 1, we present a general recipe for designing randomized delay mechanisms. The meta-algorithm receives as input privacy parameters ϵ and g as well as probability distributions B and U that depend on ϵ and g. We will then prove that if pairs of distributions satisfy an "indistinguishability" property then Algorithm 1 yields a (ϵ, g)-OSDP mechanism.

Algorithm 1 Framework for Designing a Randomized Delay Mechanism

```
Input: privacy parameter $\epsilon > 0$, maximum time gap $g > 0$, noise addition distributions $B$ and $U$
for each comment arrival time $t$ do
  if a set of batched comments arrives then
    For each comment, independently sample $d \sim B(\epsilon/2, g)$ and post the action at time $t + d$.
  else if if an unbatched comment arrives then
    Post the comment at time $t + d$ where $d \sim U(\epsilon/2, g)$
  end if
end for
```

Mechanisms within this framework satisfy two useful qualitative properties for deployment in real applications. First, because the noise is sampled at arrival time, we can tell each user the duration of the delay on their comment as soon as they create it. Second, because the noise is sampled independently for each comment, the algorithm does not require a centralized coordinator to determine post times. This enables privacy-sensitive individuals to implement the algorithm for themselves. This ability to be implemented locally is a necessary property for use in cryptocurrencies where there is no central trusted server.

Now, any choice of (B, U) can satisfy (ϵ, g)-OSDP as long as B and U are indistinguishable in the following sense:

Definition 4.1 (One-Sided Indistinguishable Distributions)**.** Let B and U be non-negative random variables. We say that the ordered pair (B, U) is (ϵ, g)-*one-sided indistinguishable* if, for any measurable set $S \subseteq \mathbb{R}$ and any $t_0 \in [0, g]$, the distributions satisfy:

$$\Pr[B \in S] \leq e^{\epsilon} \Pr[U \in S - t_0],$$

where for any $S \subseteq \mathbb{R}, t \in \mathbb{R} : \ S - t = \{s - t | s \in S\}$.

The following theorem shows sufficiency of such one-sided indistinguishable distributions for guaranteeing privacy.

Theorem 4.2 (Privacy of Randomized Delay Mechanisms)**.** *Let (B, U) be any pair of $(\epsilon/2, g)$-one-sided indistinguishable distributions. Then, Algorithm 1 using B and U as noise-addition distributions guarantees (ϵ, g)-one-sided differential privacy.*

We give the proof of the above theorem in Appendix A.1. The proof follows by observing that in neighboring inputs, a pair of comments that was batched becomes unbatched with one comment arrival moved forward by at most g time units. Hence, if B and U have a likelihood ratio bounded by $e^{\epsilon/2}$ for any values within g time units of one another, it is hard to distinguish whether the mechanism was given an input with two unbatched comments arriving g time units apart or two batched comments arriving at the same time (up to a multiplicative factor of e^{ϵ}).

4.2 Privacy-preserving delay distributions

We now describe a number of possible choices for (ϵ, g)-one-sided indistinguishable distributions (B, U) that can be used in our algorithmic framework. We show that we can use an exponential distribution, which is the one-sided version of the Laplace distribution. We can also add noise from the absolute value of the staircase distribution, which was proven in [15] to be optimal for noise addition in two-sided DP, giving smaller delay than the exponential. Alternatively, we can add noise to unbatched comments drawn from a zero-inflated uniform distribution where we add 0 delay with probability $1 - \eta$ (for some parameter η) and delay drawn from a uniform distribution with probability η.

Theorem 4.3 (Choices of One-Sided Indistinguishable Distributions)**.** *The following choices of B and U are (ϵ, g)-one-sided indistinguishable:*

(1) Exponential[5]*:* $B = g + \mathit{Exponential}(\epsilon/g)$, $U = \mathit{Exponential}(\epsilon/g)$

(2) Staircase [15][6]*:* $B = g + |\mathit{Staircase}(\epsilon, g)|$, $U = |\mathit{Staircase}(\epsilon, g)|$

(3) Uniform*:* $B = \mathit{Uniform}(g, \frac{1}{1-e^{-\epsilon}}g)$, $U = \mathit{Uniform}(0, \frac{1}{1-e^{-\epsilon}}g)$

(4) Zero-inflated Uniform with parameter η. *For* $e^{-\epsilon} < \eta \leq 1$*:*

$$B = \mathit{Uniform}\left(g, \tfrac{\eta}{\eta - e^{-\epsilon}}g\right)$$

$$U = \begin{cases} 0 & \text{with probability } 1-\eta \\ \mathit{Uniform}\left(0, \tfrac{\eta}{\eta - e^{-\epsilon}}g\right) & \text{with probability } \eta. \end{cases}$$

These choices of (B, U) *incur the following expected delays:*

(1) Exponential: $\mathbb{E}[B] = g(1 + \frac{1}{\epsilon})$ *and* $\mathbb{E}[U] = g\frac{1}{\epsilon}$

(2) Staircase*:* $\mathbb{E}[B] = g(1 + \frac{e^{\epsilon/2}}{e^{\epsilon}-1})$, $\mathbb{E}[U] = g\frac{e^{\epsilon/2}}{e^{\epsilon}-1}$

(3) Uniform: $\mathbb{E}[B] = \frac{1}{2}g\left(1 + \frac{e^{\epsilon}}{e^{\epsilon}-1}\right)$ *and* $\mathbb{E}[U] = \frac{1}{2}g\left(\frac{e^{\epsilon}}{e^{\epsilon}-1}\right)$

(4) Zero-inflated Uniform with parameter η*:* $\mathbb{E}[B] = \frac{1}{2}g\left(\eta + \frac{\eta e^{\epsilon}}{\eta e^{\epsilon}-1}\right)$ *and* $\mathbb{E}[U] = \frac{1}{2}g\left(\frac{\eta^2 e^{\epsilon}}{\eta e^{\epsilon}-1}\right)$.

The proof of the above theorem can be found in Appendix A.2. Note that the (uniform, uniform) noise additions are a special case of (uniform, zero-inflated uniform) taking $\eta = 1$. We highlight them separately in Section 4.3 as we introduce the zero-inflated uniform distribution for the first time here. In the next section, we show that a zero-inflated uniform distribution is Pareto optimal for appropriate choice of η.

Notably, the choice of parameters for the exponential and staircase distributions given in Theorem 4.3 are the optimal choice of parameters in the sense that they minimize expected delay at fixed values of privacy parameters ϵ and g when adding i.i.d. exponential or staircase noise plus a constant offset to all comments:

Theorem 4.4 (Optimal Choice of Parameters for the Exponential and Staircase Distributions)**.** *Let* B, U *be non-negative noise-addition distributions that guarantee* (ϵ, g)*-OSDP when used in Algorithm 1 where* $B = a_B + D$ *and* $U = a_U + D$ *for constants* $a_B, a_U > 0$ *and non-negative random variable* D*. Then, if* D *is an exponential random variable or a staircase random variable,* $\mathbb{E}[B]$ *and* $\mathbb{E}[U]$ *are minimized at any values of* ϵ, g *by the choice of parameters in Theorem 4.3 such that* B, U *are* $(\epsilon/2, g)$*-one-sided indistinguishable.*

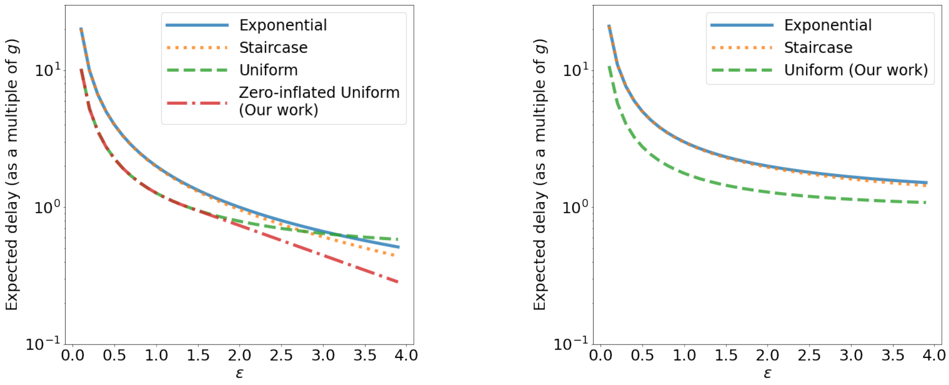
The proof of the above theorem can be found in Appendix A.3. By Theorem 4.3 and Theorem 4.4, adding i.i.d. exponential or staircase noise plus a constant offset is strictly sub-optimal in minimizing expected delay as zero-inflated uniform noise can achieve lower delay at the same privacy level.

Corollary 4.5. *Among* (ϵ, g)*-OSDP mechanisms following the framework of Algorithm 1, taking* B *and* U *to be i.i.d. exponential or staircase distributions (with constant offsets) is strictly sub-optimal in minimizing* $\mathbb{E}[B]$ *and* $\mathbb{E}[U]$ *for any values of* ϵ *and* g*. In particular, using the zero-inflated uniform mechanism with appropriate choice of* η *can achieve lower expected delay for both* $\mathbb{E}[B]$ *and* $\mathbb{E}[U]$ *at any values of privacy parameters* ϵ *and* g*.*

In this setting, the exponential and staircase distributions typically used in two-sided DP add significantly more delay than zero-inflated uniform noise, especially at small values of ϵ. In Figure 3, we show the expected delay for the optimal exponential, staircase, uniform, and zero-inflated uniform at

[5] In the notation to follow, we parameterize the exponential distribution by its rate.

[6] The staircase distribution is parameterized by 3 values ϵ, Δ, g in [15]. Here, we take Staircase(ϵ, g) to mean the staircase distribution with $\epsilon = \epsilon$, $\Delta = g$ and $\gamma = \frac{1}{1+e^{\epsilon/2}}$, which is the optimal value of γ to minimize expectation per [15].



(a) Expected delay to unbatched comments

(b) Expected delay to batched comments

Figure 3: Expected delay (as a multiple of g) for Algorithm 1 with the exponential, staircase, zero-inflated uniform and uniform distributions at varying values of ϵ. All distributions use the optimal setting of parameters at a given ϵ. The zero-inflated uniform parameter is chosen to minimize delay to unbatched comments. The delay (y axis) is plotted on a log scale.

Algorithm 2 Zero-Inflated Uniform Mechanism

```
**Input:** privacy parameter $\epsilon > 0$, maximum time gap $g > 0$, weighting of expected delay to batched comments $w \in [0, 1]$
Set $\eta = \min \left\{ e^{-\epsilon/2} \left( 1 + \sqrt{1 + e^{\epsilon/2} \frac{w}{1-w}} \right), 1 \right\}$
for each comment arrival time $t$ do
  if a set of batched comments arrives then
    For each comment, independently sample $d \sim \text{Uniform}\left(g, \frac{\eta}{\eta - e^{-\epsilon/2}} g\right)$ and post the comment at time $t + d$.
  else if an unbatched comment arrives then
    Post the comment at time $t + d$ where $d = 0$ with probability $1 - \eta$ and $d \sim \text{Uniform}\left(0, \frac{\eta}{\eta - e^{-\epsilon/2}} g\right)$ with probability $\eta$.
  end if
end for
```

each setting of ϵ. For both batched and unbatched comments, the uniform and zero-inflated uniform distributions add a factor of nearly two times less delay than the staircase and exponential at small values of ϵ. For larger values of ϵ, all of the aforementioned distributions add similar delay, with uniform adding the least delay to batched comments and the zero-inflated uniform adding the least delay to unbatched. In the next section, we formally prove that zero-inflated uniform noise is Pareto optimal and characterize the optimal choice of η for any objective function that is a weighted sum of $\mathbb{E}[U]$ and $\mathbb{E}[B]$ based on the setting of ϵ.

4.3 Pareto-optimal Algorithm

In this section, we derive the Pareto frontier (trading off the expected delay for batched and unbatched comments) of noise-addition distributions for a given (ϵ, g)-one-sided indistinguishability constraint (Definition 4.1). We show that adding zero-inflated uniform noise with appropriate choice of parameter η achieves optimal expected delay among mechanisms that add independent noise to each comment. While the optimality result holds only within the class of mechanisms that adds *independent* noise to each comment, this constraint allows for an algorithm to be implemented locally without requiring coordination by a centralized server. This constraint is a common property of many deployed privacy-preserving algorithms. For instance, local differential privacy [21] requires that randomization needed for privacy is added locally by each holder of a data-point, and the Tor anonymous network [8] protocol requires that initiators of connections choose the (random) path on which to send a message themselves.

Given an (ϵ, g)-one-sided privacy constraint, our algorithmic framework (Algorithm 1) has many choices of noise-addition distributions that can guarantee privacy. In terms of delay, there are two quantities to optimize – the delay incurred by batched comments and that incurred by unbatched comments. A natural utility objective to consider is a convex combination of the two expectations:

$$w\mathbb{E}[B] + (1-w)\mathbb{E}[U], \qquad \text{for a given parameter } w \in [0,1].$$

The parameter $w \in [0,1]$ determines how much weight is given to batched comments in the utility function. For example, a user of our algorithm may estimate the relative rate of batching in the system and set w to this value to optimize for the overall average expected delay across all comments.

We present our main algorithm as Algorithm 2. Our algorithm follows our previously introduced framework (Algorithm 1). It chooses U as a zero-inflated uniform distribution with a carefully chosen value of parameter η (dependent on ϵ and w), and chooses B as a uniform distribution. The following theorem now proves that for any privacy parameters our algorithm is indeed Pareto optimal – it optimally trades off privacy and unbatched delay and batched delay.

Theorem 4.6 (Pareto optimality of the Zero-Inflated Uniform Mechanism)**.** *Algorithm 2 is Pareto optimal between expected delay to batched and unbatched comments at a given setting of (ϵ, g) among valid (ϵ, g)-OSDP mechanisms that add independent noise to each comment. Further, given weight parameter $w \in [0,1]$ and privacy parameters (ϵ, g) as input, Algorithm 2 minimizes cost function $w\mathbb{E}[B] + (1-w)\mathbb{E}[U]$ at any given privacy level (ϵ, g) among mechanisms adding independent noise drawn from distributions B and U to batched and unbatched comments respectively.*

We give a proof sketch below, for the full proof see Appendix A.4.

Proof sketch. Roughly, the proof proceeds as follows:

- We consider any (ϵ, g)-indistinguishable noise addition distributions (B, U) added to batched and unbatched comments respectively. Using results from [15], we argue that for large enough $i \in \mathbb{N}$, we can approximate B and U arbitrarily well with random variables that have piece-wise constant probability density functions and each constant interval has length g/i.
- We establish properties of any Pareto optimal (B_i, U_i) by directly proving that we can decrease the expectation of both $\mathbb{E}[B_i]$ and $\mathbb{E}[U_i]$ for any pair of distributions that violates these properties. Taken together the properties yield the exact form of any Pareto optimal B_i and U_i. Taking limits as $i \to \infty$ gives that the Pareto frontier is realized by uniform and zero-inflated uniform distributions for some setting of η. The proof follows by directly proving that we can decrease the expectation of both $\mathbb{E}[B_i]$ and $\mathbb{E}[U_i]$ for any pair of distributions that violates these properties.
- Finally, we analytically solve for the value of parameter η in the zero-inflated uniform distribution that minimizes weighted objective $w\mathbb{E}[B] + (1-w)\mathbb{E}[U]$ for any $w \in [0,1]$.

□

As shown in Figure 4, for smaller privacy budgets where $\epsilon \leq 2\ln(2)$, there is a single point on the Pareto frontier. Adding uniform noise with no inflated probability mass at 0 minimizes $\mathbb{E}[B]$ and $\mathbb{E}[U]$ simultaneously. For larger ϵ, it is possible to trade off between $\mathbb{E}[B]$ and $\mathbb{E}[U]$, achieving near-zero delay to unbatched comments. In practice, a user can decide what value of η to use based on their preferred convex combination of $\mathbb{E}[B]$ and $\mathbb{E}[U]$.

