# Better Differentially Private Approximate Histograms and Heavy Hitters using the Misra-Gries Sketch

Christian Janos Lebeda
Basic Algorithms Research Copenhagen
IT University of Copenhagen
Copenhagen, Denmark
christian.j.lebeda@gmail.com

Jakub Tětek
Basic Algorithms Research Copenhagen
University of Copenhagen
Copenhagen, Denmark
j.tetek@gmail.com

## ABSTRACT

We consider the problem of computing differentially private approximate histograms and heavy hitters in a stream of elements. In the non-private setting, this is often done using the sketch of Misra and Gries [Science of Computer Programming, 1982]. Chan, Li, Shi, and Xu [PETS 2012] describe a differentially private version of the Misra-Gries sketch, but the amount of noise it adds can be large and scales linearly with the size of the sketch; the more accurate the sketch is, the more noise this approach has to add. We present a better mechanism for releasing a Misra-Gries sketch under $(\varepsilon, \delta)$-differential privacy. It adds noise with magnitude independent of the size of the sketch; in fact, the maximum error coming from the noise is the same as the best known in the private non-streaming setting, up to a constant factor. Our mechanism is simple and likely to be practical. In the full version of the paper we also give a simple post-processing step of the Misra-Gries sketch that does not increase the worst-case error guarantee. It is sufficient to add noise to this new sketch with less than twice the magnitude of the non-streaming setting. This improves on the previous result for $\varepsilon$-differential privacy where the noise scales linearly to the size of the sketch.

## 1. INTRODUCTION

Computing the histogram of a dataset is one of the most fundamental tasks in data analysis. At the same time, releasing a histogram may present significant privacy issues. This makes the efficient computation of histograms under privacy constraints a fundamental algorithmic question. Notably, differential privacy has become in recent years the golden standard for privacy, giving formal mathematical privacy guarantees. It would thus be desirable to have an efficient way of (approximately) computing histograms under differential privacy.



Histograms have been investigated thoroughly in the differentially private setting [1, 2, 7, 8, 10, 12, 14]. These algorithms start by computing the histogram exactly and they then add noise to ensure privacy. However, in practice, the amount of data is often so large that computing the histogram exactly would be impractical. This is, for example, the case when computing the histogram of high-volume streams such as when monitoring computer networks, online users, financial markets, and similar. In that case, we need an efficient streaming algorithm. Since the streaming algorithm would only compute the histogram approximately, the above-mentioned approach that first computes the exact histogram is infeasible. In practice, non-private approximate histograms are often computed using the Misra-Gries (MG) sketch [15]. The MG sketch of size $k$ returns at most $k$ items and their approximate frequencies $\hat{f}$ such that $\hat{f}(x) \in [f(x) - n/(k+1), f(x)]$ for all elements $x$ where $f(x)$ is the true frequency and $n$ is the length of the stream. This error is known to be optimal [5]. In this work, we develop a way of releasing a MG sketch in a differentially private way while adding only a small amount of noise.This allows us to efficiently and accurately compute approximate histograms in the streaming setting while not violating users' privacy. This can then be used to solve the heavy hitters problem in a differentially private way. Our result improves upon the work of Chan, Li, Shi, and Xu [6] who also show a way of privately releasing the MG sketch, but who need a greater amount of noise; we discuss this below.

In general, the issue with making approximation algorithms differentially private is that although we may be approximating a function with low global sensitivity, the algorithm itself (or rather the function it implements) may have a much larger global sensitivity. This increases the amount of noise required to achieve privacy using standard techniques. We get around this issue by exploiting the structure of the difference between the MG sketches for neighboring inputs. This allows us to prove that the following simple mechanism ensures $(\varepsilon, \delta)$-differential privacy: (1) We compute the Misra-Gries sketch, (2) we add to each counter independently noise distributed as Laplace$(1/\varepsilon)$, (3) we add to all counters the same value, also distributed as Laplace$(1/\varepsilon)$, (4) we remove all counters smaller than $1 + 2\ln(3/\delta)/\varepsilon$. Specifically, we show that this algorithm satisfies the following guarantees:

THEOREM 11 (SIMPLIFIED). *The above algorithm is $(\varepsilon, \delta)$-differentially private, uses $2k$ words of space, and returns a frequency oracle $\hat{f}$ with maximum error of $n/(k+1) + O(\log(1/\delta)/\varepsilon)$ with high probability for $\delta$ being sufficiently small.*

A construction for a differentially private Misra-Gries sketch has been given before by Chan et al. [6]. However, the more accurate they want their sketch to be (and the bigger it is), their approach has to add *more* noise. The reason is that they directly rely on the global $\ell_1$-sensitivity of the sketch. Specifically, if the sketch has size $k$ (and thus error $n/(k+1)$ on a stream of $n$ elements), its global sensitivity is $k$, and they thus have to add noise of magnitude $k/\varepsilon$. Their mechanism ends up with an error of $O\left(k\log(d)/\varepsilon\right)$ for $\varepsilon$-differential privacy with $d$ being the universe size. This can be easily improved to $O\left(k\log(1/\delta)/\varepsilon\right)$ for $(\varepsilon,\delta)$-differential privacy with a thresholding technique similar to what we do in step (4) of our algorithm above. This also means that they cannot get more accurate than error $\Theta\left(\sqrt{n\log(1/\delta)/\varepsilon}\right)$, no matter what value of $k$ one chooses. We achieve that the biggest error, as compared to the values from the MG sketch, among all elements is $O(\log(1/\delta)/\varepsilon)$ assuming $\delta$ is sufficiently small (we show more detailed bounds including the mean squared errors in Theorem 11). This is the same as the best private solution that starts with an exact histogram [14]. In fact, for any mechanism that outputs at most $k$ heavy hitters there exists input with error at least $n/(k+1)$ in the streaming setting [5] and input with error at least $O(\log(\min(d,1/\delta))/\varepsilon)$ [2] under differential privacy.

In the full version of this paper, we also show how to achieve pure $\varepsilon$-differential privacy with error $n/(k+1)+O(\log(d)/\varepsilon)$. Therefore the error of our mechanisms is asymptotically optimal for approximate and pure differential privacy, respectively.

Chan et al. [6] use their differentially private Misra-Gries sketch as a subroutine for continual observation and combine sketches with an untrusted aggregator. Those settings are not a focus of our work but our algorithm can replace theirs as the subroutine, leading to better results also for those settings. However, the error from noise still increases linearly in the number of merges when the aggregator is untrusted. As a side note, we show that in the case of a trusted aggregator, the approach of [6] can handle merge operations without increasing error. While that approach adds significantly more noise than ours if we do not merge, it can with this improvement perform better when the number of merges is very large (at least proportional to the sketch size).