Note that our result holds for all mechanisms that add independent noise to each comment, as the delay added to comments must be $(\epsilon/2, g)$-one-sided indistinguishable to preserve privacy. Therefore, the zero-inflated uniform mechanism (Algorithm 2) is the Pareto optimal mechanism among this class of algorithms. It may be possible to add even less delay with mechanisms that can coordinate across comments and correlate noise addition. We leave this question open for future work.

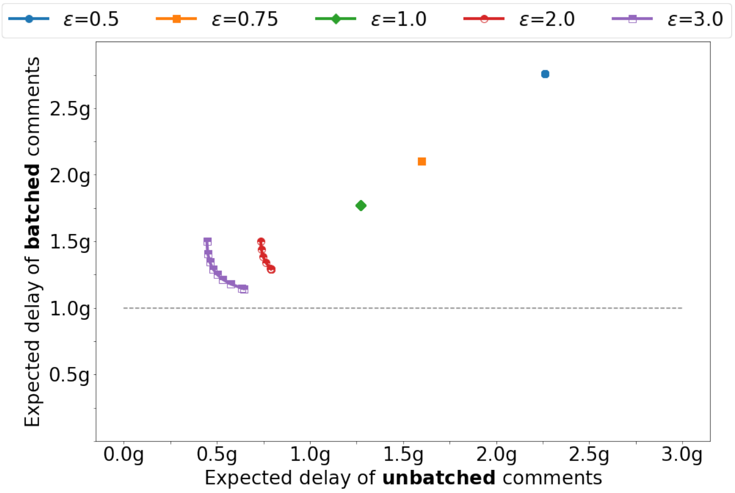


Figure 4: Pareto frontier for the expected delay added to batched and unbatched comments ($\mathbb{E}[B]$ and $\mathbb{E}[U]$) at different values of privacy parameter ϵ.

4.4 Impossibility of "Two-Sided" Differential Privacy

In the prior sections, we have characterized the privacy-utility trade-off for the one-sided relaxation of differential privacy. One might wish to obtain similar results for the standard two-sided definition of differential privacy, which would provide even stronger privacy guarantees. In this section, we prove the impossibility of guaranteeing two-sided differential privacy under the constraints of a valid comment postinging mechanism. These results motivate the further modeling assumptions on the adversary's prior knowledge about batching and attempted attacks that are used in the definition of (ϵ, g)-OSDP.

First, we recall the standard definition of two-sided differential privacy. The key difference between this definition and our one-sided Definition 3.2 is in the formulation of "neighboring" inputs. In our one-sided definition, we use an asymmetric relation for neighboring inputs where one input with an additional batched pair of comments neighbors an input with one fewer pair. This captures the notion that batching is sensitive while the absence of batching is insensitive. For two-sided DP, we will give a definition with an abstract notion of neighbors and then concretely instantiate this definition with different possible notions of neighboring inputs. Critically, we will consider *symmetric* relations for neighboring inputs in the definition of two-sided DP. This corresponds to preventing an adversary from inferring both whether batching occurred and whether batching did not occur.

Recall that $\mathcal{M}_T(A)$ denotes the output of the mechanism up to time T. Then:

Definition 4.7 (Two-Sided Differential Privacy for Batched Arrivals:)**.** For any $\epsilon \geq 0, \delta \in [0,1]$, a comment posting mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if, for any time horizon T and for any subset $S \subseteq \text{Range}(\mathcal{M}_T)$ of possible outputs of the mechanism:

$$\Pr[\mathcal{M}_T(A') \in S] \leq e^{\epsilon}\Pr[\mathcal{M}_T(A) \in S],$$

where A and A' are two *"neighboring"* sets of comment arrivals.

Now, we state our main impossibility result. We consider three natural definitions of neighboring sets of comment arrivals. The first definition adds or removes a comment from the set of comment arrivals corresponding to the notion of "unbounded" differential privacy in the literature [9]. The second definition moves a comment from being batched to unbatched in neighboring inputs by changing its timestamp, corresponding to the notion of "bounded" differential privacy in the literature [10]. Finally, the third definition restricts the second definition of neighbors further by placing a bound on how far a comment can move (which we call g), similar to the practice of constraining the domain of possible inputs to a differentially private mechanism. We show that it is not possible to guarantee privacy for any of these notions of neighbors:

Theorem 4.8 (Impossibility of Two-Sided Differential Privacy)**.** *For any of the following natural definitions of "neighboring" sets of comment arrivals, there is no two-sided differentially private, valid comment posting mechanism with delay scaling as* $o(1/\delta)$*:*

Definition of "Neighboring" Sets of Comment Arrivals		***Impossibility Result***
(1)	*Add or remove a batched comment*	*No* valid (ϵ, δ)*-DP posting mechanism for* $\epsilon < \infty, \delta < 1$
(2)	*Move a batched comment to another arrival time where it is no longer batched*	*No* valid (ϵ, δ)*-DP posting mechanism for* $\epsilon < \infty, \delta < 1$
(3)	*Move a batched comment by at most* g *units of time to another arrival time where it is no longer batched*	*For any* $D \geq 0$*, any* valid (ϵ, δ)*-DP posting mechanism delays a comment by at least* D *with probability* $\geq 1 - 2\delta\left(\frac{D}{g} + 1\right)$

The proof of the above theorem can be found in Appendix A.5. Intuitively, we cannot guarantee privacy with definition (1) of neighbors because it would require creating a fake comment since a comment that exists in one input does not exist in the adjacent input. It is not possible to satisfy privacy with definition (2) of neighbors, as a comment could move arbitrarily far in time, requiring infinite delay to be added to comments. For definition (3) of neighbors, we show that we can define a sequence of neighboring inputs such that a comment is shifted g units of time in the future on every other input in the sequence. Since the privacy guarantee must hold pairwise between each neighboring input in the sequence, the mechanism can only release comments within time D with probability of roughly $\delta D/g$ in order to make inputs that are D/g neighbors away from each other in the sequence sufficiently indistinguishable from one another.

Note that even if we considered mechanisms acting on a finite time horizon, the proof above suggests the only mechanism admitted under two-sided DP using definition (2) is the trivial mechanism that releases all comments at the end of the time period:

Corollary 4.9. *Suppose comments are known to arrive only during a finite time horizon* T *after which no more comments will arrive or be posted. Then, any valid posting mechanism that satisfies two-sided* (ϵ, δ)*-DP using Definition (2) of neighbors in Theorem 4.8 posts all comments at time* T*.*

This is both intuitively and formally sufficient for preserving privacy from timing attacks since it eliminates all timing information, but is expensive in terms of delay incurred. In particular, in the peer review setting, releasing all comments simultaneously at the end of the review period eliminates potential for replies and ongoing discussion.

It follows from the impossibility of definition (3) of neighboring sets that there is no valid comment posting mechanism satisfying differential privacy with $\delta = 0$ for this notion of neighbors, since any differentially private mechanism would violate the property that valid comment posting mechanisms eventually release all comments. Additionally, even taking $\delta > 0$, the probability of experiencing a delay longer than D only decreases *linearly* in δ and D. Typically, δ is selected to be $o(1/n)$ [11], where n is the database size—in our case, the number of comments in the observed stream. For $g = O(1)$, this implies that any mechanism satisfying a two-sided DP guarantee for g-neighboring inputs (and choosing $\delta = o(1/n)$) has a non-negligible probability of delaying comments by $\Omega(n)$.

5 Practical Considerations for Implementation

In this section, we address two important practical considerations to putting into practice our privacy formulation and algorithm. First, we provide theoretically motivated heuristics for setting the parameter g in the privacy definition. Second, we give simple extensions to the privacy model and algorithm that allow for handling the realistic setting where batched comments do not arrive all at the same exact time, but rather with a short duration in between.

5.1 Setting privacy parameters

Recall that our privacy definition includes a parameter g that captures what types of inputs can be neighbors. In particular, g bounds how far apart in time a pair of potentially-batched comments could arrive if batching had *not* taken place. In this section, we provide a heuristic for setting g in practice. We will argue that a reasonable way to set g for a given comment is as a percentile of an empirical distribution of comment inter-arrival times. For example, in a peer-reviewed conference we might set g to be the median inter-arrival time of comments at a similar prior conference. Alternatively, larger conferences commonly classify papers into tracks, so g could be chosen for each track individually. We provide more examples of setting g in practice in our experiments on Wikipedia and Bitcoin in Section 6.

First, we motivate this heuristic by modeling an adversary conducting a hypothesis test to determine if a comment was batched or not. The privacy parameters g and ϵ can be chosen based on the desired (in)efficacy of this adversary's test. A natural way to model a privacy attack is to consider an adversary—say, a meta-reviewer who submitted a paper to a conference—who suspects that a comment c made on their paper may share a reviewer with one of the papers in the set C of papers they are handling. The adversary conducts a hypothesis test to determine whether the comment they received arrived in a batch with any comment on papers in that set. Let t_1 denote the arrival time of c and let t_2 denote the arrival time of the comment in C that arrives closest in time to t_1. The adversary knows that if the comments did not arrive in a batch, then they arrived with a gap $t_2 - t_1$ following some distribution $\mathcal{D}$ (for instance, this might be the empirical distribution of comment inter-arrival times on the previous day). If the pair of comments does arrive in a batch, the adversary assumes they arrived simultaneously. Thus, the adversary wishes to distinguish between the following hypotheses:

$$\begin{aligned} &H_0 : \ t_2 - t_1 \sim \mathcal{D} && (\textit{c is not batched with any comment in } C) \\ &H_1 : \ t_1 = t_2 && (\textit{c is batched with at least one comment in } C) \end{aligned}$$

The adversary will observe the output of the mechanism and decide to either accept or reject the null hypothesis. If they reject the null hypothesis, they conclude that the comment was batched with a comment in C. Their hypothesis test is defined by "rejection region" R, or the set of outputs on which the adversary concludes that batching occurred. The quality of a given test is determined by the trade-off between its "power" and "type I error":

$$\begin{aligned} \textit{Power} &= \Pr[\mathcal{M}(S) \in R; H_1] \\ \textit{Type I Error} &= \Pr[\mathcal{M}(S) \in R; H_0] \end{aligned}$$

Similar to prior work on differential privacy [42], [20], we show that an adversary conducting a hypothesis test to determine if batching occurred will face a poor trade-off between power and type I error given an output of a mechanism that is OSDP with gap g:

Proposition 5.1. *If a mechanism $\mathcal{M}$ satisfies (ϵ, g)-OSDP, then for any comment c, set of comments C arriving with inter-arrival time distribution $\mathcal{D}$, and any hypothesis test deciding if c was batched with a consecutively arriving comment in C:*

$$\textit{Power} \leq \frac{e^{\epsilon}}{F_{\mathcal{D}}(g)}(\textit{Type I Error})$$

where $F_{\mathcal{D}}(g) = Pr[|x| \leq g; x \sim \mathcal{D}]$ is the CDF of inter-arrival times.

The proof of this proposition can be found in Appendix A.6. This interpretation of the (ϵ, g)-OSDP guarantee in terms of error rates of an attacker's hypothesis test motivates our heuristic to choose the parameter g. Previous work on timing attacks [37, 26] measures the success of attacks in terms of the trade-off between power and type I error. In particular, these works report a single number "error crossover rate," the point at which *type I error* $= 1 -$ *power*. We envision the system operator (i.e., the entity adding the delay) first specifying a tolerable error crossover rate; for example, consistent with prior work on timing attacks [37, 26], the operator might choose to tolerate an error crossover rate of 0.25. Next, the system operator should choose a privacy parameter ϵ. Since the interpretation of ϵ is similar to traditional two-sided DP, operators may use common heuristics for selecting ϵ; for instance, our operator might choose $\epsilon = 0.8$. Given these parameters, Proposition 5.1 shows how to select g to ensure that the desired error crossover rate is satisfied. In our running example, we would choose g to be the 75th percentile of the inter-arrival time distribution.

Algorithm 3 Framework for Handling Non-Simultaneous Batching

```
Input: privacy parameter $\epsilon > 0$, maximum gap $g > 0$, batching threshold $0 \leq \beta < g$, noise addition distributions $B$ and $U$
for comment arriving at time $t$ do
    Hold the comment until time $t + \beta$.
    if the same reviewer batched another comment during time $t$ to $t + \beta$ then
        Sample $d \sim B(\epsilon/2, g + \beta)$ and post the action at time $t + \beta + d$.
    else
        Post the comment at time $t + \beta + d$ where $d \sim U(\epsilon/2, g + \beta)$
    end if
end for
```

5.2 Handling Non-Simultaneous Batching

In our basic model of batching, we make the idealized assumption that all comments in a batch arrive at the same exact clock time. In practice, in many settings, batched actions will not be taken at the *exact* same time, but rather with some short delay between them. For example, it is natural for a Wikipedia editor to spend many minutes working on a revision, so revisions in a single batch may arrive with a few minutes of delay in between. Likewise, reviewers in peer review may comment on papers one after the other, leading to a short delay despite batching.

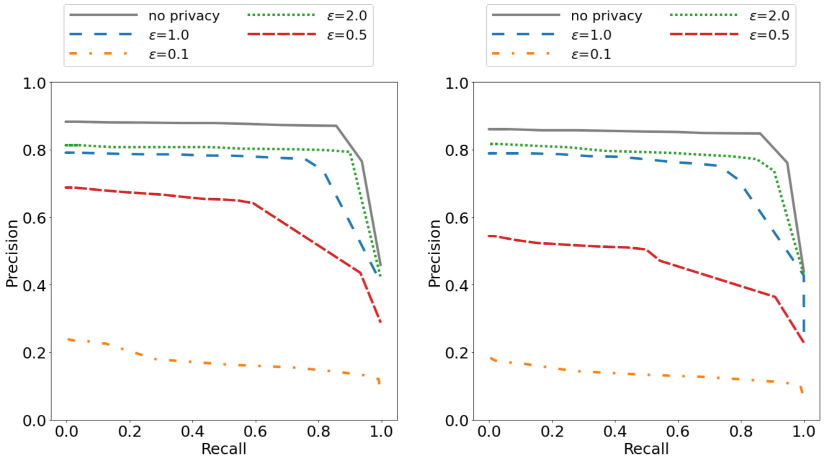
In this section, we describe a simple extension to our model and algorithm that allows us to handle non-simultaneity in practice. We introduce a new threshold β, below which we consider two comments to have been batched — if two comments come from the same reviewer within time β we consider them to have arrived in a batch. We will assume that $\beta < g$, as we wish to capture scenarios where batching leads a comment to arrive earlier than it would have without batching. We can capture this scenario by replacing the notion of neighbors in our model with the following:

Definition 5.2 (g-Neighboring Comment Arrival Sets with β-batching)**.** For $\beta < g$, a set of comment arrivals $A^{(B)}$ is g-neighboring with β-batching to set A, if $A^{(B)}$ can be obtained from A by batching together a pair of comments that arrive separately within g time units of one another in A, moving the later comment to within β of the earlier comment. Specifically, $\exists (c, t, p, r), (c', t', p', r) \in A$ such that $p \neq p'$, $0 \leq t' - t \leq g$ and $A^{(B)} = A \setminus \{c'\} \cup \{(c', t'', p', r)\}$ where $0 \leq |t'' - t| \leq \beta$.

We define privacy the same as in Definition 3.2, but with this modified notion of g-neighboring with β-batching. In what follows, we describe how we incorporate this relaxed notion of batching into our algorithm.

First, we propose a simple front-end change that can be employed in conjunction with any mechanism in our algorithmic framework of randomized delay mechanisms (Algorithm 1) if we trust users to accurately report when they will engage in batching. The solution is to ask users when they create a comment if they plan on creating more comments on their other papers within the next β units of time (and hence will generate batched comments). If the user answers affirmatively, then we treat their current comment as well as any subsequent comments they make within β time units as batched and add delay drawn from B to the batched comments. If not, we add delay from U to the unbatched comments. Here, we take (B, U) to be one-sided $(\epsilon/2, g + \beta)$-indistinguishable. Since neighboring inputs can differ on two comments with arrival times at $(t, t - \beta)$ and $(t + \beta, t + g)$ respectively, it is now necessary to add noise from $(\epsilon/2, g + \beta)$-indistinguishable distributions to preserve privacy by the same reasoning as Theorem 4.2.

In settings where we do not expect users to reliably report that they will batch tasks, we can use a simple extension to our algorithmic framework, described in Algorithm 3, where we delay all comments by an additional β units of time, using that duration to determine whether or not the comment was batched. The algorithm pays an additional β in overhead to decide whether a comment was batched or not. Privacy follows by the same reasoning as in Theorem 4.2. In general, our initial problem formulation captures the most essential features of the problem of preserving privacy in the presence of batching. As we have shown in this section, it is straightforward to extend our model to better capture the properties specific to a given application.



(a) g set to 11 minutes

(b) g set to 36 minutes

Figure 5: Accuracy in linking pairs of Wikipedia article revisions within the category "21st-century American Politicians" based on batched timing (averaged over 5 runs of the randomized privacy mechanism).

	Mean Delay				Maximum Delay			
	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$
(a) $g = 11$	118	31	20	15	229	54	33	22
(b) $g = 36$	343	83	50	35	672	152	88	56

Table 1: Mean and maximum delay (in minutes) added to Wikipedia article revisions within the category "21st-century American Politicians" for g set to the (a) 25th and (b) 50th percentile of the historical inter-arrival distribution.

6 Experiments

We conduct two sets of experiments using publicly available data on Wikipedia article revisions and Bitcoin transactions.

6.1 Wikipedia

In a dataset of revisions on all Wikipedia articles from January 1st to 31st, 2022 obtained from the WikiMedia API, we aggregate over 3.5 million article revisions (after filtering out bot accounts), averaging roughly 80 revisions per minute. Due to the high baseline rate of editing, it would be difficult for an adversary to identify that two revisions are batched without narrowing down the set of possible articles they consider. Therefore, we focus on a subset of Wikipedia revisions within which an adversary tries to link editors. One natural clustering of articles likely to contain batched revisions is by category: each article on Wikipedia is associated with a set of categories capturing the main topics covered. In the following experiments, we analyze articles belonging to the category "21-st century American Politicians." We chose this category because it contains potentially controversial political topics so editors may have privacy concerns in editing these pages. For instance, one news report describes how editors of Donald Trump's Wikipedia page (one of the pages captured in the category) "are fighting a brutal, petty battle over every word [28]." Additionally, this category receives a large number of revisions per month, yielding a sample size of 13,430 revisions. Among this set of revisions, roughly 20% were generated in a batch with another revision on a page in the same category (where we consider revisions to be batched if they arrive within 5 minutes of one another and are made by the same user). The threshold of 5 minutes captures 92% of pairs of immediately consecutive revisions by a single editor on different articles within this category.

While we do not have access to the true identities of editors who use multiple accounts, we can track all revisions made by the same account and identify when this account engages in batching. We simulate an attack where an adversary tries to link revisions to their creator on the basis of timing, while ignoring the usernames of editors. We consider a simple attack model that proves to be quite effective in the absence of any privacy-preserving mechanism. In the attack, the adversary tries to classify each pair of revisions on two different articles as either batched or unbatched. The adversary chooses a cutoff $I \geq 0$: if a pair of revisions are posted within I minutes of one another, the adversary classifies the pair as batched and concludes that the comments were made by the same person, and if not, the adversary classifies the pair as unbatched (in which case, the adversary draws no conclusion). When the adversary correctly classifies a batched pair as batched, we call this a true positive, while if the adversary incorrectly classifies an unbatched pair as batched, we call this a false positive. The adversary can trade off between false positives and true positives by choosing the value of I accordingly, with higher values of I yielding more true positives, but also more false positives, than smaller values of I. The efficacy of the attack is measured in terms of its precision and recall where precision $= \frac{\text{number of pairs correctly classified as batched}}{\text{total number of pairs classified as batched}}$ and recall $= \frac{\text{number of pairs correctly classified as batched}}{\text{total number of pairs that were batched}}$. An effective attack has simultaneously high precision and recall.

In Figure 5, we show the precision and recall of this attack under various settings of privacy parameters ϵ and g. We find that attack efficacy is quite high when no privacy mechanism is deployed — for instance, it is possible to obtain recall of 85% at a precision of 80%. We then apply the zero-inflated uniform mechanism (Algorithm 2) and measure the reduction in attack efficacy over the "no privacy" baseline. We run the mechanism with w set to 1, as this minimizes the worst-case expected delay added to any single comment in the system. Because batching is not perfectly simultaneous on Wikipedia — editors take time between making each revision — we simulate deployment of the user interface extension to Algorithm 2 described in Section 5.2 where $\beta = 5$ minutes. We set g based on the heuristic from Section 5.1 where g is a percentile of the inter-arrival distribution of revisions made in the first week of the month. We then simulate deployment of the algorithm over the last three weeks of the month. Using this method, we can set $g = 11$ minutes by choosing the 25th percentile or $g = 36$ minutes at the 50th percentile. The experiment reveals that precision and recall are significantly improved by use of the mechanism as shown in Figure 5. In terms of delay, Table 1 shows the mean and maximum delay added to comments. We provide additional results, setting g based on the 75th percentile of the inter-arrival distribution in Appendix D.1.

Thus we find that Algorithm 2 renders the privacy attack much less effective while introducing reasonable delay. For instance, taking $g = 11$ and $\epsilon = 0.5$ corresponds to an average delay of roughly 1 hour 20 minutes and maximum delay of 2.5 hours, but makes the attack substantially less accurate: the attack now achieves around 65% recall at 60% precision compared to the non-private baseline which achieves 85% recall at 80% precision. The heuristic attack used in Figure 5 may not be optimal for an adversary who has knowledge of the zero-inflated uniform mechanism, but not access to the internal randomness of the mechanism. Identifying an optimal attack is beyond the scope of this work. However, since the same noise distribution B is added to all comments that arrive in a batch, we expect the heuristic attack to perform well in expectation.

6.2 Bitcoin

In Bitcoin, we wish to protect against linkage attacks on users of Bitcoin who use multiple addresses to transmit currency to the same recipient address at the same time. We aggregate data of all confirmed transactions broadcast to the Bitcoin peer-to-peer network in the week of August 1, 2022 to August 7, 2022, consisting of approximately 250,000 transactions per day. While we cannot tie different addresses to real-world identities, for the purposes of our experiments, we consider the following proxy: we define a "batch" to have occurred when two transactions from different input addresses are sent to the same output address within 1 minute of one other. This represents a key use-case of our algorithm, wherein a person holding Bitcoin in multiple addresses wishes to draw from these different sources to complete a transfer to a single output address. After filtering for transactions originating from addresses with unusually high volume of transactions that likely represent cryptocurrency exchanges, there are about 3,000 transactions per day arriving in a batch per our definition, representing 1.2% of all transactions.

We consider a privacy attack similar to the linkage attack described in the Wikipedia application. In the Bitcoin setting, an adversary tries to identify whether pairs of transactions arrived in a batch or not. The adversary observes the times at which transactions to the same output address are broadcast to the Bitcoin P2P network and applies a threshold to the time difference between the pair to decide whether the transactions arrived in a batch. In a "basic" attack, the adversary uses a single threshold

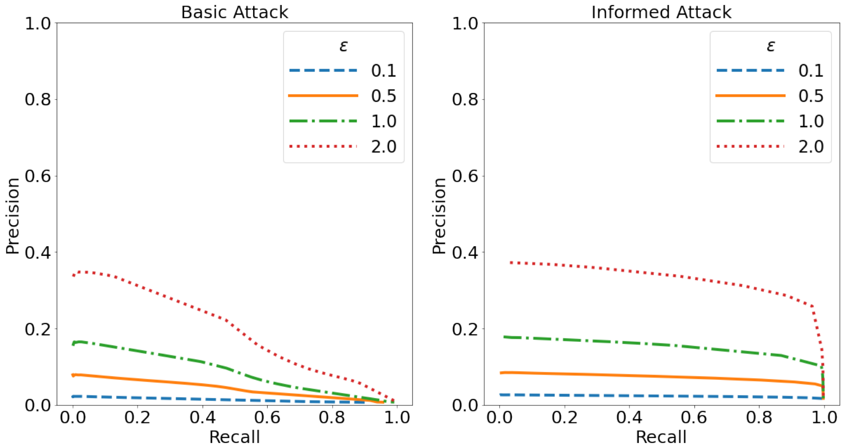


Figure 6: Performance of basic and informed linkage attacks on Bitcoin transactions when g is set to the median historical inter-arrival time for an output address.

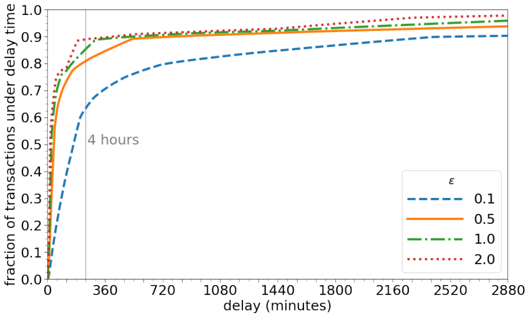


Figure 7: Cumulative distribution of delay added to batched Bitcoin transactions (averaged over 5 trials). Delay is drawn from a privacy-preserving uniform distribution with g set to the median of the inter-arrival time of transactions to an output address within the past 7 days.

for all transactions. In an "informed" attack, we assume the adversary knows the value of g that was used by the privacy mechanism for each transaction (which the mechanism may vary by output address) and sets a per-address threshold as a linear function of the g used for that address. In incorporating this additional information about the privacy mechanism, the adversary can obtain a better trade-off between false positives and true positives. We measure efficacy of the attack in terms of precision and recall. Since we define batching to occur when multiple inputs are sent to the same output address within 1 minute of each other, the adversary can observe exactly when batching occurred if no privacy mechanism is deployed and obtain a precision and recall of 100% in identifying whether transactions arrived at the same time or not (recall that in this experiment, we lack ground truth about batched transactions).

To obscure the timing of transactions, we simulate the zero-inflated uniform mechanism (Algorithm 2) to add delay to the time at which transactions are broadcast to the Bitcoin P2P network. In order to select the value of g, we estimate the inter-arrival distribution of transactions to a given output address in the prior 7 days and set g to a percentile of this distribution. In particular, in this section we use the median of the inter-arrival distribution. In Appendix D.2, we give additional results for experiments where g is set to the 25th and 75th percentile of the inter-arrival distribution. If the output address of a transaction received no other transactions in the prior 7 days, we set g to 10 minutes, as this is the baseline duration of time a Bitcoin user has to wait for a transaction to be confirmed on the blockchain. Most ($> 90\%$) unbatched transactions are sent to output addresses with no recent transaction history, so we use the value of $g = 10$ for these transactions. However, roughly 80% of batched transactions are sent to output addresses with transaction history.

The use of Algorithm 2, with g set per output address, makes it difficult to identify whether transactions to the same output address arrived at the same time. For $\epsilon = 1$, even the informed attack has precision of only 20% at high recall. The basic attack performs much worse, indicating that an adversary needs to incorporate additional information about baseline inter-arrivals of transactions in order to accurately identify batching.

This improvement in privacy comes at the expense of added latency. In Figure 7, we show the cumulative density function of delay added to batched Bitcoin transactions averaged over 5 samples from the privacy-preserving uniform distribution. In general, we can add delay of less than 4 hours to most transactions. For the setting of $\epsilon = 1$, the mechanism adds delay of under 2 hours to 70% of transactions. While this is slower than a Bitcoin transaction when no privacy mechanism is used, it is still substantially faster than many other means of transferring money, like wire transfers. As such, privacy-sensitive users could realistically deploy this algorithm in their Bitcoin wallets to protect the unlinkability of their transactions.

7 Discussion

This work introduces the problem of anonymity compromise caused by task batching in pseudonymous forums. We propose defenses and theoretically and empirically establish the efficacy of these solutions.

Global Ordering. We find in empirical evaluations of Wikipedia data that the zero-inflated uniform mechanism is likely to release article revisions in a different order than they arrived. In our experiments, at reasonable settings of the privacy parameters, roughly 10% of revisions were re-ordered within an article. This can create confusion when there are dependencies between article revisions. A similar problem arises in peer review, where comments may respond to one another. In Appendix C, we discuss a privacy-preserving queue-based mechanism that outputs delayed comments in the same order in which they arrived. While this algorithm does not satisfy the (ϵ, g)-OSDP guarantee, it satisfies a different relaxation of differential privacy. An open question is whether the uniform zero-inflated mechanism can be extended to enforce ordering constraints for an appropriate privacy guarantee.

Partial adoption. In actual deployments, many participants may be privacy-insensitive and opt out of additional protections that preserve anonymity at the cost of increased delay. Our privacy guarantee holds for any pair of events where each event uses the delay mechanism independently of what other users choose to do. So, for a single user who deploys the zero-inflated uniform mechanism on all events, it will be difficult for an adversary to tell whether any pair of their events is batched. However, there may be additional amplification of privacy that comes from widespread usage and permits lower setting of g and ϵ with the same privacy guarantees in practice. Quantifying the dependence of adoption rate on privacy guarantees is an interesting open question.

Acknowledgments

This research was approved by the CMU Institutional Review Board (IRB). This work was supported in parts by NSF grants CIF: 1763734, 1705007 and RI: 2200410, ONR grant N000142212181, and the Air Force Office of Scientific Research grant FA9550-21-1-0090. The authors gratefully acknowledge the support of the Bill & Melinda Gates Foundation and the Sloan Foundation.