Another approach that can be used is to use a randomized frequency oracle to recover heavy hitters. However, it seems hard to do this with the optimal error size. In its most basic form [11, Appendix D], this approach needs noise of magnitude $\Theta(\log(d)/\varepsilon)$, even if we have a sketch with sensitivity 1 (the approach increases the sensitivity to $\log(d)$, necessitating the higher noise magnitude), leading to maximum error at least $\Omega(\log(k)\log(d)/\varepsilon)$. Bassily, Nissim, Stemmer, and Guha Thakurta [3] show a more involved approach which reduces the maximum error coming from the noise to $\Theta((\log(k)+\log(d))/\varepsilon)$, but at the cost of increasing the error coming from the sketch by a factor of $\log(d)$. This means that even if we had a sketch with error $\Theta(n/k)$ and sensitivity 1, neither of these two approaches would lead to optimal guarantees, unlike the algorithm we give in this paper.

See https://github.com/JakubTetek/Differentially-Private-Misra-Gries for sample implementations of the algorithms we present in this paper.

## 2. TECHNICAL OVERVIEW

*Misra-Gries sketch.* Since our approach depends on the properties of the MG sketch, we describe it here. Readers familiar with the MG sketch may wish to skip this paragraph. We describe the standard version; in Section 5 we use a slight modification, but we do not need that here.

Suppose we receive a sequence of elements from some universe. At any time, we will be storing at most $k$ of these elements. Each stored item has an associated counter, other elements have implicitly their counter equal to 0. When we process an element, we do one of the following three updates: (1) if the element is being stored, increment its counter by 1, (2) if it is not being stored and the number of stored items is $< k$, store the element and set its counter to 1, (3) otherwise decrement all $k$ counters by 1 and remove those that reach 0. The exact guarantees on the output will not be important now, and we will discuss them in Section 5.

*Our contributions.* We now sketch how to release an MG sketch in a differentially private way.

Consider two neighboring data streams $S=(S_1,\cdots,S_n)$ and $S'=(S_1,\cdots,S_{i-1},S_{i+1},\cdots,S_n)$ for some $i\in[n]$. At step $i-1$, the state of the MG sketch on both inputs is exactly the same. $MG_S$ then receives the item $S_i$ while $MG_{S'}$ does not. This either increments one of the counters of $MG_S$ (possibly by adding an element and raising its counter from 0 to 1) or decrements all its counters. In $\ell_1$ distance, the vector of the counters thus changes by at most $k$. One can show by induction that this will stay this way: at any point in time, $\|MG_S - MG_{S'}\|_1 \le k$. By a standard global sensitivity argument, one can achieve pure DP by adding noise of magnitude $k/\varepsilon$ to each count. This is the approach used in [6]. Similarly, we could achieve $(\varepsilon,\delta)$-DP by using the Gaussian mechanism [9] with noise magnitude proportional to the $\ell_2$-sensitivity, which is $\sup_{S,S'}\|MG_S - MG_{S'}\|_2 \le \sqrt{k}$. We want to instead achieve noise with magnitude $O(1/\varepsilon)$ at each count. To this end, we need to exploit the structure of $MG_S - MG_{S'}$.

What we just described requires that we add the noise to the counts of all items in the universe, also to those that are not stored in the sketch. This results in the maximum error of all frequencies depending on the universe's size, which we do not want. However, it is known that this can be easily solved under $(\varepsilon,\delta)$-differential privacy by only adding noise to the stored items and then removing values smaller than an appropriately chosen threshold [14]. This may introduce additional error – for this reason, we end up with error $O(\log(1/\delta)/\varepsilon)$. As this is a somewhat standard technique, we ignore this in this section, we assume that the sketches $MG_S$ and $MG_{S'}$ store the same set of elements; the thresholding allows us to remove this assumption, while allowing us to add noise only to the stored items, at the cost of only getting approximate DP.

We now focus on the structure of $MG_S - MG_{S'}$. After we add to $MG_S$ the element $S_i$, it either holds (1) that $MG_S - MG_{S'}$ is a vector of all 0's and one 1 or (2) that $MG_S - MG_{S'} = -\mathbf{1}^k$ (We use $\mathbf{1}^k$ the denote the dimension $k$ vector of all ones). We show by induction that this will remain the case as more updates are done to the sketches (note that the remainders of the streams are the same). We do not sketch the proof here, as it is quite technical.

How do we use the structure of $MG_S - MG_{S'}$ to our advantage? We add noise twice. First, we independently add to each counter noise distributed as Laplace$(1/\varepsilon)$. Second, we add to all counters the same value, also distributed as Laplace$(1/\varepsilon)$. That is, we release $MG_S + \text{Laplace}(1/\varepsilon)^{\otimes k} + \text{Laplace}(1/\varepsilon)\mathbf{1}^k$ (For $D$ being a distribution, we use $D^{\otimes k}$ to denote the $k$-dimensional distribution consisting of $k$ independent copies of $D$). Intuitively speaking, the first noise hides the difference between $S$ and $S'$ in case (1) and the second noise hides the difference in case (2). We now sketch why this is so for worse constants: $2/\varepsilon$ in place of $1/\varepsilon$. When proving this formally, we use a more technical proof which leads to the better constant.

We now sketch why this is differentially private. Let $m_S$ be the mean of the counters in $MG_S$ for $S$ being an input stream. We may represent $MG_S$ as $(MG_S - m_S\mathbf{1}, m_S)$ (note that there is a bijection between this representation and the original sketch). We now argue that the $\ell_1$-sensitivity of this representation is $< 2$ (treating the representation as a $(k+1)$-dimensional vector for the sake of computing the $\ell_1$ distances). Consider the first case. In that case, the averages $m_S, m_{S'}$ differ by $1/k$. As such, $MG_S - m_S\mathbf{1}^k$ and $MG_{S'} - m_{S'}\mathbf{1}^k$ differ by $1/k$ at $k-1$ coordinates and by $1 - 1/k$ at one coordinate. The overall $\ell_1$ change of the representation is thus

$$(k-1)\cdot\frac{1}{k} + (1-1/k) + 1/k = 2 - 1/k < 2.$$

Consider now the second case when $MG_S - MG_{S'} = -\mathbf{1}^k$. Thus, $MG_S - m_S = MG'_S - m_{S'}$. At the same time $|m_S - m_{S'}| = 1$. This means that the $\ell_1$ distance between the representations is 1. Overall, the $\ell_1$-sensitivity of this representation is $< 2$.