References

[1] Bitcoin Wiki. https://en.bitcoin.it/wiki/Transaction#Input. Accessed on November 3, 2022. 1

[2] Bitcoin Wiki. https://en.bitcoin.it/wiki/Common-input-ownership_heuristic. Accessed on November 3, 2022. 1

[3] Bitcoin Wiki. Address reuse. https://en.bitcoin.it/wiki/Address_reuse, 2021. Accessed on April 21, 2021. 1

[4] C. Blank, S. Zaman, A. Wesley, P. Tsiamyrtzis, D. R. Da Cunha Silva, R. Gutierrez-Osuna, G. Mark, and I. Pavlidis. *Emotional Footprints of Email Interruptions*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2020. 1

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 3, A.4.2

[6] W. Ding, G. Kamath, W. Wang, and N. B. Shah. Calibration with privacy in peer review. In *ISIT*, 2022. 2

[7] W. Ding, N. B. Shah, and W. Wang. On the privacy-utility tradeoff in peer-review data analysis. In *AAAI Privacy-Preserving Artificial Intelligence (PPAI-21) workshop*, 2020. 2

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. `https://svn-archive.torproject.org/svn/projects/design-paper/tor-design.pdf`, 06 2004. 4.3

[9] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 4.4

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. 1, 2, 4, 4.4

[11] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. 4.4

[12] P. Flach, S. Spiegler, B. Golénia, S. Price, J. Guiver, R. Herbrich, T. Graepel, and M. Zaki. Novel tools to streamline the conference review process: Experiences from SIGKDD'09. *SIGKDD Explor. Newsl.*, 11(2):63–67, May 2010. 2

[13] A. Forte, N. Andalibi, and R. Greenstadt. Privacy, anonymity, and perceived risk in open collaboration: A study of tor users and wikipedians. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, page 1800–1811, New York, NY, USA, 2017. Association for Computing Machinery. 1

[14] H. Ge, M. Welling, and Z. Ghahramani. A Bayesian model for calibrating conference review scores. Manuscript, 2013. Available online `http://mlg.eng.cam.ac.uk/hong/unpublished/nips-review-model.pdf` Last accessed: April 4, 2021. 2

[15] Q. Geng and P. Viswanath. The optimal mechanism in differential privacy. In *2014 IEEE International Symposium on Information Theory*, pages 2371–2375, 2014. 2, 4, 4.2, (2), 6, 4.3, (2), A.3, A.4.2, (iii)

[16] O. Javidbakht and P. Venkitasubramaniam. Delay anonymity tradeoff in mix networks: Optimal routing. *IEEE/ACM Transactions on Networking*, 25(2):1162–1175, 2017. 2

[17] S. Jecmen, N. B. Shah, F. Fang, and V. Conitzer. Tradeoffs in preventing manipulation in paper bidding for reviewer assignment. In *ICLR workshop on ML Evaluation Standards*, 2022. 2

[18] S. Jecmen, H. Zhang, R. Liu, N. B. Shah, V. Conitzer, and F. Fang. Mitigating manipulation in peer review via randomized reviewer assignments. In *NeurIPS*, 2020. 2

[19] S. Kadloor, P. Venkitasubramaniam, and N. Kiyavash. Preventing timing analysis in networks: A statistical inference perspective. *IEEE Signal Processing Magazine*, 30(5):76–85, 2013. 2

[20] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017. 5.1

[21] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008. 4.3

[22] I. Kotsogiannis, S. Doudalis, S. Haney, A. Machanavajjhala, and S. Mehrotra. One-sided differential privacy. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 493–504, 2020. 1, 2, 3

[23] K. Kushlev and E. W. Dunn. Checking email less frequently reduces stress. *Computers in Human Behavior*, 43:220–228, 2015. 1

[24] K. Kushlev and E. W. Dunn. Stop checking email so often, Jan 2015. 1

[25] C. J. Lee. Commensuration bias in peer review. *Philosophy of Science*, 82(5):1272–1283, 2015. 2

[26] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright. Timing attacks in low-latency mix systems. In A. Juels, editor, *Financial Cryptography*, pages 251–265, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 2, 5.1

[27] M. L. Littman. Collusion rings threaten the integrity of computer science research. *Communications of the ACM*, 64(6):43–44, 2021. 2

[28] A. Mak. Inside the brutal, petty war over donald trump's wikipedia page, May 2019. 6.1

[29] G. Mark, S. T. Iqbal, M. Czerwinski, P. Johns, A. Sano, and Y. Lutchyn. Email duration, batching and self-interruption: Patterns of email use on productivity and stress. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 1717–1728, 2016. 1

[30] F. K. Maurer, T. Neudecker, and M. Florian. Anonymous coinjoin transactions with arbitrary values. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 522–529. IEEE, 2017. 3

[31] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: Characterizing payments among men with no names. *Commun. ACM*, 59(4):86–93, mar 2016. 1

[32] K. Moore. How to improve productivity with time batching. Monday blog, Nov 2021. Accessed on April 25, 2022. 1

[33] M. Murphy. If you haven't tried time batching, you'll be shocked at how quickly it improves your productivity and happiness. Forbes Magazine, 2021. Accessed on April 25, 2022. 1

[34] R. Noothigattu, N. Shah, and A. Procaccia. Loss functions, axioms, and peer review. *Journal of Artificial Intelligence Research*, 2021. 2

[35] M. Roos, J. Rothe, J. Rudolph, B. Scheuermann, and D. Stoyan. A statistical approach to calibrating the scores of biased reviewers: The linear vs. the nonlinear model. In *Multidisciplinary Workshop on Advances in Preference Handling*, 2012. 2

[36] N. B. Shah. An overview of challenges, experiments, and computational solutions in peer review (extended version). https://www.cs.cmu.edu/~nihars/preprints/SurveyPeerReview.pdf Shorter version published in the Communications of the ACM., 2022. 2

[37] V. Shmatikov and M.-H. Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In D. Gollmann, J. Meier, and A. Sabelfeld, editors, *Computer Security – ESORICS 2006*, pages 18–33, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2, 5.1

[38] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich. Stadium: A distributed metadata-private messaging system. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, page 423–440, New York, NY, USA, 2017. Association for Computing Machinery. 2

[39] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, page 137–152, New York, NY, USA, 2015. Association for Computing Machinery. 2

[40] T. N. Vijaykumar. Potential organized fraud in on-going asplos reviews, Nov 2020. 2

[41] J. Wang and N. B. Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *AAMAS*, 2019. 2

[42] L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010. 5.1

[43] Wikipedia Terms of Service. Sockpuppetry - alternative accounts. Wikipedia TOS, 2022. Accessed on April 28,2022. 1

[44] R. Wu, C. Guo, F. Wu, R. Kidambi, L. van der Maaten, and K. Weinberger. Making paper reviewing robust to bid manipulation attacks. In *ICML*, 2021. 2

Appendices

In Appendix A, we present proofs of results that were claimed but not proven in the main text. In Appendix B, we detail the methods used to measure the prevalence of batching and resulting deanonymization risk in peer review. In Appendix C, we describe an alternative privacy formulation that gives rise to a queue-based mechanism which preserves the order of comment arrivals. Finally, in Appendix D we give additional empirical results of experiments on Wikipedia and Bitcoin for additional parameter settings not presented in the main text.

A Proofs

In this section we present proofs of results that were claimed but not proven in the main text. Throughout we will use the following notation to denote element-wise addition and subtraction for a set: for any $S \subseteq \mathbb{R}, t \in \mathbb{R}$, we define $S - t := \{s - t | s \in S\}$.

A.1 Proof of Theorem 4.2 (Privacy of Random Delay Mechanisms with Indistinguishable Noise-Addition Distributions)

First, we prove a general necessary and sufficient condition to guarantee (ϵ, g)-OSDP when a mechanism adds independent noise from distributions B and U to batched and unbatched comments respectively.

Lemma A.1. *Let $\mathcal{M}$ be any mechanism that adds independent random delay to comments with delay drawn from distribution B for batched comments and U for unbatched comments. Then, $\mathcal{M}$ is (ϵ, g)-OSDP if and only if $\forall S, S' \in \mathbb{R}$ and $\forall t_0 \in [0, g]$ it holds that*

$$Pr[B \in S]Pr[B \in S'] \leq e^{\epsilon}Pr[U \in S]Pr[U \in S' - t_0].$$

Proof. First, let $\mathcal{M}$ be any (ϵ, g)-OSDP mechanism adding independent random delay to comments with delay drawn from distributions B and U. Suppose for the sake of contradiction that there exists some $S, S' \in \mathbb{R}$ and $t_0 \in [0, g]$ such that $\Pr[B \in S]\Pr[B \in S'] > e^{\epsilon}\Pr[U \in S]\Pr[U \in S' - t_0]$. Let A and $A^{(B)}$ be g-neighboring inputs differing in the arrival time of a single comment. In $A^{(B)}$, a pair of comments c and c' arrive in a batch at time 0. In A, comment c arrives unbatched at time 0 and comment c' arrives unbatched at time t_0. All other comments arrive at the same times in A and $A^{(B)}$. Let O denote the set of possible outputs where c is posted at a time in S and c' is posted at a time in S' and all other comments are posted at any time in $\mathbb{R}$. Then, since delay is added independently to each comment: $\Pr[\mathcal{M}(A^{(B)}) \in O] = \Pr[B \in S]\Pr[B \in S']$ and $\Pr[\mathcal{M}(A) \in O] = \Pr[U \in S]\Pr[U \in S' - t_0]$. However, by the initial assumption $\Pr[B \in S]\Pr[B \in S'] > e^{\epsilon}\Pr[U \in S]\Pr[U \in S' - t_0]$ contradicting the (ϵ, g)-OSDP of $\mathcal{M}$.

Now, we prove the other direction. Let $\mathcal{M}$ be any mechanism adding independent random delay to comments with delay drawn from distributions B and U such that $\forall S, S' \in \mathbb{R}$ and $\forall t_0 \in [0, g]$ it holds that

$$\text{Pr}[B \in S]\text{Pr}[B \in S'] \leq e^{\epsilon}\text{Pr}[U \in S]\text{Pr}[U \in S' - t_0]. \tag{1}$$

Note that taking $S = \mathbb{R}$, $\text{Pr}[B \in S] = \text{Pr}[U \in S] = 1$ so it must hold that $\forall S' \in \mathbb{R}, t_0 \in [0, g]$

$$\text{Pr}[B \in S'] \leq e^{\epsilon}\text{Pr}[U \in S' - t_0]. \tag{2}$$

Let A and $A^{(B)}$ be any g-adjacent comment arrival sets. Let c, c' denote the pair of comments that arrive in a batch together in $A^{(B)}$ but do not arrive in a batch together in A. In A, the two comments both arrive at time t, while in $A^{(B)}$ comment c arrives at time t and comment c' arrives at time $t + t_0$ with $t_0 \in [0, g]$ by the definition of g-adjacency. All other comments arrive at the same time in A and $A^{(B)}$. Let o and o' denote the randomized times at which the mechanism $\mathcal{M}$ releases comments c and c' respectively.

Let O be any set of possible outputs of the mechanism during time horizon T and let S denote the values of o in O and S' the values of o' in S'. Then, because $\mathcal{M}$ adds noise independently to each comment and all comments other than c and c' are equivalent in A and $A^{(B)}$, the probabilities factor as

$$\text{Pr}[\mathcal{M}(A^{(B)}) \in O] = k\text{Pr}[o \in S; A^{(B)}]\text{Pr}[o' \in S'; A^{(B)}]$$
$$\text{and } \text{Pr}[\mathcal{M}(A) \in O] = k\text{Pr}[o \in S; A^{(B)}]\text{Pr}[o' \in S'; A^{(B)}]$$

where k captures the probability that all comments other than S and S' are posted at post times in the set of outputs O.

Consider two cases for the size of the batch in which c and c' arrive in $A^{(B)}$. First, suppose the batch has 2 comments. Then, on input A, both comments are unbatched, so for some $t_0 \in [0, g]$ and $d_U, d'_U \overset{\text{iid}}{\sim} U$: $o = t + d_U$ and $o' = t + t_0 + d'_U$. On input $A^{(B)}$, $o = t + d_B$ and $o' = t + d'_B$ where $d_B, d'_B \overset{\text{iid}}{\sim} B$. Therefore,

$$\text{Pr}[\mathcal{M}(A^{(B)}) \in O] = k\text{Pr}[t + d_B \in S]\text{Pr}[t + d'_B \in S'] = k\text{Pr}[B \in S - t]\text{Pr}[B \in S' - t]$$
$$\text{and } \text{Pr}[\mathcal{M}(A) \in O] = k\text{Pr}[t + d_U \in S]\text{Pr}[t + t_0 + d'_U \in S'] = k\text{Pr}[U \in S - t]\text{Pr}[U \in S' - t - t_0].$$

So, by Inequality (1) we have $\text{Pr}[\mathcal{M}(A^{(B)}) \in O] \leq e^{\epsilon}\text{Pr}[\mathcal{M}(A) \in O]$. In the case where the batch containing c and c' has only two comments, the probability of the output on input $A^{(B)}$ remains the same, but on A, $o = t + d_B$, since comment c is still treated as batched, so $\text{Pr}[\mathcal{M}(A) \in O] = k\text{Pr}[B \in S - t]\text{Pr}[U \in S' - t - t_0]$ and by Inequality (2), $\text{Pr}[\mathcal{M}(A^{(B)}) \in O] \leq e^{\epsilon}\text{Pr}[\mathcal{M}(A) \in O]$ so $\mathcal{M}$ is (ϵ, g)-OSDP. □

Let $\mathcal{M}$ be any mechanism adding independent delay from B and U to batched and unbatched comments respectively, where B and U are $(\epsilon/2, g)$-one-sided indistinguishable distributions. Consider any $S, S' \in \mathbb{R}$ and $t_0 \in [0, g]$. Then, by indistinguishability $\text{Pr}[B \in S] \leq e^{\epsilon/2}\text{Pr}[U \in S]$ and $\text{Pr}[B \in S'] \leq e^{\epsilon/2}\text{Pr}[U \in S - t_0]$, so $\text{Pr}[B \in S]\text{Pr}[B \in S'] \leq e^{\epsilon}\text{Pr}[U \in S]\text{Pr}[U \in S' - t_0]$. Applying Lemma A.1 we conclude that $\mathcal{M}$ is (ϵ, g)-OSDP completing the proof.

A.2 Proof of Theorem 4.3 (Privacy-preserving distributions)

First, note that by the definition of one-sided indistinguishability (Definition 4.1), if B and U have probability density functions b and u respectively then B and U are (ϵ, g)-one-sided indistinguishable if and only if $\frac{b(t)}{u(t-t_0)} \leq e^{\epsilon} \quad \forall t \geq 0, t_0 \in [0, g]$ for which $b(t) > 0$. So,

(1) Exponential: for any $t < g$, $b(t) = 0$ while for any $t \geq g, t_0 \in [0, g]$ it holds that $\frac{b(t)}{u(t-t_0)} = \frac{\exp\{-\epsilon(t-g)/g\}}{\exp\{-\epsilon(t-t_0)/g)\}} \leq \frac{\exp\{-\epsilon(t-g)/g\}}{\exp\{-\epsilon t/g)\}} = e^{\epsilon}$.

(2) Staircase: by indistinguishability of the staircase distribution proven in [15].

(3) Uniform: for any $t \in [g, \frac{1}{1-e^{-\epsilon}} g]$, $t_0 \in [0, g]$ we have that $\frac{b(t)}{u(t-t_0)} = \frac{(1-e^{-\epsilon})/(e^{-\epsilon} g)}{(1-e^{-\epsilon})/g} = e^{\epsilon}$ and $b(t) = 0$ for all other values of t so $b(t) = 0 \leq e^{\epsilon} u(t - t_0)$ for all other values of t.

(4) Zero-inflated uniform: for any closed interval $[a, b] \subset [0, g)$ or $[a, b] \subset [\frac{\eta}{\eta - e^{-\epsilon}} g, \infty)$, we have that $\Pr[B \in S] = 0$. For any interval $[a, b] \subseteq [g, \frac{\eta}{\eta - e^{-\epsilon}} g]$, we have that $\Pr[B \in [a, b]] = (b - a)\frac{p - e^{-\epsilon}}{e^{-\epsilon} g}$ while for any $t_0 \in [0, g]$ we have that $\Pr[U \in [a - t_0, b - t_0]] \geq (b - a)\frac{p - e^{-\epsilon}}{g}$ so the ratio $\frac{\Pr[B \in S]}{\Pr[U \in S - t_0]}$ is bounded by e^{ϵ} for any measurable set S.

A.3 Proof of Theorem 4.4 (Optimal choice of parameters for exponential and staircase distributions)

First, in the following two lemmas we argue that the offset terms a_B and a_U must be set to $a_B = g$ and $a_U = 0$ in any expectation-minimizing pair of distributions that guarantees privacy.