This means that adding noise from Laplace$(2/\varepsilon)^{\otimes k+1}$ to this representation of $MG_S$ will result in $\varepsilon$-differential privacy. The resulting value after adding the noise is $(MG_S - m_S\mathbf{1}^k + \text{Laplace}(2/\varepsilon)^{\otimes k}, m_S + \text{Laplace}(2/\varepsilon))$. In the original vector representation of $MG_S$, this corresponds to $MG_S + \text{Laplace}(2/\varepsilon)^{\otimes k} + \text{Laplace}(2/\varepsilon)\mathbf{1}^k$ and, by post-processing, releasing this value is also differentially private. But this is exactly the value we wanted to show is differentially private!

## 3. PRELIMINARIES

*Setup of this paper.* We use $\mathcal{U}$ to denote a universe of elements. We assume that $\mathcal{U}$ is a totally ordered set of size $d$. That is, $\mathcal{U} = [d]$ where $[d] = \{1, \ldots, d\}$. Given a stream $S \in \mathcal{U}^{\mathbb{N}}$ we want to estimate the frequency in $S$ of each element of $\mathcal{U}$. Our algorithm outputs a set $T \subseteq \mathcal{U}$ of keys and a frequency estimate $c_i$ for all $i \in T$. The value $c_j$ is implicitly 0 for any $j \notin T$. Let $f(x)$ denote the true frequency of $x$ in the stream $S$. Our goal is to minimize the largest error between $c_x$ and $f(x)$ among all $x \in \mathcal{U}$.

*Differential privacy.* Differential privacy is a rigorous definition for describing the privacy loss of a randomized mechanism introduced by Dwork, McSherry, Nissim, and Smith [8]. Intuitively, differential privacy protects privacy by restricting how much the output distribution can change when replacing the input from one individual. This is captured by the definition of neighboring datasets. We use the add-remove neighborhood definition for differential privacy.

**Definition 1** (Neighboring Streams)**.** *Let $S$ be a stream of length $n$. Streams $S$ and $S'$ are neighboring denoted $S \sim S'$ if there exists an $i$ such that $S = (S'_1, \ldots, S'_{i-1}, S'_{i+1}, S'_{i+1}, \ldots, S'_{n+1})$ or $S' = (S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_n)$.*

**Definition 2** (Differential Privacy [9])**.** *A randomized mechanism $\mathcal{M} : \mathcal{U}^{\mathbb{N}} \to \mathcal{R}$ satisfies $(\varepsilon, \delta)$-differential privacy if and only if for all pairs of neighboring streams $S \sim S'$ and all measurable sets of outputs $Z \subseteq \mathcal{R}$ it holds that*

$$\Pr[\mathcal{M}(S) \in Z] \leq e^{\varepsilon}\Pr[\mathcal{M}(S') \in Z] + \delta \ .$$

The privacy guarantees are parameterized by the values $\varepsilon$ and $\delta$. Smaller values for these privacy parameters imply stronger privacy guarantees. Differential privacy gives us a trade-off between privacy and utility. If we require strong privacy guarantees we can lower the privacy parameters. However, doing so forces us to add more noise to the output.

Several other properties such as composition, group privacy, and closure under post-processing follow directly from the Definition 2. We refer to the book by Dwork and Roth [9] for a longer introduction to these concepts.

Samples from a Laplace distribution are used in many differentially private algorithms, most notably the Laplace mechanism [8]. We write Laplace$(b)$ to denote a random variable with a Laplace distribution with scale $b$ centered around 0. We sometimes abuse notation and write Laplace$(b)$ to denote the value of a random variable drawn from the distribution. Our mechanism also works with other noise distributions. We briefly discuss this in Section 5.2.

**Definition 3** (Laplace distribution)**.** *The probability density and cumulative distribution functions of the Laplace distribution centered around 0 with scale parameter $b$ are $f_b(x) = \frac{1}{2b}e^{-|x|/b}$, and $\Pr[\text{Laplace}(b) \leq x] = \frac{1}{2}e^{x/b}$ if $x < 0$ and $1 - \frac{1}{2}e^{-x/b}$ for $x \geq 0$.*

## 4. RELATED WORK

Chan et al. [6] show that the global $\ell_1$-sensitivity of a Misra-Gries sketch is $\Delta_1 = k$. (They actually show that the sensitivity is $k+1$ but they use a different definition of neighboring datasets that assumes $n$ is known. Applying their techniques under our definition yields sensitivity $k$.) They achieve privacy by adding noise with scale $k/\varepsilon$ to all elements in the universe and keep the top-$k$ noisy counts. This gives an expected maximum error of $O(k\log(d)/\varepsilon)$ with $\varepsilon$-DP for $d$ being the universe size. They use the algorithm as a subroutine for continual observation and merge sketches with an untrusted aggregator. Those settings are not a focus of our work but our algorithm can replace theirs as the subroutine.

Böhler and Kerschbaum [4] worked on differentially private heavy hitters with no trusted server by using secure multi-party computation. One of their algorithms adds noise to the counters of a Misra-Gries sketch. They avoid adding noise to all elements in the universe by removing noisy counts below a threshold which adds an error of $O(\log(1/\delta)/\varepsilon)$. This is a useful technique for hiding differences in keys between neighboring sketches that removes the dependency on $d$ in the error. Unfortunately, as we explain in the full version of our paper, their mechanism uses the wrong sensitivity. The sensitivity of the sketch is $k$. If the magnitude of noise and the threshold are increased accordingly the error of their approach is $O(k\log(k/\delta)/\varepsilon)$.

A closely related problem is that of implementing frequency oracles in the streaming setting under differential privacy. This has been studied in e.g. [11,16,17]. These approaches do not directly return the heavy hitters. The simplest approach for finding the heavy hitters is to iterate over the universe which might be infeasible. However, there are constructions for finding heavy hitters with frequency oracles more efficiently (see Bassily et al. [3]). However, as we discussed in the introduction, the approach of [3] leads to worse maximum error than what we get unless the sketch size is very large and the universe size is small.

# 5. DIFFERENTIALLY PRIVATE MISRA-GRIES SKETCH

In this section, we present our algorithm for privately releasing Misra-Gries sketches. We first present our variant of the non-private Misra-Gries sketch in Algorithm 1 and later show how we add noise to achieve $(\varepsilon, \delta)$-differential privacy. The algorithm we use differs slightly from most implementations of MG in that we do not remove elements that have weight 0 until we need to re-use the counter. This will allow us to achieve privacy with slightly lower error.

At all times, $k$ counters are stored as key-value pairs. We initialize the sketch with dummy keys that are not part of $\mathcal{U}$. This guarantees that we never output any elements that are not part of the stream, assuming we remove the dummy counters as post-processing.

The algorithm processes the elements of the stream one at a time. At each step one of three updates is performed: (1) If the next element of the stream is already stored the counter is incremented by 1. (2) If there is no counter for the element and all $k$ counters have a value of at least 1 they are all decremented by 1. (3) Otherwise, one of the elements with a count of zero is replaced by the new element.

In case (3) we always remove the smallest element with a count of zero. This allows us to limit the number of keys that differ between sketches for neighboring streams as shown in Lemma 5. The choice of removing the minimum element is arbitrary but the order of removal must be independent of the stream so that it is consistent between neighboring datasets. The limit on differing keys allows us to obtain a slightly lower error for our private mechanism. However, it is still possible to apply our mechanism with standard implementations of MG. We discuss this in Section 5.1.

**Algorithm 1:** Misra-Gries (MG)

**Input:** Positive integer $k$ and stream $S \in \mathcal{U}^{\mathbb{N}}$

$T \leftarrow \{d+1, \ldots, d+k\}$ // Start with $k$ dummy counters
$c_i \leftarrow 0$ for all $i \in T$
**foreach** $x \in S$ **do**
  **if** $x \in T$ **then** // Branch 1
    $c_x \leftarrow c_x + 1$
  **else if** $c_i \geq 1$ *for all* $i \in T$ **then** // Branch 2
    $c_i \leftarrow c_i - 1$ for all $i \in T$
  **else** // Branch 3
    Let $y \in T$ be the smallest key satisfying $c_y = 0$
    $T \leftarrow (T \cup \{x\}) \setminus \{y\}$
    $c_x \leftarrow 1$
**return** $T, c$

The same guarantees about correctness hold for our version of the MG sketch, as for the original version. This can be easily shown, as the original version only differs in that it immediately removes any key whose counter is zero. Since the counters for items not in the sketch are implicitly zero, one can see by induction that the estimated frequencies by our version are exactly the same as those in the original version. We still need this modified version, as the set of keys it stores is different from the original version, which we use below. The fact that the returned estimates are the same however allows us to use the following fact

**Fact 4** (Bose et al. [5])**.** *Let $\hat{f}(x)$ be the frequency estimates given by an MG sketch of size $k$ for $n$ being the input size. Then for all $x \in \mathcal{U}$, it holds $\hat{f}(x) \in [f(x) - n/(k+1), f(x)]$, where $f(x)$ is the true frequency of $x$.*

Note that this is optimal for any mechanism that returns a set of at most $k$ elements. This is easy to see for an input stream that contains $k+1$ distinct elements each with frequency $n/(k+1)$ since at least one element must be removed.

We now analyze the value of $MG_S - MG_{S'}$ for $S, S'$ being neighboring inputs (recall Definition 1). We will then use this in order to prove privacy. As mentioned in Section 4, Chan et al. [6] showed that the $\ell_1$-sensitivy for Misra-Gries sketches is $k$. They show that this holds after processing the element that differs for neighboring streams and use induction to show that it holds for the remaining stream. Our analysis follows a similar structure. We expand on their result by showing that the sets of stored elements for neighboring inputs differ by at most two elements when using our variant of Misra-Gries. We then show how all this can be used to get differential privacy with only a small amount of noise.

**Lemma 5.** *Let $T, c \leftarrow \mathrm{MG}(k, S)$ and $T', c' \leftarrow \mathrm{MG}(k, S')$ be the outputs of Algorithm 1 on a pair of neigboring streams $S \sim S'$ such that $S'$ is obtained by removing an element from $S$. Then $|T \cap T'| \geq k-2$ and all counters not in the intersection have a value of at most 1. Moreover, it holds that either (1) $c_i = c'_i - 1$ for all $i \in T'$ and $c_j = 0$ for all $j \notin T'$ or (2) there exists an $i \in T$ such that $c_i = c'_i + 1$ and $c_j = c'_j$ for all $j \neq i$.*

*Proof.* We sketch here the proof of the lemma. The full proof is in the full version of the paper.

Let $x = S_i$ be the element in stream $S$ which is not in stream $S'$. Since the streams are identical in the first $i-1$ steps the sketches are clearly the same before step $i$. If there is no counter for $x$ in the sketch and all $k$ counters have a non-zero value we execute Branch 2 of Algorithm 1. The difference between the MG sketches for $S$ and $S'$ then corresponds to case (1) of the lemma. If there is a counter for $x$ we execute Branch 1 of Algorithm 1. Finally, if there is no counter for $x$ and at least one counter with weight 0 we execute Branch 3 of Algorithm 1. After executing either Branch 1 or 3 the difference between the MG sketches corresponds to case (2) of the lemma. In the full proof we restrict the difference between the two sketches to one of six states that all fall under either case (1) or (2). We show that if we are in one of the states at step $j$ we will be in one of the states at step $j+1$ using a comprehensive case-by-case analysis. Since the difference between the sketches is in one of the states after step $i$ the lemma holds by induction. □

Next, we consider how to add noise to release the Misra-Gries sketch under differential privacy. Recall that Chan et al. [6] achieves privacy by adding noise to each counter which scales with $k$. We avoid this by utilizing the structure of sketches for neighboring streams shown in Lemma 5. We sample noise from Laplace$(1/\varepsilon)$ independently for each counter, but we also sample one more random variable from the same distribution which is added to all counters. Small values are then discarded using a threshold to hide differences in the sets of stored keys between neighboring inputs. This is similar to the technique used by e.g. [14]. The algorithm takes the output from MG as input. We sometimes write $\text{PMG}(k, S)$ as a shorthand for $\text{PMG}(\text{MG}(k, S))$.