Lemma A.2. *Let B, U be non-negative noise-addition distributions that guarantee (ϵ, g)-OSDP when used in Algorithm 1 where $B = a_B + B_0$ and $U = a_U + U_0$ for constants $a_B, a_U > 0$ and random variables B_0 and U_0 with support $[0, \infty)$. Then, $a_B - a_U \geq g$.*

Proof. By Lemma A.1, in order for privacy to hold it must be that $\forall S, S' \subseteq \text{support}(B), t_0 \in [0, g]$:

$$\frac{\Pr[B \in S]\Pr[B \in S']}{\Pr[U \in S]\Pr[U \in S' - t_0]} \leq e^{\epsilon}.$$

so taking $t_0 = g$ and $B = a_B + B_0$ and $U = a_U + U_0$ we have that

$$\frac{\Pr[B_0 \in S - a_B]\Pr[B_0 \in S' - a_B]}{\Pr[U_0 \in S - a_U]\Pr[U_0 \in S' - (a_U + g)]} \leq e^{\epsilon}.$$

Suppose for the sake of contradiction that $a_B < a_U + g$. Then, taking $S = S' = [a_B, a_U + g)$ we have that $\Pr[B_0 \in S' - a_B] = \Pr[B_0 \in [0, a_U + g - a_B)] > 0$, but $\Pr[U_0 \in S' - (a_U + g)] = \Pr[U_0 \in [a_B - a_U - g, 0)] = 0$ so the likelihood ratio is unbounded yielding a contradiction. □

Lemma A.3. *Let B, U be non-negative noise-addition distributions that guarantee (ϵ, g)-OSDP when used in Algorithm 1 where $B = a_B + B_0$ and $U = a_U + U_0$ for constants $a_B, a_U > 0$ and random variables B_0 and U_0 where either B_0 and U_0 are both exponential random variables or staircase random variables. Then, $B' = g + B_0$ and $U' = 0 + U_0$ guarantee (ϵ, g)-OSDP when used in Algorithm 1.*

Proof. Let $B = a_B + B_0$ and $U = a_U + U_0$ be distributions that satisfy (ϵ, g)-OSDP when used in Algorithm 1. First, define $B' = B - a_U$ and $U' = U - a_U$. Note that by Lemma A.2, $a_B > a_U$ so B' is still non-negative. Since we both random variables are shifted by the same constant offset, B' and U' still satisfy the sufficient condition to guarantee privacy in Lemma A.1. Now, suppose that $a_U = 0$ and $a_B > g$. Note that both the staircase distribution or the exponential distribution have monotonically decreasing probability density functions above 0 so $\Pr[B_0 \in S - a_B] \geq \Pr[B_0 \in S - g]$. Therefore, setting $B' = g + B_0$ the sufficient condition for privacy in Lemma A.1 still holds. □

Now, taking $B = g + D$ and $U = D$, by Lemma A.1, distribution D must satisfy the condition that $\forall S, S' \subseteq [g, \infty), t_0 \in [0, g)$

$$\frac{\Pr[D \in S - g]\Pr[D \in S' - g]}{\Pr[D \in S]\Pr[D \in S' - t_0]} \leq e^{\epsilon}.$$

Taking $t_0 = 0$ and $S = S'$, the privacy constraint requires that $\forall S \subseteq [g, \infty)$:

$$\frac{\Pr[D \in S - g]}{\Pr[D \in S]} \leq e^{\epsilon/2}. \tag{3}$$

So, if D is an exponential distribution with rate parameter λ, then $\forall x \in [g, \infty)$

$$\frac{\lambda \exp\{-\lambda(x - g)\}}{\lambda \exp\{-\lambda x\}} = \exp\{\lambda g\} \leq \exp\{\epsilon/2\}$$

Then, the expectation of B and U is minimized by taking $\lambda = \frac{\epsilon}{2g}$.

If D is a staircase distribution, it follows from the proof of optimality in [15] (Theorem 4), that the staircase distribution with parameters $(\epsilon', \Delta, \gamma)$ set to $\epsilon' = \epsilon/2$, $\Delta = g$ and $\gamma = \frac{1}{1+e^{\epsilon/2}}$ respectively is optimal in minimizing the expectation of D while respecting Inequality (3) completing the proof.

A.4 Proof of Theorem 4.6 (Pareto frontier)

The proof will proceed in three parts. First, in Section A.4.1 we argue that we can restrict attention to distributions B and U such that B and U are $(\epsilon/2, g)$-one-sided indistinguishable. Second, in Section A.4.2, we prove that among $(\epsilon/2, g)$-one-sided indistinguishable distributions any Pareto optimal pair of distributions must be zero-inflated uniform distributions. Finally, in Section A.4.3 we derive the optimal choice of parameters of the zero-inflated uniform distribution as a function of privacy parameters ϵ, g and choice of weighted utility function $w\mathbb{E}[B] + (1-w)\mathbb{E}[U]$.

A.4.1 Restricting attention to $(\epsilon/2, g)$-one-sided indistinguishable distributions

We being by arguing that we can restrict attention to finding optimal noise addition distributions B, U such that B and U are $(\epsilon/2, g)$-one-sided indistinguishable distributions (Definition 4.1) and then use these distributions within the framework of Algorithm 1 to design an optimal mechanism.

Lemma A.4. *Let $\mathcal{M}$ be any valid (ϵ, g)-OSDP comment posting mechanism that adds independent noise drawn from distributions B and U to batched and unbatched comments respectively. Then, (B, U) must be (ϵ, g)-one-sided indistinguishable.*

Proof. By Lemma A.1, in order for privacy to hold for a mechanism that adds independent noise drawn from distributions B and U respectively, it must be that $\forall S, S' \subseteq \mathbb{R}$ such that $\Pr[B \in S] > 0$ and $\Pr[B \in S'] > 0$ and $\forall t_0 \in [0, g]$:

$$\frac{\Pr[B \in S]\Pr[B \in S']}{\Pr[U \in S]\Pr[U \in S' - t_0]} \leq e^{\epsilon}.$$

Then, taking $S = \mathbb{R}$, $\frac{\Pr[B \in S]}{\Pr[U \in S]} = 1$, so $\frac{\Pr[B' \in S']}{\Pr[U' \in S' - t_0]} \leq e^{\epsilon}\ \forall S' \subseteq \mathbb{R}, t_0 \in [0, g]$. □

Note that Algorithm 1 (of which optimal Algorithm 2 is an instance) adds noise from $(\epsilon/2, g)$-indistinguishable distributions, which is a stronger condition than requiring (ϵ, g)-indistinguishable distributions. We will prove below (in Lemma A.12) that for any Pareto optimal (ϵ, g)-indistinguishable distributions (B, U), U must be monotonically non-increasing above 0. It follows that the distributions must be $(\epsilon/2, g)$-indistinguishable in order for Algorithm 1 to be (ϵ, g)-OSDP:

Lemma A.5. *Let $\mathcal{M}$ be any valid (ϵ, g)-OSDP comment posting mechanism that adds independent noise drawn from distributions B and U to batched and unbatched comments respectively where U is monotonically non-increasing (above 0). Then, (B, U) must be $(\epsilon/2, g)$-one-sided indistinguishable (Definition 4.1).*

Proof. By Lemma A.1, it must be that $\forall S, S' \subseteq \mathbb{R}$ such that $\Pr[B \in S] > 0$ and $\Pr[B \in S'] > 0$ and $\forall t_0 \in [0, g]$:

$$\frac{\Pr[B \in S]\Pr[B \in S']}{\Pr[U \in S]\Pr[U \in S' - t_0]} \leq e^{\epsilon}.$$

Taking $S = S'$ and $t_0 = 0$ gives $\frac{\Pr[B \in S]}{\Pr[U \in S]} \leq e^{\epsilon/2}\ \forall S \subseteq \mathbb{R}$. Since U is non-increasing, $\Pr[U \in S - t_0] \geq \Pr[U \in S]$ for $t_0 \geq 0$, so $\frac{\Pr[B \in S]}{\Pr[U \in S - t_0]} \leq e^{\epsilon/2}$ as well and B and U are $(\epsilon/2, g)$-one-sided indistinguishable. □

A.4.2 Pareto optimal distributions

The main portion of this proof characterizes Pareto optimal distributions (B, U) such that B and U are (ϵ, g)-one-sided indistinguishable. From Section A.4.1, we can then choose $(\epsilon/2, g)$-indistinguishable distributions for use in Algorithm 1 to obtain an optimal mechanism.

Let $\mathcal{P}_{\epsilon,g}$ denote the set of all pairs of (ϵ, g)-one-sided indistinguishable distributions (Definition 4.1). To derive the Pareto frontier of $\mathcal{P}_{\epsilon,g}$, we follow the high-level approach of [15], which derives the optimal *two-sided* differential privacy noise-addition distribution. The proof proceeds by showing that if B and U are (ϵ, g)-one-sided indistinguishable distributions added to batched and unbatched comments respectively, then:

1. B and U can be approximated arbitrarily well by a random variable defined by an appropriately chosen piece-wise constant probability density function.
2. We derive various properties of Pareto optimal B and U by showing that we can shift probability mass around in the piece-wise constant approximations to B and U, such that we decrease expected delay while maintaining indistinguishability. In particular, we show that B must place 0 probability mass below g and any Pareto optimal B must be monotonically non-increasing above g. We show that U is uniquely defined by B to put as little probability mass at each point as possible to maintain indistinguishability with B and put any excess probability mass at 0. We then prove that these properties imply that the zero-inflated uniform distribution is Pareto optimal.

For a random variable X and for any positive integer $i > 0$, define a random variable X_i that approximates X where X_i has probability density function $f_i^{(X)}(\cdot)$ with constant density over intervals of length $\frac{g}{i}$:

$$f_i^{(X)}(t) = \begin{cases} \frac{\Pr\left(X \in \left[k\frac{g}{i}, (k+1)\frac{g}{i}\right)\right)}{\frac{g}{i}} & \text{if } t \in [k\frac{g}{i}, (k+1)\frac{g}{i}) \text{ for } k \in \mathbb{N} \\ 0 & \text{if } t < 0. \end{cases} \tag{4}$$

Given $(B, U) \in \mathcal{P}_{\epsilon,g}$, for any positive integer $i > 0$ define (B_i, U_i) to be the random variables with probability density functions $f_i^{(B)}(\cdot)$ and $f_i^{(U)}(\cdot)$ taken to be the step-function approximations to B and U defined in Equation (3). Since the probability density function of each distribution is piece-wise constant, we define a "probability density sequence" of each distribution ($\{b_k^{(i)}\}_{k=0}^{\infty}$ and $\{u_k^{(i)}\}_{k=0}^{\infty}$ respectively) to be the sequence of values of the pdf for each constant interval of length g/i. For instance, $b_0^{(i)}$ corresponds to the constant probability density for values in range 0 to g/i while $b_i^{(i)}$ corresponds to the probability density over range g to $(g+1)/i$.

Lemma A.6 (Piecewise Constant Approximation)**.** *For any $B, U \in \mathcal{P}_{\epsilon,\gamma}$ and $i \in \mathbb{N}$ the following properties hold for piece-wise constant approximations (B_i, U_i) to (B, U) with probability density functions $f_i^{(B)}$ and $f_i^{(U)}$ respectively:*

(i) (Valid Probability Distributions) $f_i^{(B)}$ and $f_i^{(U)}$ are non-negative functions that integrate to 1*.*

(ii) (Indistinguishability) $(B_i, U_i) \in \mathcal{P}_{\epsilon,\gamma}$.

(iii) (Convergence of Expected Value) $\lim_{i\to\infty}(\mathbb{E}[B_i], \mathbb{E}[U_i]) = (\mathbb{E}[B], \mathbb{E}[U])$.

Proof. We prove each claim separately:

(i) For any random variable X with approximation X_i we have

$$\int_0^{\infty} f_i^{(X)}(t)dt = \sum_{k=0}^{\infty} \int_{\left[\frac{kg}{i}, \frac{(k+1)g}{i}\right)} f_i^{(X)}(t)dt = \sum_{k=0}^{\infty} \Pr(X \in [\tfrac{kg}{i}, \tfrac{(k+1)g}{i})) = 1.$$

(ii) For any $\ell \in \{0, \dots, \min(i, k)\}$:

$$\frac{b_k^{(i)}}{u_{k-\ell}^{(i)}} = \frac{\Pr(B \in [kg/i, (k+1)g/i])}{\Pr(U \in [(k-\ell)g/i, (k-\ell+1)g/i])} \le e^{\epsilon}$$

by indistinguishability of B and U and since the interval in the denominator is the same length interval as the numerator shifted by at most g to the left. Hence, for any $t \in [0, \infty), t_0 \in [0, g]$: $\frac{B_i(t)}{U_i(t-t_0)} \le e^{\epsilon}$ so $(B_i, U_i) \in \mathcal{P}_{\epsilon,g}$.

(iii) In [15] Lemma 19 in Appendix B proves that for any random variable X and approximation X_i defined as above, $\lim_{i\to\infty}\mathbb{E}[X_i] = \mathbb{E}[X]$. So, $\lim_{i\to\infty}(\mathbb{E}[B_i], \mathbb{E}[U_i]) = (\lim_{i\to\infty}\mathbb{E}[B_i], \lim_{i\to\infty}\mathbb{E}[U_i]) = (\mathbb{E}[B], \mathbb{E}[U])$.

□

It follows from from parts (ii) and (iii) of Lemma A.6 that

Corollary A.7. *For any fixed* $[w_B, w_U] \in [0,1]^2$ *with* $w_B + w_U = 1$*:*

$$\inf_{(B_i,U_i)\in\bigcup_{i=1}^{\infty}\mathcal{P}_{\epsilon,g}^{(i)}} w_B\mathbb{E}[B_i] + w_U\mathbb{E}[U_i] = \inf_{(B,U)\in\mathcal{P}_{\epsilon,g}} w_B\mathbb{E}[B] + w_U\mathbb{E}[U].$$

Now, we show that deriving the Pareto frontier of $\mathcal{P}_{\epsilon,g}$ is equivalent to optimizing any weighted sum of $\mathbb{E}[B]$ and $\mathbb{E}[U]$ because the feasible region is convex. Therefore, we can focus on characterizing $\mathcal{P}_{\epsilon,g}^{(i)}$ that are optimal for the weighted sum objective and take the limit as $i \to \infty$ to derive the entire Pareto frontier of $\mathcal{P}_{\epsilon,g}$.

Lemma A.8. *If* (B, U) *is Pareto optimal, then it minimizes some weighted sum of* $\mathbb{E}[B]$ *and* $\mathbb{E}[U]$*:* $\exists(w_B, w_U) \in [0,1]^2$ *with* $w_B + w_U = 1$ *such that*

$$(B,U) \in \operatorname*{arg\,min}_{(B',U')\in\mathcal{P}_{\epsilon,g}} w_B\mathbb{E}[B'] + w_U\mathbb{E}[U'].$$

Proof. We argue that the feasible region $\{(\mathbb{E}[B], \mathbb{E}[U]) \mid (B,U) \in \mathcal{P}_{\epsilon,g}\}$ is convex. Take $(B_1, U_1), (B_2, U_2) \in \mathcal{P}_{\epsilon,g}$ with $E_1 = (\mathbb{E}[B_1], \mathbb{E}[U_1])$ and $E_2 = (\mathbb{E}[B_2], \mathbb{E}[U_2])$. For any $p \in [0,1]$ define random variable B_3 to be the random variable that samples B_1 with probability p and B_2 with probability $(1-p)$ and define U_3 accordingly with respect to U_1, U_2. Then, for any measurable set $S \subseteq \mathbb{R}$,

$$\Pr[B_3 \in S] = p\Pr[B_1 \in S] + (1-p)\Pr[B_2 \in S] \leq e^{\epsilon}p\Pr[U_1 \in S] + (1-p)e^{\epsilon}\Pr[U_2 \in S] = e^{\epsilon}\Pr[U_3 \in S]$$

so $(B_3, U_3) \in \mathcal{P}_{\epsilon,g}$ and have expectations $pE_1 + (1-p)E_2$. Then, we apply the fact that all points in the Pareto frontier of a convex feasible region are solutions to a weighted sum optimization problem (see, for instance, Boyd [5, Chapter 4.7]). □

Properties of Pareto Optimal B_i, U_i:

Below, we establish the following properties of any Pareto optimal $(B_i, U_i) \in \mathcal{P}_{\epsilon,g}^{(i)}$ for any $i \in \mathbb{N}$ with probability density sequences $\{b_k^{(i)}\}_{k=0}^{\infty}$ and $\{u_k^{(i)}\}_{k=0}^{\infty}$ respectively:

(1) $b_k^{(i)} = 0$ for all $k < i$, $\{b_k^{(i)}\}$ is non-increasing for all $k \geq i$, and $b_k^{(i)}$ is bounded by $b_k^{(i)} \leq \frac{1-e^{-\epsilon}}{e^{-\epsilon}g}$ for all k.