**Algorithm 2:** Private Misra-Gries (PMG)

**Parameters:** $\varepsilon, \delta > 0$
**Input** : Output from Algorithm 1: $T, c \leftarrow \text{MG}(k, S)$

$\tilde{T} \leftarrow \emptyset$
Sample $\eta \sim \text{Laplace}(1/\varepsilon)$
**foreach** $x \in T$ **do**
  $c_x \leftarrow c_x + \eta + \text{Laplace}(1/\varepsilon)$
  **if** $c_x \geq 1 + 2\ln(3/\delta)/\varepsilon$ **then**
    $\tilde{T} \leftarrow \tilde{T} \cup \{x\}$
    $\tilde{c}_x \leftarrow c_x$
**return** $\tilde{T}, \tilde{c}$

We prove the privacy guarantees in three steps. First, we show that changing either a single counter or all counters by 1 does not change the output distribution significantly (Corollary 7). This assumes that, for neighboring inputs, the set of stored elements is exactly the same. By Lemma 5, we have that the difference between the sets of stored keys is small and the corresponding counters are $\leq 1$. Relying on the thresholding, we bound the probability of outputting one of these keys (Lemma 8). Finally, we combine these two lemmas to show that the privacy guarantees hold for all cases (we do this in Lemma 9).

**Lemma 6.** *Let us have $x, x' \in \mathbb{R}^k$ such that one of the following three cases holds*

1. *$\exists i \in [k]$ such that $|x_i - x'_i| = 1$ and $x_j = x'_j$ for all $j \neq i$.*
2. *$x_i = x'_i - 1$ for all $i \in [k]$.*
3. *$x_i = x'_i + 1$ for all $i \in [k]$.*

*Then we have for any measurable set $Z$ that*

$$\Pr[x + \text{Laplace}(1/\varepsilon)^{\otimes k} + \text{Laplace}(1/\varepsilon)\mathbf{1}^k \in Z] \leq e^{\varepsilon} \Pr[x' + \text{Laplace}(1/\varepsilon)^{\otimes k} + \text{Laplace}(1/\varepsilon)\mathbf{1}^k \in Z]$$

*Proof.* For the sake of brevity, we let $L = \text{Laplace}(1/\varepsilon)$. Throughout the proof, we construct sets by applying a translation to all elements of another set. That is, for any $\phi \in \mathbb{R}^k$ and measurable set $Z$ we define $Z - \phi = \{a \in \mathbb{R}^k | a + \phi \in Z\}$. We first focus on the simpler case (1). It holds by the law of total expectation that

$$\begin{aligned}\Pr[x + L^{\otimes k} + L\mathbf{1}^k \in Z] &= \\ \mathbb{E}_{N\sim L}\big[\Pr[L^{\otimes k} \in Z - x - N\mathbf{1}^k | N]\big] &\leq \\ e^{\varepsilon}\mathbb{E}_{N\sim L}\big[\Pr[L^{\otimes k} \in Z - x' - N\mathbf{1}^k | N]\big] &= \\ e^{\varepsilon}\Pr[x' + L^{\otimes k} + L\mathbf{1}^k \in Z]&\end{aligned}$$

where to prove the inequality, we used that for any measurable set $A$, it holds

$$\Pr[L^{\otimes k} \in A] \leq e^{\varepsilon}\Pr[L^{\otimes k} \in A - \phi]$$

for any $\phi \in \mathbb{R}^k$ with $\|\phi\|_1 \leq 1$ (see [8]). Specifically, we have set $A = Z - x - N\mathbf{1}^k$ and $\phi = x - x'$ such that $\|\phi\|_1 = 1$.

We now focus on the cases (2), (3). We will prove below that for $x, x'$ satisfying one of the conditions (2), (3) and for any measurable $A, Z$ and $N_1 \sim L^{\otimes k}$, it holds

$$\begin{aligned}\Pr[x + N_1 + L\mathbf{1}^k \in Z | N_1 \in A] \\ \leq e^{\varepsilon}\Pr[x' + N_1 + L\mathbf{1}^k \in Z | N_1 \in A]\end{aligned}$$

This allows us to argue like above:

$$\begin{aligned}\Pr[x + L^{\otimes k} + L\mathbf{1}^k \in Z] &= \\ \mathbb{E}_{N_1\sim L^{\otimes k}}\big[\Pr[x + N_1 + L\mathbf{1}^k \in Z | N_1]\big] &\leq \\ e^{\varepsilon}\mathbb{E}_{N_1\sim L^{\otimes k}}\big[\Pr[x' + N_1 + L\mathbf{1}^k \in Z | N_1]\big] &= \\ e^{\varepsilon}\Pr[x' + L^{\otimes k} + L\mathbf{1}^k \in Z]&\end{aligned}$$

which would conclude the proof. Let $g : \mathbb{R} \to \mathbb{R}^k$ be the function $g(a) = a\mathbf{1}^k$ and define $g^{-1}(B) = \{a \in \mathbb{R} | g(a) \in B\}$ and note that $g$ is measurable. We focus on the case (2); the same argument works for (3) as we discuss below. It holds

$$\begin{aligned}\Pr[x + N_1 + L\mathbf{1}^k \in Z | N_1 \in A] &= \\ \Pr[L\mathbf{1}^k \in Z - x - N_1 | N_1 \in A] &= \\ \Pr[L \in g^{-1}(Z - x - N_1) | N_1 \in A] &= \\ \Pr[L \in g^{-1}(Z - x' - \mathbf{1}^k - N_1) | N_1 \in A] &= \\ \Pr[L \in g^{-1}(Z - x' - N_1) - 1 | N_1 \in A] &\leq \\ e^{\varepsilon}\Pr[L \in g^{-1}(Z - x' - N_1) | N_1 \in A] &= \\ e^{\varepsilon}\Pr[L\mathbf{1}^k \in Z - x' - N_1 | N_1 \in A] &= \\ e^{\varepsilon}\Pr[x' + N_1 + L\mathbf{1}^k \in Z | N_1 \in A].&\end{aligned}$$

To prove the inequality, we again used the standard result that for any measurable $A$, $\Pr[L \in A] \leq e^{\varepsilon}\Pr[L \in A - 1]$ holds. The same holds for $A + 1$; this allows us to use the exact same argument in case (3), in which the proof is the same except that $-1$ on lines 4,5 of the manipulations is replaced by $+1$. □

**Corollary 7.** *Let $T, c$ and $T', c'$ be two sketches such that $T = T'$ and one of following holds:*

1. *$\exists i \in T$ such that $|c_i - c'_i| = 1$ and $c_j = c'_j$ for all $j \neq i$.*
2. *$c_i = c'_i - 1$ for all $i \in T$.*
3. *$c_i = c'_i + 1$ for all $i \in T$.*

*Then for any measurable set of outputs $Z$, we have:*

$$\Pr[\text{PMG}(T, c) \in Z] \leq e^{\varepsilon}\Pr[\text{PMG}(T', c') \in Z]$$

*Proof.* Consider first a modified algorithm PMG′ that does not perform the thresholding: that is, if we remove the condition on line 5. It can be easily seen that PMG′ only takes the vector $c$ and releases $c + \text{Laplace}(1/\varepsilon)^{\otimes k} + \text{Laplace}(1/\varepsilon)\mathbf{1}^k$. We have just shown in Lemma 6 that this means that for any measurable $Z'$,

$$\Pr[\text{PMG}'(T,c) \in Z'] \leq e^{\varepsilon} \Pr[\text{PMG}'(T',c') \in Z'].$$

Let $\tau(x) = x$ for $x \geq 1 + 2\ln(3/\delta)/\varepsilon$ and 0 otherwise. Since $\text{PMG}(T,c) = \tau(\text{PMG}'(T,c))$, it then holds

$$\Pr[\text{PMG}(T,c) \in Z] = \Pr[\text{PMG}'(T,c) \in \tau^{-1}(Z)] \leq$$
$$e^{\varepsilon} \Pr[\text{PMG}'(T',c') \in \tau^{-1}(Z)] = e^{\varepsilon} \Pr[\text{PMG}(T',c') \in Z]$$

as we wanted to show. □

Next, we bound the effect on the output distribution from keys that differ between sketches by $\delta$.

**Lemma 8.** *Let $T, c$ and $T', c'$ be two sketches of size $k$ and let $\hat{T} = T \cap T'$. If we have that $|\hat{T}| \geq k-2$, $c_i = c'_i$ for all $i \in \hat{T}$, and for all $x \notin \hat{T}$, it holds $c_x, c'_x \leq 1$. Then for any measurable set $Z$, it holds*

$$\Pr[\text{PMG}(T,c) \in Z] \leq \Pr[\text{PMG}(T',c') \in Z] + \delta$$

*Proof.* Let $\text{PMG}'(T,c)$ denote a mechanism that executes $\text{PMG}(T,c)$ and performs post-processing by discarding any elements not in $\hat{T}$. It is easy to see that $(a)$ $\Pr[\text{PMG}'(T,c) \in Z] = \Pr[\text{PMG}'(T',c') \in Z]$ since the input sketches are identical for all elements in $\hat{T}$. Moreover, for any output $\tilde{T}, \tilde{c} \leftarrow \text{PMG}(T,c)$ for which $\tilde{T} \subseteq \hat{T}$, the post-processing does not affect the output. This gives us the following inequalities: $(b)$ $\Pr[\text{PMG}(T,c) \in Z] \leq \Pr[\text{PMG}'(T,c) \in Z] + \Pr[\tilde{T} \not\subseteq \hat{T}]$ and $(c)$ $\Pr[\text{PMG}'(T',c') \in Z] \leq \Pr[\text{PMG}(T,c) \in Z] + \Pr[\tilde{T}' \not\subseteq \hat{T}]$. Combining equations $(a)-(c)$, we get the inequality $\Pr[\text{PMG}(T,c) \in Z] \leq \Pr[\text{PMG}(T',c') \in Z] + \Pr[\tilde{T} \not\subseteq \hat{T}] + \Pr[\tilde{T}' \not\subseteq \hat{T}]$.

As such, the Lemma holds if $\Pr[\tilde{T} \not\subseteq \hat{T}] + \Pr[\tilde{T}' \not\subseteq \hat{T}] \leq \delta$. That is, it suffices to prove that with probability at most $\delta$ any noisy count for elements not in $\hat{T}$ is at least $1 + 2\ln(3/\delta)/\varepsilon$. The noisy count for such a key can only exceed the threshold if one of the two noise samples added to the key is at least $\ln(3/\delta)/\varepsilon$. From Definition 3 we have $\Pr[\text{Laplace}(1/\varepsilon) \geq \ln(3/\delta)/\varepsilon] = \delta/6$. There are at most 4 keys not in $\hat{T}$ which are in $T \cup T'$ and therefore at most 6 noise samples affect the probability of outputting such a key (the 4 individual Laplace noise samples and the 2 global Laplace noise samples, one for each sketch). By a union bound the probability that any of these samples exceeds $\ln(3/\delta)/\varepsilon$ is at most $\delta$. □

We are now ready to prove the privacy guarantee of Algorithm 2.

**Lemma 9.** *Algorithm 2 is $(\varepsilon, \delta)$-differentially private for any $k$.*

*Proof.* The Lemma holds if and only if for any pair of neighboring streams $S \sim S'$ and any measurable set $Z$ we have:

$$\Pr[\text{PMG}(T,c) \in Z] \leq e^{\varepsilon} \Pr[\text{PMG}(T',c') \in Z] + \delta,$$

where $T, c \leftarrow \text{MG}(k, S)$ and $T', c' \leftarrow \text{MG}(k, S')$ denotes the non-private sketches for each stream.

We prove the guarantee above using an intermediate sketch that "lies between" $T, c$ and $T', c'$. The sketch has support $T'$ and we denote the counters as $\hat{c}$. By Lemma 5, we know that $|T \cap T'| \geq k-2$ and all counters in $c$ and not in $T \cap T'$ are at most 1. We will now come up with some conditions on $\hat{c}$ such that if these conditions hold, the lemma follows. We will then prove the existence of such $\hat{c}$ below. Assume that $\hat{c}_i = c_i$ for all $i \in T \cap T'$ and $\hat{c}_j \leq 1$ for all $j \in T' \setminus T$. Lemma 8 then tells us that

$$\Pr[\text{PMG}(T,c) \in Z] \leq \Pr[\text{PMG}(T',\hat{c}) \in Z] + \delta.$$

Assume also that one of the required cases for Corollary 7 holds between $\hat{c}$ and $c'$. We have

$$\Pr[\text{PMG}(T',\hat{c}) \in Z] \leq e^{\varepsilon} \Pr[\text{PMG}(T',c') \in Z].$$

Therefore, if such a sketch $T', \hat{c}$ exists for all $S$ and $S'$ the lemma holds since

$$\begin{aligned}\Pr[\text{PMG}(T,c) \in Z] &\leq \Pr[\text{PMG}(T',\hat{c}) \in Z] + \delta \\ &\leq e^{\varepsilon} \Pr[\text{PMG}(T',c') \in Z] + \delta \ .\end{aligned}$$