(2) $u_k^{(i)}$ is fully determined by choice of $b_k^{(i)}$, that is, $u_k^{(i)} = e^{-\epsilon}b_i^{(i)}$ for all $k \in [1, i)$, $u_k^{(i)} = e^{-\epsilon}b_k^{(i)}$ for all $k \geq i$, and $u_0^{(i)} = \frac{i}{g}(1 - e^{-\epsilon} - \frac{(i-1)}{i}ge^{-\epsilon}b_i^{(i)}) \geq \frac{1-e^{-\epsilon}}{g}$.

Lemma A.9 (Support of B)**.** *Let B and U be any (ϵ, g)-one-sided indistinguishable distributions. Then, Pr$[B < g] = 0$.*

Proof. By indistinguishability $\frac{\Pr(B\in[0,g))}{\Pr(U\in[-g,0))} \leq e^{\epsilon}$, but by non-negativity, $\Pr(U \in [-g, 0)) = 0$. So, $\Pr(B \in [0, g)) = 0$. □

Note that by definition of B_i, the above lemma proves that $b_k^{(i)} = 0$ for all $k < i$, since any interval below i corresponds to the density of the random variable at a value below g.

Lemma A.10 (Upper bound on b)**.** *For any, $(B,U) \in \mathcal{P}_{\epsilon,g}$, if B has probability density function b, then:*

$$b(t) \leq \frac{(1-e^{-\epsilon})}{e^{-\epsilon}g} \quad \forall t \in [0, \infty).$$

Proof. Since $b(\cdot)$ is non-negative and integrates to 1 it must be bounded. Take any $t^* \in \arg\max_{t\in[g,\infty)} b(t)$. Then,

$$
\begin{aligned}
1 &= \int_0^\infty u(t)\ dt \tag{5}\\
&= \int_0^{t^*-g} u(t)\ dt + \int_{t^*-g}^{t^*} u(t)\ dt + \int_{t^*}^{\infty} u(t)\ dt \tag{6}\\
&\geq \int_0^{t^*-g} e^{-\epsilon} b(t+g)\ dt + \int_{t^*-g}^{t^*} e^{-\epsilon} b(t^*)\ dt + \int_{t^*}^{\infty} e^{-\epsilon} b(t)\ dt \tag{7}\\
&= g e^{-\epsilon} b(t^*) + \int_g^\infty e^{-\epsilon} b(t)\ dt \tag{8}\\
&= g e^{-\epsilon} b(t^*) + e^{-\epsilon}, \tag{9}
\end{aligned}
$$

where (3) follows from the indistinguishability definition and (1) and (5) follow since B and U both must integrate to 1 to be valid probability density functions. Then, $\max\limits_{t\in[0,\infty)} b(t) = b(t^*) \leq \frac{1-e^{-\epsilon}}{e^{-\epsilon}g}$.

□

Lemma A.11 (B_i determines U_i)**.** *For any $i \in \mathbb{N}$, let $(B_i, U_i) \in \mathcal{P}_{\epsilon,g}^{(i)}$ be Pareto optimal distributions (within $\mathcal{P}_{\epsilon,g}^{(i)}$) with probability density sequences $b_0^{(i)}, b_1^{(i)}, \ldots$ and $u_0^{(i)}, u_1^{(i)}, \ldots$ respectively. Then, $\forall k \in \mathbb{Z}_{>0}$ it holds that $u_k^{(i)} = \max\limits_{j\in[0,i]} e^{-\epsilon} b_{k+j}^{(i)}$ and $u_0^{(i)} = \frac{i}{g}\left(1 - \sum\limits_{k=1}^{\infty} \frac{g}{i} u_k^{(i)}\right)$.*

Proof. Informally, this proof will argue that if U_i has any "excess" probability mass in an interval greater than 0, we can move that probability mass to the interval at 0 and reduce the expectation of U_i. By Lemma A.6, B_i and U_i are (ϵ, g)-one-sided indistinguishable so $\forall k \in \mathbb{Z}_{>0}$ it must be that $u_k^{(i)} \geq \max\limits_{j\in[0,i]} e^{-\epsilon} b_{k+j}^{(i)}$. Assume for the sake of contradiction that there is some value $\ell > 0$ for which $u_\ell^{(i)} > \max\limits_{j\in[0,i]} e^{-\epsilon} b_{\ell+j}^{(i)} =: M$. Then, define U_i' to have $u_\ell'^{(i)} = M$, $u_0'^{(i)} = u_0^{(i)} + u_\ell^{(i)} - M$ and $u_k'^{(i)} = u_k^{(i)}$ for all other values of k. Then, U_i' is still a valid probability distribution and is (ϵ, g)-indistinguishable from B_i, but has lower expected value than U' contradicting the Pareto optimality of (B_i, U_i). The value of $u_0^{(i)}$ follows by requiring that the probability densities integrate to 1. □

Lemma A.12 ($\{b_k^{(i)}\}$ and $\{u_k^{(i)}\}$ are non-increasing)**.** *For any $i \in \mathbb{N}$, let $(B_i, U_i) \in \mathcal{P}_{\epsilon,g}^{(i)}$ be Pareto optimal distributions (within $\mathcal{P}_{\epsilon,g}^{(i)}$) with probability density sequences $b_0^{(i)}, b_1^{(i)}, \ldots$ and $u_0^{(i)}, u_1^{(i)}, \ldots$ respectively. Then, $\forall k \geq i$ it must be that $b_k^{(i)} \geq b_{k+1}^{(i)}$ and $\forall k \geq 0$ it must be that $u_k^{(i)} \geq u_{k+1}^{(i)}$.*

Proof. Suppose that $(B_i, U_i) \in \mathcal{P}_{\epsilon,g}^{(i)}$ are a Pareto optimal pair of distributions with density sequences $\{b_0^{(i)}, b_1^{(i)}, \ldots\}$ and $\{u_0^{(i)}, u_1^{(i)}, \ldots\}$ respectively. We will construct new random variables (B_i', U_i') with monotonically non-increasing density sequences $\{b_0'^{(i)}, b_1'^{(i)}, \ldots\}$ and $\{u_0'^{(i)}, u_1'^{(i)}, \ldots\}$ and argue that $\mathbb{E}[B_i'] \leq \mathbb{E}[B]$ and $\mathbb{E}[U_i'] \leq \mathbb{E}[U_i]$. We construct the new density sequences and a permutation $\pi : \mathbb{N} \to \mathbb{N}$ mapping $\{b_k^{(i)}\}$ to $\{b_k'^{(i)}\}$ as follows.

$\Pr[B_i' < g] = 0$ by Lemma A.9, so:

$$
\begin{aligned}
&b_k'^{(i)} = b_k^{(i)} = 0, \quad \forall k \in \mathbb{Z}, 0 \leq k \leq (i-1)\\
&\pi(k) = k, \quad \forall k \in \mathbb{Z}, 0 \leq k \leq (i-1).
\end{aligned}
$$

Then, we sort $\{b_k^{(i)}\}$ by moving the interval with highest probability mass in $\{b_k^{(i)}\}$ (breaking ties to the left) as far to the left as possible in $\{b_k'^{(i)}\}$:

$$\begin{aligned}
&\forall m \in \mathbb{Z}, m \geq i: \\
&I_m = \operatorname*{arg\,max}_{k \in \mathbb{N} \setminus \{\pi(j) | j < m\}} b_k^{(i)} \\
&\pi(m) = \min_{n \in I_m} n \\
&b_m'^{(i)} = b_{\pi(m)}^{(i)}.
\end{aligned}$$

Finally, by Lemma A.11, $\{u_k'\}$ must be determined by $\{b_k'\}$ in order to be Pareto optimal, so take:

$$u_k'^{(i)} = \begin{cases} e^{-\epsilon} b_k'^{(i)} & k \geq i \\ e^{-\epsilon} b_i'^{(i)} & 1 \leq k \leq (i-1) \\ \frac{i}{g}(1 - e^{-\epsilon}) - (i-1)e^{-\epsilon} b_i'^{(i)} & k = 0 \end{cases}$$

First, we argue that $(B_i', U_i') \in \mathcal{P}_{\epsilon,g}^{(i)}$. $\{b_k'\}$ defines a valid probability distribution since $\{b_k'^{(i)}\}$ is a permutation of $\{b_k^{(i)}\}$ so the distribution integrates to 1. Then, by construction, $\{u_k'^{(i)}\}$ is also a valid probability density sequence and integrates to 1. By Lemma A.10, $b_{\pi(i)}^{(i)} \leq \frac{1-e^{-\epsilon}}{e^{-\epsilon} g}$ so $u_0'^{(i)} \geq e^{-\epsilon} b_{\pi(i)}^{(i)} = e^{-\epsilon} b_i'^{(i)}$. Hence, the two distributions satisfy the (g, ϵ)-indistinguishability constraint by construction since b' is non-increasing above interval i, and $u_k'^{(i)} \geq e^{-\epsilon} b_i'^{(i)}$ $\forall k \leq i$ and $u_k'^{(i)} = e^{-\epsilon} b_k'^{(i)}$ $\forall k \geq i$.

Now, we argue that $\mathbb{E}[B_i'] \leq \mathbb{E}[B_i]$ since $\{b_k'^{(i)}\}$ is a permutation of $\{b_k^{(i)}\}$ that shifts probability mass to the left. By construction $\forall t \in [0, \infty)$, it holds that $\Pr[B_i' \leq t] \geq \Pr[B_i \leq t]$. So,

$$\mathbb{E}[B_i'] = \int_0^\infty 1 - \Pr[B_i' \leq t]\, dt \leq \int_0^\infty 1 - \Pr[B_i \leq t]\, dt = \mathbb{E}[B_i].$$

Finally, we want to show that $\mathbb{E}[U_i'] \leq \mathbb{E}[U_i]$. We will analyze the contribution to the expectation coming from intervals below i and above i separately.

Note that the expectation $\mathbb{E}[U_i] = \sum_{k=0}^\infty \left(u_k^{(i)} \frac{g}{i}\right)\left(\frac{2k+1}{2}\frac{g}{i}\right)$ so we can split the difference between the expectations as follows:

$$2\left(\frac{i}{g}\right)^2 (\mathbb{E}[U_i] - \mathbb{E}[U_i']) = \sum_{k=i}^{\infty} (2k+1)(u_k^{(i)} - u_k'^{(i)}) + \sum_{k=0}^{i-1} (2k+1)(u_k^{(i)} - u_k'^{(i)}).$$

Now, we state the following two observations, which we will apply repeatedly in the remainder of the proof:

(i) $\forall k \geq i: \; u_{\pi(k)}^{(i)} \geq e^{-\epsilon} b_{\pi(k)}^{(i)} = u_k'^{(i)}$, by indistinguishability of B and U and the definition of U'.

(ii) $\pi(\cdot)$ is a bijection on $[i, \infty)$ so $\sum_{k=i}^\infty u_k^{(i)} = \sum_{k=i}^\infty u_{\pi(k)}^{(i)}$.

By properties (i) and (ii) above, there is "excess probability density" above interval i in U compared to U' of

$$M = \sum_{k=i}^{\infty} u_k^{(i)} - u_k'^{(i)} = \sum_{k=i}^{\infty} u_{\pi(k)}^{(i)} - u_k'^{(i)} \geq 0.$$

Since $\sum_0^\infty u_k'^{(i)} = \sum_0^\infty u_k^{(i)}$, by symmetry there is excess probability mass of M below i in U' compared to U:

$$M = \sum_{k=0}^{i-1} u_k'^{(i)} - u_k^{(i)}.$$

Since $\{u'^{(i)}_k\}_{k=1}^{\infty}$ is non-increasing and by properties (i) and (ii) above the $\{u'^{(i)}_k\}$ are a permutation of $\{u^{(i)}_k\}$ with some values increased, the difference in expectations between U and U' above interval i is minimized by putting all of the excess probability mass M in interval i, so:

$$\sum_{k=i}^{\infty}(2k+1)(u^{(i)}_k - u'^{(i)}_k) \geq M(2i+1).$$

To analyze the difference in expectations coming from intervals in $k \in [0, i-1]$, we first argue that U' puts more probability mass on 0 than U, that is $u'^{(i)}_0 \geq u^{(i)}_0$ In particular, we will argue that $\sum_{k=1}^{\infty} u'^{(i)}_k \leq \sum_{k=1}^{\infty} u^{(i)}_k$. By indistinguishability, $\forall k \geq 1$ $u^{(i)}_k \geq e^{-\epsilon} b^{(i)}_{k+i}$ and $u^{(i)}_k \geq e^{-\epsilon} b^{(i)}_k$ so

$$\begin{aligned}\sum_{k=1}^{\infty} u^{(i)}_k &\geq \sum_{k=1}^{\pi(i)-i-1} e^{-\epsilon} b^{(i)}_{k+i} + \sum_{k=\pi(i)-i}^{\pi(i)-1} e^{-\epsilon} b^{(i)}_i + \sum_{k=\pi(i)}^{\infty} e^{-\epsilon} b^{(i)}_k \\ &= \sum_{k=i}^{\pi(i)-1} e^{-\epsilon} b^{(i)}_k + \sum_{k=\pi(i)}^{\infty} e^{-\epsilon} b^{(i)}_k + (i-1)e^{-\epsilon} B_i = \sum_{k=1}^{\infty} u'^{(i)}_k.\end{aligned}$$

Next, we argue that $u^{(i)}_k$ is non-decreasing on $[1, i-1]$. By Lemma A.11, for $\forall k \in \mathbb{Z}, (i-1) \geq k \geq 1:$ $u^{(i)}_k = \max_{j \in [i, i+k]} e^{-\epsilon} b^{(i)}_j \leq \max_{j \in [i, i+k+1]} e^{-\epsilon} b^{(i)}_j = u^{(i)}_{k+1}$. Therefore, putting the excess probability mass M in U' compared to U as far to the right as possible gives

$$\sum_{k=0}^{i-1}(2k+1)u^{(i)}_k \geq u'^{(i)}_0 + \sum_{k=0}^{i-1}\left((2k+1)\left(u'^{(i)}_k - \frac{M}{i-1}\right)\right) = \left(\sum_{k=0}^{i-1}(2k+1)u'^{(i)}_k\right) - M(i+1),$$

so

$$\sum_{k=0}^{i-1}(2k+1)(u'^{(i)}_k - u^{(i)}_k) \leq M(i+1).$$

Thus, we conclude that

$$\begin{aligned}2\left(\tfrac{i}{g}\right)^2(\mathbb{E}[U] - \mathbb{E}[U']) &= \sum_{k=i}^{\infty}(2k+1)(u^{(i)}_k - u'^{(i)}_k) + \sum_{k=0}^{i-1}(2k+1)(u^{(i)}_k - u'^{(i)}_k) \\ &\geq M(2i+1) - M(i+1) \\ &\geq Mi \geq 0,\end{aligned}$$

giving $\mathbb{E}[U] \geq \mathbb{E}[U']$. □

Pareto Frontier of B_i, U_i:

Now, we use the properties of Pareto optimal B_i, U_i to give an exact characterization of the probability density functions of Pareto optimal B_i, U_i:

Lemma A.13. *For any $i \in \mathbb{N}$, let $S_L = \{(B_i, U_i) \in \mathcal{P}^{(i)}_{\epsilon,g} : b^{(i)}_i = L$ and (B_i, U_i) are Pareto optimal$\}$ be all distributions in the Pareto frontier of $\mathcal{P}^{(i)}_{\epsilon,g}$ where $b^{(i)}_i$ is fixed to be some value $L \leq \frac{1-e^{-\epsilon}}{e^{-\epsilon}g}$. Then, either $S_L = \emptyset$ or S_L contains a single pair of distributions where letting $n = \lfloor \frac{i}{g} \cdot \frac{1}{L} \rfloor$:*

(i) $b^{(i)}_k = L$ for $k \in [i, n]$, $b^{(i)}_{n+1} = \frac{i}{g}(1 - \frac{g}{i}nL)$, and $b^{(i)}_k = 0$ for all other values of k.