It remains to prove the existence of $\hat{c}$ such that $\hat{c}_i = c_i$ for all $i \in T \cap T'$ and $\hat{c}_j \leq 1$ for all $j \in T' \setminus T$ and such that one of the conditions $(1)-(3)$ of Corollary 7 holds between $\hat{c}$ and $c'$. We first consider neighboring streams where $S'$ is obtained by removing an element from $S$. From Lemma 5 we have two cases to consider. If $c_i = c'_i - 1$ for all $i \in T'$ we simply set $\hat{c} = c$. Recall that we implicitly have $c_i = 0$ for $i \notin T$. Therefore the sketch satisfies the two conditions above since $\hat{c}_i = c_i$ for all $i \in \mathcal{U}$ and condition (2) of Corollary 7 holds. In the other case where $c_i = c'_i + 1$ for exactly one $i \in T$ there are two possibilities. If $i \in T'$ we again set $\hat{c} = c$. When $i \notin T'$ there must exist at least one element $j \in T'$ such that $c'_j = 0$ and $j \notin T$. We set $\hat{c}_j = 1$ and $\hat{c}_i = c'_i$ for all $i \neq j$. In both cases $\hat{c}_i = c_i$ for all $i \in T \cap T'$ and $\hat{c}_j$ is at most one for $j \notin T$. There is exactly one element with a higher count in $\hat{c}$ than $c'$ which means that condition (1) of Corollary 7 holds.

If $S$ is obtained by removing an element from $S'$ the cases from Lemma 5 are flipped. If $c_i - 1 = c'_i$ for all $i \in T$ and $c'_j = 0$ for $j \notin T$ we set $\hat{c}_i = c_i$ if $i \in T$ and $\hat{c}_i = 1$ otherwise. It clearly holds that $\hat{c}_i = c_i$ for all $i \in T \cap T'$ and $\hat{c}_j \leq 1$ for all $j \notin T$. Since $\hat{c}_i = c'_i + 1$ for all $i \in T'$ condition (3) of Corollary 7 holds. Finally, if $c_i + 1 = c'_i$ for exactly one $i \in T'$ we simply set $\hat{c} = c$. $\hat{c}_i = c_i$ clearly holds for all $i \in T \cap T'$, $\hat{c}_j = 0$ for all $j \notin T$, and condition (1) of Corollary 7 holds between $\hat{c}$ and $c'$. □

Next, we analyze the error compared to the non-private sketch. We state the error in terms of the largest error among all elements of the sketch. Recall that we implicitly say that the count is zero for any element not in the sketch.

**Lemma 10.** *Let $\tilde{T}, \tilde{c} \leftarrow \text{PMG}(T,c)$ denote the output of Algorithm 2 for any sketch $T, c$ with $|T| = k$. Then with probability at least $1 - \beta$ we have*

$$\tilde{c}_x \in \left[ c_x - \frac{2\ln\left(\frac{k+1}{\beta}\right)}{\varepsilon} - 1 - \frac{2\ln\left(3/\delta\right)}{\varepsilon}, c_x + \frac{2\ln\left(\frac{k+1}{\beta}\right)}{\varepsilon} \right]$$

*for all $x \in T$ and $\tilde{c}_x = 0$ for all $x \notin T$.*

*Proof.* The two sources of error are the noise samples and the thresholding step. We begin with a simple bound on the absolute value of the Laplace distribution.

$$\Pr\left[|\text{Laplace}(1/\varepsilon)| \geq \frac{\ln((k+1)/\beta)}{\varepsilon}\right] =$$
$$2 \cdot \Pr\left[\text{Laplace}(1/\varepsilon) \leq -\frac{\ln((k+1)/\beta)}{\varepsilon}\right] = \beta/(k+1) \ .$$

Since $k+1$ samples are drawn we know by a union bound that the absolute value of all samples is bounded by $\ln((k+1)/\beta)/\varepsilon$ with probability at least $1-\beta$. As such the absolute error from the Laplace samples is at most $2\ln((k+1)/\beta)/\varepsilon$ for all $x \in T$ since two samples are added to each count. Removing noisy counts below the threshold potentially adds an additional error of at most $1 + 2\ln(3/\delta)/\varepsilon$. It is easy to see that $\tilde{c}_x = 0$ for all $x \notin T$ since the algorithm never outputs any such elements. □

**Theorem 11.** PMG$(k, S)$ *satisfies* $(\varepsilon, \delta)$*-differential privacy. Let* $f(x)$ *denote the frequency of any element* $x \in \mathcal{U}$ *in* $S$ *and let* $\hat{f}(x)$ *denote the estimated frequency of* $x$ *from the output of* PMG$(k, S)$*. For any* $x$ *with* $f(x) = 0$ *we have* $\hat{f}(x) = 0$ *and with probability at least* $1 - \beta$ *we have for all* $x \in \mathcal{U}$ *that*

$$\hat{f}(x) \in \left[f(x) - \frac{2\ln\left(\frac{k+1}{\beta}\right)}{\varepsilon} - 1 - \frac{2\ln(3/\delta)}{\varepsilon} - \frac{|S|}{k+1}, f(x) + \frac{2\ln\left(\frac{k+1}{\beta}\right)}{\varepsilon}\right]$$

*Moreover, the algorithm outputs all* $x$*, such that* $\hat{f}(x) > 0$ *and there are at most* $k$ *such elements.* PMG$(k, S)$ *uses* $2k$ *words of memory. For any fixed* $x \in \mathcal{U}$*, the mean squared error is* $\mathbb{E}[(\hat{f}(x) - f(x))^2] \leq 3\left(1 + \frac{2+2\ln(3/\delta)}{\varepsilon} + \frac{|S|}{k+1}\right)^2$.

*Proof.* The space complexity is clearly as claimed, as we are storing at any time at most $k$ items and counters. We focus on proving privacy and correctness.

If $f(x) = 0$ we know that $x \notin T$ where $T$ is the keyset after running Algorithm 1. Since Algorithm 2 outputs a subset of $T$ we have $\hat{f}(x) = 0$. The first part of the Theorem follows directly from Fact 4 and Lemmas 9 and 10.

We now bound the mean squared error. There are three sources of error. Let $r_1$ be the error coming from the Laplace noise, $r_2$ from the thresholding, and $r_3$ the error made by the MG sketch. Then

$$\begin{aligned}\mathbb{E}[(\hat{f}(x) - f(x))^2] =& \mathbb{E}[(r_1 + r_2 + r_3)^2] \\ \leq& 3(\mathbb{E}[r_1^2] + \mathbb{E}[r_2^2] + \mathbb{E}[r_3^2])\end{aligned}$$

by equivalence of norms (for any dimension $n$ vector $v$, $\|v\|_1 \leq \sqrt{n}\|v\|_2$). The errors $r_2, r_3$ are deterministically bounded $r_2 \leq 1+2\ln(3/\delta)/\varepsilon$ and $r_3 \leq |S|/(k+1)$. $\mathbb{E}[r_1^2]$ is the variance of the Laplace noise; we added two independent noises each with scale $1/\varepsilon$ and thus variance $2/\varepsilon^2$ for a total variance of $4/\varepsilon^2$. This finishes the proof. □

## 5.1 Privatizing standard versions of MG

The privacy of our mechanism as presented in Algorithm 2 relies on our variant of the Misra-Gries algorithm. Our sketch can contain elements with a count of zero. However, elements with a count of zero are removed in the standard version of the sketch. As such, sketches for neighboring datasets can differ for up to $k$ keys if one sketch stores $k$ elements with a count of 1 and the other sketch is empty. It is easy to change Algorithm 2 to handle these implementations. We simply increase the threshold to $1 + 2\ln\left(\frac{k+1}{2\delta}\right)/\varepsilon$ since the probability of outputting any of the $k$ elements with a count of 1 is bounded by $\delta$.

## 5.2 Tips for practitioners

Here we discuss some technical details to keep in mind when implementing our mechanism.

The output of the Misra-Gries algorithm is an associative array. In Algorithm 2 we add appropriate noise such that the associative array can be released under differential privacy. However, for some implementations of associative arrays such as hash tables the order in which keys are added affects the data structure. Using such an implementation naively violates differential privacy but it is easily solved either by outputting a random permutation of the key-value pairs or using a fixed order e.g. sorted by key.

We present our mechanism with noise sampled from the Laplace distribution. However, the distribution is defined for real numbers which cannot be represented on a finite computer. This is a known challenge and precision-based attacks still exist on popular implementations [13]. Since the output of MG is discrete the distribution can be replaced by the Geometric mechanism [12] or one of the alternatives introduced in [2]. Our mechanism would still satisfy differential privacy but it might be necessary to change the threshold in Algorithm 2 slightly to ensure that Lemma 8 still holds. Our proof of Lemma 8 works for the Geometric mechanism from [12] when increasing the threshold to $1 + 2\lceil\ln(6e^\varepsilon/((e^\varepsilon+1)\delta))/\varepsilon\rceil$.

Lastly, it is worth noting that the analysis for Lemma 8 is not tight. We bound the probability of outputting a small key by bounding the value of all relevant samples by $\ln(3/\delta)/\varepsilon$ which is sufficient to guarantee that the sum of any two samples does not exceed $2\ln(3/\delta)/\varepsilon$. This simplifies the proof and presentation significantly however one sample could exceed $\ln(3/\delta)/\varepsilon$ without any pair of samples exceeding $2\ln(3/\delta)/\varepsilon$. A tighter analysis would improve the constant slightly which might matter for practical applications.

# 6. PRIVATIZING MERGED SKETCHES

In practice, it is often important that we may merge sketches. This is for example commonly used when we have a dataset distributed over many servers. Each dataset consists of multiple streams in this setting, and we want to compute an aggregated sketch over all streams. We say that datasets are neighboring if we can obtain one from the other by removing a single element from one of the streams. If the aggregator is untrusted we must add noise to each sketch before performing any merges. This is the setting in [6] and we can run their merging algorithm. However, since we add noise to each sketch the error scales with the number of sketches. In particular, the error from the thresholding step of Algorithm 2 scales linearly in the number of sketches for worst-case input. In the full version of the paper we show how we can sometimes achieve smaller error in the setting where aggregators are trusted.


## Acknowledgment

The authors are supported by the VILLUM Foundation grant 16582. We thank Rasmus Pagh for suggesting this problem and for helpful discussions. We thank Martin Aumüller for his valuable feedback.

## 7. REFERENCES


[1] M. Aumüller, C. J. Lebeda, and R. Pagh. Representing sparse vectors with differential privacy, low error, optimal space, and fast access. *Journal of Privacy and Confidentiality*, 12(2), Nov. 2022.

[2] V. Balcer and S. Vadhan. Differential privacy on finite computers. *Journal of Privacy and Confidentiality*, 9(2), Sep. 2019.

[3] R. Bassily, K. Nissim, U. Stemmer, and A. Guha Thakurta. Practical locally private heavy hitters. *Advances in Neural Information Processing Systems*, 30, 2017.

[4] J. Böhler and F. Kerschbaum. Secure multi-party computation of differentially private heavy hitters. In Y. Kim, J. Kim, G. Vigna, and E. Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 2361–2377. ACM, 2021.

[5] P. Bose, E. Kranakis, P. Morin, and Y. Tang. Bounds for frequency estimation of packet streams. In J. F. Sibeyn, editor, *SIROCCO 10: Proceedings of the 10th Internaltional Colloquium on Structural Information Complexity, June 18-20, 2003, Umeå Sweden*, volume 17 of *Proceedings in Informatics*, pages 33–42. Carleton Scientific, 2003.

[6] T.-H. H. Chan, M. Li, E. Shi, and W. Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 140–159. Springer, 2012.

[7] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran. Differentially private summaries for sparse data. In *ICDT*, pages 299–311. ACM, 2012.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. pages 265–284, 2006.

[9] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[10] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath. The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1176–1184, 2015.

[11] B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. On the power of multiple anonymous messages. *IACR Cryptol. ePrint Arch.*, page 1382, 2019.

[12] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

[13] S. Haney, D. Desfontaines, L. Hartman, R. Shrestha, and M. Hay. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers. *CoRR*, abs/2207.13793, 2022.

[14] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180. ACM, 2009.

[15] J. Misra and D. Gries. Finding repeated elements. *Science of Computer Programming*, 2(2):143–152, 1982.

[16] R. Pagh and M. Thorup. Improved utility analysis of private countsketch. In *Advances in Neural Information Processing Systems*, volume 35, pages 25631–25643, 2022.

[17] F. Zhao, D. Qiao, R. Redberg, D. Agrawal, A. El Abbadi, and Y.-X. Wang. Differentially private linear sketches: Efficient implementations and applications. In *Advances in Neural Information Processing Systems*, volume 35, pages 12691–12704, 2022.