(ii) $u^{(i)}_k = e^{-\epsilon}L$ for $k \in [1, n]$, $u^{(i)}_{n+1} = e^{-\epsilon}b^{(i)}_{n+1}$, $u^{(i)}_0 = \frac{i}{g}(1 - e^{-\epsilon} - \frac{(i-1)}{i}ge^{-\epsilon}L)$ and $u^{(i)}_k = 0$ for all other k.

so B_i is a "nearly uniform" distribution above g with any excess probability mass in the final constant interval, and U_i has the same probability mass as B_i discounted by $e^{-\epsilon}$ except in a small band around 0 *where it may have inflated probability mass.*

Proof. First, note that fixing $b_i^{(i)} = L$, by Lemma A.11 we have that $u_k^{(i)}$ is fully determined by L for $k \in [0, i)$. Therefore, for any Pareto optimal U_i, B_i with $b_i^{(i)} = L$:

$$\mathbb{E}[U_i] = \Pr[U_i < g]\mathbb{E}[U_i|U_i < g] + \Pr[U_i \geq g]\mathbb{E}[U_i|U_i \geq g] = C_L + (1 - e^{-\epsilon})\mathbb{E}[B_i],$$

where C_L is a constant determined by L. Therefore, there is a unique minimizer of $\mathbb{E}[U_i]$ and $\mathbb{E}[B_i]$ over S_L that is obtained by minimizing $\mathbb{E}[B_i]$. Since B_i is monotonically non-increasing above i, the distribution that minimizes its expectation puts mass equal to $b_i^{(i)} = L$ at as many intervals as possible giving $n = \lfloor \frac{i}{g}\frac{1}{L} \rfloor$ intervals with $b_k^{(i)} = L$ and any remaining mass needed to make the distribution integrate to 1 in the final interval, yielding the unique optimal distributions for (B_i, U_i). □

Taking limits as $i \to \infty$ of each distribution in the set of distributions from Lemma A.13 yields exactly the set of zero-inflated Uniform distributions in Theorem 4.6, so we conclude that any optimizer of a weighted sum objective must come from this set of distributions and hence the Pareto frontier consists of Zero-inflated Uniform distributions.

A.4.3 Optimal choice of parameter η

Finally, we derive the optimal choice of parameter η given ϵ, g and weighting parameter w. From Theorem 4.3 the zero-inflated Uniform with parameters ϵ, g has expectation: $\mathbb{E}[B] = \frac{1}{2}g\left(\eta + \frac{\eta}{\eta - e^{-\epsilon}}\right)$ and $\mathbb{E}[U] = \frac{1}{2}g\left(\frac{\eta^2}{\eta - e^{-\epsilon}}\right)$. Therefore, by Pareto optimality of the zero-inflated Uniform proven in Section A.4.2, for any $w \in [0, 1]$, the weighted sum of the expectations can be optimized by choosing

$$\eta^* \in \arg\min_{\eta \in (e^{-\epsilon}, 1]} w\left(\eta + \frac{\eta}{\eta - e^{-\epsilon}}\right) + (1 - w)\frac{\eta^2}{\eta - e^{-\epsilon}}.$$

This objective is convex on $(e^{-\epsilon}, 1]$ as it has second derivative with respect to η of $w\left(1 + \frac{1}{\eta - e^{-\epsilon}}\right) + (1 - w)\frac{\eta^2}{\eta - e^{-\epsilon}} > 0$ for any $w \in [0, 1]$ and $\eta \in (e^{-\epsilon}, 1]$.

The first derivative of this objective with respect to η is $\frac{1}{(\eta - e^{-\epsilon})^2}\left(w(e^{-2\epsilon} - e^{-\epsilon}) + \eta(\eta - 2e^{-\epsilon})\right)$. Note that for any w, the derivative begins at a negative value on the interval $(e^{-\epsilon}, 1]$ and is increasing on this interval. Therefore, letting $\hat{\eta}$ denote the value at which the first derivative is 0, we obtain $\hat{\eta} = e^{-\epsilon}\left(1 + \sqrt{1 + e^{\epsilon}\frac{w}{1 - w}}\right)$. Since $\hat{\eta}$ must fall in the interval $(e^{-\epsilon}, 1]$ we take $\eta^* = \min\{1, \hat{\eta}\}$ to get the optimal η given in Algorithm 2, where η^* is optimal since the utility function must be decreasing on $[e^{-\epsilon}, 1)$ in the case that $\hat{\eta} > 1$.

A.5 Proof of Theorem 4.8 (Impossibility of Two-Sided DP)

We prove the result for each of following definitions of "neighboring" separately:

(1) *Add or remove a batched comment.* Consider any input A where an instance of batching occurs at some time t. Let A' be identical to A, except some comment c that arrived in a batch at time t does not arrive at all in A'. Then on input A', since any valid comment posting mechanism cannot generate fake data, for any $d > 0$ and time $T = t + d$, the mechanism outputs c at time T with probability 0. However, if the mechanism is (ϵ, δ)-DP with $\epsilon < \infty$, then for any release time T the mechanism outputs c within time T with probability at most $\delta < 1$ and so the mechanism violates the eventual release of all comments property.

(2) *Move a batched comment to another arrival time where it is no longer batched.* Consider any input A with an instance of batching that occurs at some time t. Fix any time horizon $T = t + d$ where $d > 0$. Define A' to be an identical set with one comment c moved from time t to time T. Since a valid comment posting mechanism must delay comments and cannot generate fake data, the mechanism outputs comment c at time T or later on input A' with probability 1. However, if the mechanism is (ϵ, δ)-DP with $\epsilon < \infty$ then it must delay comment c until at least time $T = t + d$ with probability at least $1 - \delta$. Taking d to be arbitrarily large, the mechanism violates the eventual release of all comments property for any $\delta < 1$.

(3) *Move a batched comment by at most g units of time to another arrival time where it is no longer batched.* Let $A^{(1)}$ be an input where a single comment arrives every g units of time. Then, define $A^{(1)'}$ to be a neighboring input to $A^{(1)}$ where c_2 arrives in a batch with c_1 at time 0. Define $A^{(2)}$ to be a neighboring input to $A^{(1)'}$ where c_1 and c_2 arrive separately with c_1 at time g and c_2 at time 0 and so on:

$$\begin{aligned} A^{(1)} &= \{c_1, t = 0\}, \{c_2, t = g\}, \{c_3, t = 2g\}, \ldots \\ A^{(1)'} &= \{c_1, c_2, t = 0\}, \emptyset, \{c_3, t = 2g\}, \ldots \\ A^{(2)} &= \{c_2, t = 0\}, \{c_1, t = g\}, \{c_3, t = 2g\}, \ldots \\ A^{(2)'} &= \{c_2, t = 0\}, \{c_1, c_3, t = g\}, \emptyset, \ldots \end{aligned}$$

Now, for any j: $A^{(j)}$ and $A^{(j)'}$ are neighbors and $A^{(j)}$ and $A^{(j-1)'}$ are neighbors. On input $A^{(j)}$, comment c_1 arrives at time jg and so any valid comment posting therefore posts c_1 at time jg or later with probability 1 since it can only delay comments. Likewise, because $A^{(j-1)'}$ neighbors $A^{(j)}$ and the mechanism cannot generate fake data, any (ϵ, δ)-DP mechanism releases c_1 at a time earlier than jg with probability at most δ on input $A^{(j-1)'}$. Since $A^{(j-1)}$ neighbors $A^{(j-1)'}$, the mechanism releases c_1 at a time earlier than j with probability at most 2δ on this input. Thus, on input $A^{(1)}$, comment c_1 gets posted before time jg with probability less than $2j\delta$. This suggests that the comment gets delayed by at least D with probability at least $1 - 2\delta(\frac{D}{g} + 1)$.

A.6 Proof of Proposition 5.1 (Hypothesis Testing Interpretation of OSDP)

Fix comment c_1 and let c_2 denote the closest comment to arrive in C. Let R denote the rejection region of the adversary's chosen hypothesis test. Let $A^{(B)}$ be any arrival set where $t_1 = t_2$. Let A_d be an identical arrival set, except that c_2 arrives unbatched d units of time after c_1 (so $t_2 - t_1 = d$) and let A'_d be an identical arrival set except that c_1 arrives d units of time after c_2. Then, conditioning on the event that $t_2 - t_1 \leq g$, we have that for any rejection region R:

$$\begin{aligned} \textit{Type I Error} &\geq \sum_{n=0}^{g} \Pr[t_2 - t_1 = d; \mathcal{D}] \Pr[\mathcal{M}(A_d) \in R] + \sum_{n=0}^{g} \Pr[t_1 - t_2 = d; \mathcal{D}] \cdot \Pr[\mathcal{M}(A'_d) \in R] \\ &\geq e^{-\epsilon} F_{\mathcal{D}}(g) \Pr[\mathcal{M}(A^{(B)}) \in R] = e^{-\epsilon} F_{\mathcal{D}}(g) \textit{Power}, \end{aligned}$$

where the second line follows from the one-sided differential privacy guarantee on g-adjacent inputs.

B Estimation of Batching Deanonymization Risk Statistics

Recall that in Section 1, we provided statistics on the rate of batching at a peer-reviewed conference. We used these statistics in Figure 2 to estimate the linkage risk arising due to observing batched comments. In this section, we provide details about the measurement method used to estimate the batching statistics.

In order to estimate the prevalence of batching in the peer-review process of a conference, we measure the following statistics. For any individual reviewer or meta-reviewer, we order all of their comments on all papers in increasing order of post time. If two comments arrive immediately next to each other in this sequence and were made on different papers, we consider these to be "consecutive comments from the same (meta)-reviewers on different papers." Note that this excludes comments that are made on the same paper by the same (meta)-reviewer consecutively, because consecutive comments by the same (meta)-reviewer on the same paper do not generate additional linkage risk for the (meta)-reviewer. For example, consider the following sequence of comment arrivals from a single (meta)-reviewer (where units of time are minutes from the start of the commenting period):

$$(c_1, p_1, t_1 = 0), (c_2, p_2, t_2 = 5), (c_3, p_2, t_2 = 6), (c_4, p_2, t_2 = 8)(c_5, p_3, t_3 = 100).$$

In this example, we count the first two comments (c_1 and c_2) and the last two comments (c_4 and c_5) as consecutive arrivals on different papers. We then capture the rate of batching under 5 minutes by computing the number of consecutive comments that arrive within 5 minutes of each other divided by the total number of consecutive comment arrivals. So, in the example above, the rate of batching is 50% since comments c_1 and c_2 arrive within 5 minutes of one another, while c_4 and c_5 do not. Applying this measurement method to a dataset of comments made by reviewers and meta-reviewers on papers at a top Computer Science conference, we find that there is a 30.10% chance that a comment arrives in a batch with a consecutive comment from the same (meta)-reviewer.

For a baseline, we additionally compute how often comments from *different* (meta)-reviewers may appear at times close to each other. We look at each pair of distinct reviewers from the set of all reviewers. We then calculate whether any pair of comments from these two (meta)-reviewers arrived within a cutoff of 5 minutes of one another. We find that there is a 0.66% chance that a randomly chosen pair of (meta)-reviewers makes a pairs of comments that arrive within 5 minutes of one another. We note that the first statistic capturing the rate of batching excludes reviewers who made only a single comment in the entire conference, as it is not possible for these reviewers to engage in batching. In contrast, the second statistic capturing the baseline rate of close arrivals includes cases where a reviewer makes only a single comment. These comments are counted in the statistic, since any comment may appear to be batched with an anonymized comment made by a different reviewer from the perspective of an observer who does not know reviewer identities.

C A Queue-Based Mechanism for Privacy Against Batched Timing Attacks

In this section, we discuss an alternative privacy formulation that we call "ϵ-batching privacy" and give an algorithm that satisfies privacy under this formulation by delaying comments using a queue to preserve privacy. In doing so, our queue-based mechanism *preserves the ordering in which comments arrive*, a property that may be useful in certain applications. The privacy guarantees are not directly comparable to (ϵ, g)-OSDP because we make substantially different sets of assumptions in the adversarial model. However, one can think of both approaches as responses to the impossibility results for standard two-sided proven in Section 4.4. While (ϵ, g)-OSDP relaxes two-sided DP by introducing a bound g on the gap between unbatched comments and by making the notion of neighbors asymmetric, ϵ-batching privacy introduces distributional assumptions on the inputs that capture an adversary's uncertainty about comment arrivals.

C.1 Problem Formulation

In this problem formulation, we assume that comment arrivals are drawn i.i.d. from some unknown distribution over papers and reviewers. We call this the arrival process. We assume *discrete time* comment arrivals over an infinite time horizon so comments arrive at each time-step drawn from this unknown distribution.

First, we present the arrival process if no batching occurs. In the absence of batching, a single comment arrives at every unit of time. We make an *i.i.d. assumption* on arrivals. At each time-step, the paper-reviewer pair associated with the comment is drawn independently from a (potentially unknown) probability distribution $\mathcal{D}$ over $\mathcal{P} \times \mathcal{R}$ (where $\mathcal{P}$ is the set of all papers and $\mathcal{R}$ is the set of all reviewers). For instance, $\mathcal{D}$ could be a uniform distribution over $\mathcal{P} \times \mathcal{R}$ although it need not be uniform or even known to the algorithm. We say $A \leftarrow \mathcal{A}^{(0)}$ if the arrivals are drawn from this no-batching process.

An instance of *potential batching* consists of multiple comments. The batch arrives at a single time-step, but the adversary is uncertain as to which papers and reviewers are in the batch. Thus, when potential batching occurs, the arrival process remains the same except for one modification—batches consisting of more than one comment arrive at specific fixed time-steps. Formally, let B be a multi-set of time-steps at which batching occurs. The arrival process proceeds as follows:

- On time-steps not contained in B, no batching occurs and a single comment arrives.
- For each time-step contained in B, an additional comment arrives due to batching. For instance, if $B = \{10, 10, 15\}$ then a single comment arrives at each time-step, but two additional comments arrive at time 10 due to batching and one additional comment arrives at time 15 due to batching.

The paper-reviewer pairs associated with the batched comments are drawn independently with replacement from distribution $\mathcal{D}$. We say that $A \leftarrow \mathcal{A}^{(B)}$ if the arrivals are drawn from this process with batchings occurring at time-steps in $\mathcal{A}^{(B)}$. We allow comments to arrive according to $\mathcal{A}^{(B)}$ for any finite multi-set of time-steps B. We do not assume any prior knowledge of either B nor $|B|$.

Then, we define a comment posting mechanism to be ϵ-batching private in this formulation, if the mechanism obscures whether the inputted comment arrival set arrived per the batching process (with any number of batches) or the no batching process (whereby 0 batches appeared):

Definition C.1 (Batching Privacy)**.** A comment posting mechanism $\mathcal{M}$ is *ϵ-batching private* with respect to arrival processes $(\mathcal{A}^{(0)}, \mathcal{A}^{(B)})$ if for all time horizons $T \geq 1$, all finite batching multi-sets B, and any output of the mechanism between time 1 and T, S_T:

$$\Pr[\mathcal{M}(A) = S_T; A \leftarrow \mathcal{A}^{(B)}] \leq e^{\epsilon}\Pr[\mathcal{M}(A) = S_T; A \leftarrow \mathcal{A}^{(0)}] \text{ and}$$
$$\Pr[\mathcal{M}(A) = S_T; A \leftarrow \mathcal{A}^{(0)}] \leq e^{\epsilon}\Pr[\mathcal{M}(A) = S_T; A \leftarrow \mathcal{A}^{(B)}].$$

Note that unlike typical differential privacy formulations, this notion of privacy requires distributional assumptions on the data-generating process as we assume that comments are generated by an i.i.d. arrival model.

C.2 Results

Under this formulation, we design a mechanism described in Algorithm 4 that delays comments by deploying them to a queue. The algorithm guarantees *perfect batching privacy* ($\epsilon = 0$), as shown in the following result.

Proposition C.2 (Privacy)**.** *Algorithm 4 guarantees perfect batching privacy ($\epsilon = 0$) for comments arriving according to $\mathcal{A}^{(0)}$ and $\mathcal{A}^{(B)}$ for any B.*

Proof. Fix a time horizon T and multi-set of batching times B. We let $\mathcal{D}(c)$ denote the probability of observing the comment c under distribution $\mathcal{D}$. When the algorithm is applied to comments drawn according to the no batching process, one comment arrives at each time-step and all comments are posted immediately so by the i.i.d. assumption, $\Pr[\mathcal{M}(A) = c_{1:T}; A \leftarrow \mathcal{A}^{(0)}] = \prod_{i=1}^{T} \mathcal{D}(c_t)$.

If comments were drawn according to the process where batching occurred at times B, then at any time-step before the first instance of batching occurs the mechanism posts the single comment that arrives so the probability of observing output $\{c\}$ is $\mathcal{D}(c)$ independent of other-timesteps. On the first instance of batching, the mechanism posts one of the batched comments chosen uniformly at random from the batch, so due to the i.i.d. arrivals of the batch the probability of observing this output $\{c\}$ at this time-step is also $\mathcal{D}(c)$. At any later time-step, the algorithm posts the comment at the top of the queue, which consists of previous comments that arrived i.i.d. drawn from $\mathcal{D}$. Therefore, the probability of observing any output is still $\Pr[\mathcal{M}(A) = c_{1:T}; A \leftarrow \mathcal{A}^{(B)}] = \prod_{i=1}^{T} \mathcal{D}(c_t)$. □

The algorithm delays comments by a deterministic value depending on the number of batched comments that have arrived already.

Proposition C.3 (Delay)**.** *If comments arrive according to $\mathcal{A}^{(B)}$, then Algorithm 4 adds worst-case delay to any comment equal to $|B|$.*

Proof. After the last instance of batching in B, there are $T + |B|$ comments that have arrived in total. The mechanism posts the earliest-arriving comment at each time-step and delays the incoming comment so the queue has length $|B|$ and any single incoming comment is delayed for $|B|$ timesteps before being posted. Any comments arriving before all instances have batching have occurred are delayed by the number of additional comments arriving due to batching at an earlier time-step, so have delay less than $|B|$. □

Algorithm 4 Queue Mechanism

```
Initialize empty queue $Q = \emptyset$
for t= $1, 2, \ldots$ do
  if set of batched comments $A$ arrives then
    if $Q \neq \emptyset$ then
      Dequeue comment $c'$ from $Q$ and post it.
      Enqueue all comments in $A$ to $Q$ in a random order.
    else
      Choose $c \in A$ uniformly at random to post.
      Enqueue all comments in $A \setminus \{c\}$ to $Q$ in a random order.
      Post comment $c$ immediately.
    end if
  else if a single comment $c$ arrives then
    if Q$\neq \emptyset$ then
      Dequeue comment $c'$ from $Q$ and post it.
      Enqueue comment $c$ to $Q$.
    else
      Post comment $c$.
    end if
  end if
end for
```

In fact, this perfectly private mechanism is optimal for this privacy formulation as it achieves the best possible worst-case delay to any comment at any value of ϵ. In particular, at any setting of ϵ any batching-private comment posting mechanism must delay a comment by at least $|B|$ in the worst-case:

Proposition C.4 (Lower Bound, Minimum Delay)**.** *Any comment posting mechanism guaranteeing ϵ-batching privacy with any $\epsilon < \infty$ for comments arriving according to $\mathcal{A}^{(0)}$ and $\mathcal{A}^{(B)}$ must introduce delay of at least $|B|$ to at least one comment when applied to comments arriving according to $\mathcal{A}^{(B)}$.*

It follows immediately that since the Queue Mechanism (Algorithm 4) achieves this lower bound it is optimal among ϵ-batching private mechanisms in minimizing worst-case delay:

Corollary C.5. *For any setting of privacy parameter ϵ, Algorithm 4 is optimal among ϵ-batching private comment posting mechanisms in minimizing the worst-case delay added to any comment.*

Proof. Let $T' = \max\{B\}$ be the latest time-step when batching occurs and $T = T' + |B|$. Then, if comments arrive according to $\mathcal{A}^{(B)}$, $T' + |B| + 1 = T + 1$ comments arrive up until time $T' + 1$. Assume for the sake of contradiction that all of the comments arriving before time $T' + 1$ are posted with delay strictly less than $|B|$. Then, when acting on comments arriving according to $\mathcal{A}^{(B)}$, the mechanism must post at least $T + 1$ comments within time horizon T (with probability 1). However, under arrival process $\mathcal{A}^{(0)}$, only T comments have arrived up until T, so no mechanism can ever output $T + 1$ comments up until time T. Hence, any output of the mechanism up until time T on comments arriving per $\mathcal{A}^{(B)}$ contains $T + 1$ comments with probability 1, while for comments arriving per $\mathcal{A}^{(0)}$ any output up until time T contains $T + 1$ comments with probability 0. □

The above formulation and corresponding queue-based mechanism offer an alternative approach to provide privacy in light of the impossibility results for two-sided DP. Here, we relax the problem by introducing distributional assumptions on inputs to the mechanism. While this does not yield a privacy-delay trade-off in ϵ, it allows for a mechanism that preserves the ordering of comments. As noted in Section 7, an interesting direction of future work is to understand how we might make the Zero-Inflated Uniform Mechanism order-preserving as well.

D Additional Experimental Results

In this section, we provide experimental results that augment those presented in the main text.

D.1 Wikipedia

In the main text, we showed results setting $g = 11$ minutes by choosing the 25th percentile of prior inter-arrival times for the category "21-st century American Politicains" and $g = 36$ minutes at the 50th percentile. Here, we provide additional results, setting $g = 79$ minutes based on the 75th percentile of the inter-arrival distribution as shown in Figure 8 and Table 2. Algorithm 2 adds significantly higher delay at this setting of g, and consequently the adversary's batched timing linkage attack performs quite poorly. For instance, taking $\epsilon = 0.5$ corresponds to an average delay of roughly 3 hours and maximum delay of 6 hours, but renders the attack highly inaccurate: the attack now achieves around 80% recall at 10% precision compared to the non-private baseline which achieves 85% recall at 80% precision.

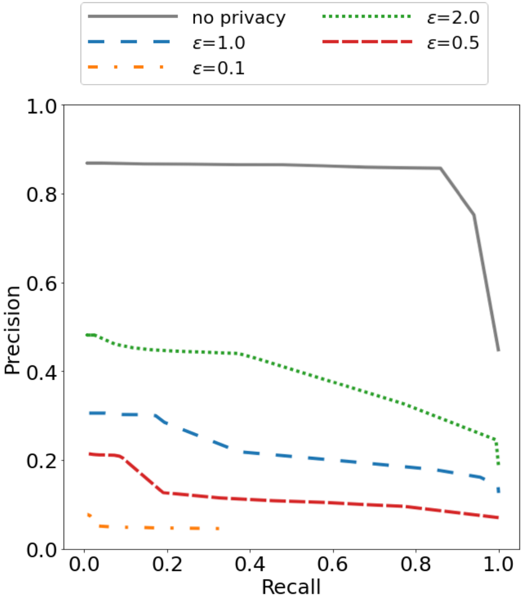


Figure 8: Accuracy in linking pairs of Wikipedia article revisions within the category "21st-century American Politicians" based on batched timing (averaged over 5 runs of the randomized privacy mechanism) for g set to 79 minutes.

	Mean Delay				Maximum Delay			
	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$
$g = 79$	820	192	115	77	1615	360	205	129

Table 2: Mean and maximum delay (in minutes) added to Wikipedia article revisions within the category "21st-century American Politicians" for g set to the 75th percentile of the historical inter-arrival distribution.

D.2 Bitcoin

In the main text, we showed results using Algorithm 2 with g set to the median of the historical inter-arrival times of transactions sent to a given output address (with a default of 10 minutes when there were no prior transactions.) In this section, we give results for alternative settings of g. In Figure 9 and Figure 10 we show the delay added to comments and the success of attacks when g is set to a more lenient value based on the 25-th percentile of historical transaction inter-arrival times. In Figure 11 and Figure 12 we show results for a stricter setting of g to the 75-th percentile of historical transaction inter-arrival times.

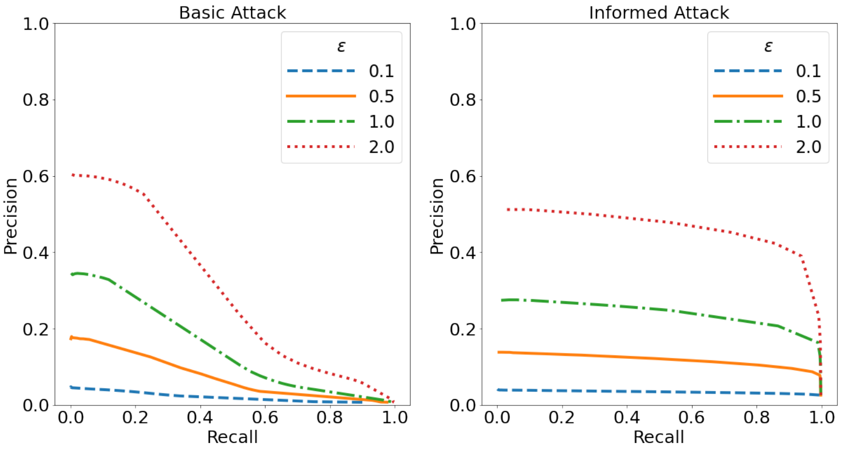


Figure 9: Performance of basic and informed attacks on Bitcoin transactions when g is set to the 25th percentile historical inter-arrival time for an output address.

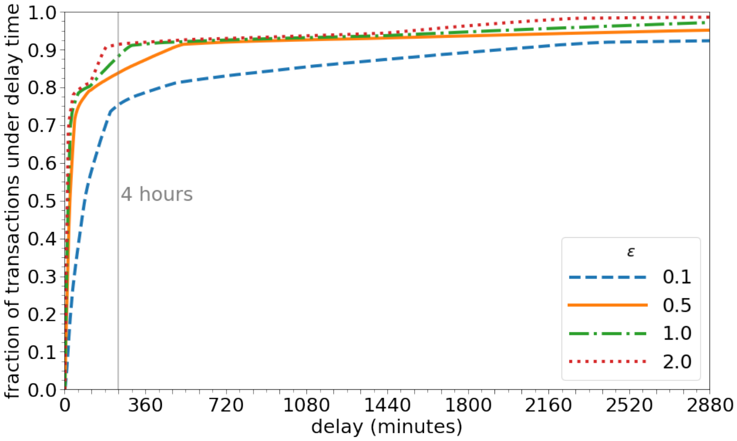


Figure 10: Cumulative distribution of delay added to batched Bitcoin transactions (averaged over 5 trials). Delay is drawn from a privacy-preserving uniform distribution with g set to the 25th percentile of the inter-arrival time of transactions to an output address within the past 7 days.

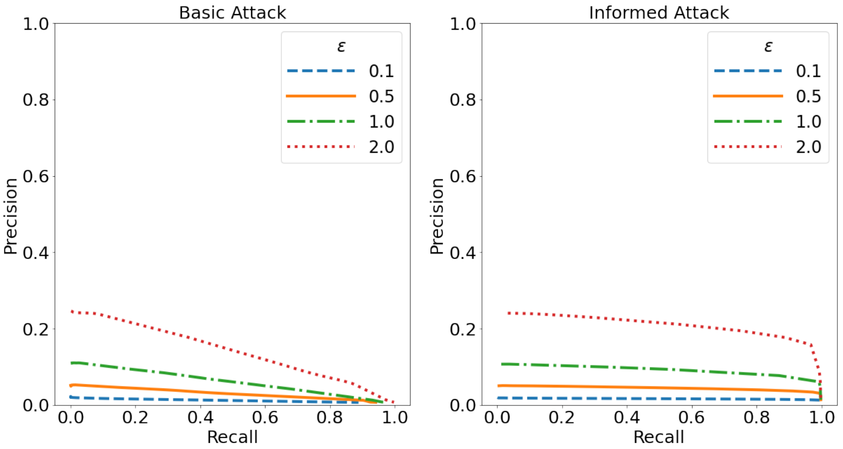


Figure 11: Performance of basic and informed attacks on Bitcoin transactions when g is set to the median historical inter-arrival time for an output address.

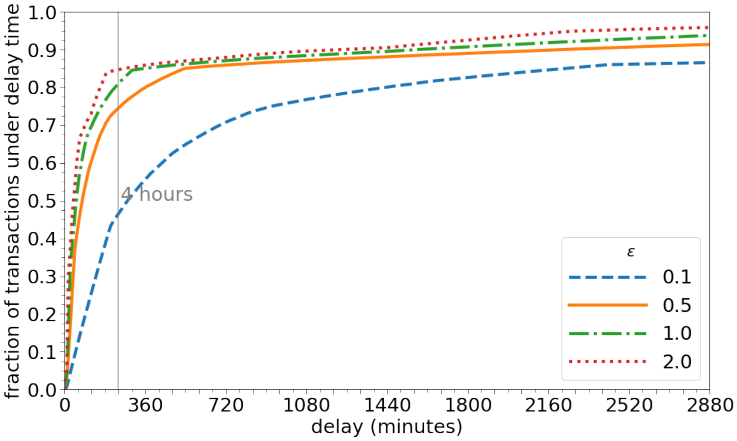


Figure 12: Cumulative distribution of delay added to batched Bitcoin transactions (averaged over 5 trials). Delay is drawn from a privacy-preserving uniform distribution with g set to the median of the inter-arrival time of transactions to an output address within the past 7 days.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract specifically describes each finding of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: We discuss future work in section 7, but do not discuss limitations in detail.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   We give proofs both in the main text and appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed in section 6 (Experiments).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open Access to Data and Code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: available on GitHub, linked to in paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are given in the experimental section (6).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our precision-recall curves and delay plots do not show error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: There were not significant compute resources used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn’t make it into the paper).

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have preserved anonymity and gotten IRB approval for this work.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Privacy risks are discussed in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for Existing Assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: Does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: Disclosed to IRB board.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.