Pan-private Algorithms Via Statistics on Sketches

Darakhshan Mir, S. Muthukrishnan, Aleksandar Nikolov, Rebecca N. Wright
Department of Computer Science
Rutgers University, Piscataway, NJ 08854
{mir, muthu, anikolov, rebecca.wright}@cs.rutgers.edu

ABSTRACT

Consider fully dynamic data, where we track data as it gets inserted and deleted. There are well developed notions of private data analyses with dynamic data, for example, using differential privacy. We want to go beyond privacy, and consider privacy together with security, formulated recently as *pan-privacy* by Dwork et al. (ICS 2010). Informally, pan-privacy preserves differential privacy while computing desired statistics on the data, *even if the internal memory of the algorithm is compromised* (say, by a malicious break-in or insider curiosity or by fiat by the government or law).

We study pan-private algorithms for basic analyses, like estimating distinct count, moments, and heavy hitter count, with fully dynamic data. We present the first known pan-private algorithms for these problems in the fully dynamic model. Our algorithms rely on sketching techniques popular in streaming: in some cases, we add suitable noise to a previously known sketch, using a novel approach of calibrating noise to the underlying problem structure and the projection matrix of the sketch; in other cases, we maintain certain statistics on sketches; in yet others, we define novel sketches. We also present the first known lower bounds explicitly for pan privacy, showing our results to be nearly optimal for these problems. Our lower bounds are stronger than those implied by differential privacy or dynamic data streaming alone and hold even if unbounded memory and/or unbounded processing time are allowed. The lower bounds use a noisy decoding argument and exploit a connection between pan-private algorithms and data sanitization.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Privacy; H.2.8 [**Database Applications**]: Statistical Databases

General Terms

Algorithms, Security, Theory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
PODS'11, June 13–15, 2011, Athens, Greece.
Copyright 2011 ACM 978-1-4503-0660-7/11/06 ...$10.00.

Keywords

Differential Privacy, Pan-Privacy

1. INTRODUCTION

Consider the following simple, motivating example. Say we keep track of visitors that enter or leave a large facility (offline sites like a corporate or government office or online like websites). When queried, we wish to determine how many different visitors are on-site. This is a *distinct count* query. Unlike a data publication scenario where data is static after it is published, here the data is dynamic, varying over time, and the distinct count query may be posed any time, or even multiple times.

Our focus is first on privacy. Known methods for instance would be able to maintain the list of all IDs currently on site, and when queried, compute the precise answer D but return $D + \alpha$ for some suitable α that balances utility of the approximate distinct count against compromising the privacy of any particular ID. This intuitive approach has been formalized and a rich theory of (differential) privacy now exists for limitations and successes of answering this and many other queries privately.

Now, we go beyond privacy, and consider security. In particular, suppose the program—that tracks the data and answers the query—is compromised. Of course, this may happen because a malicious intruder hacks the system. But more subtly, this may happen because an insider with access, such as a systems administrator, may turn curious or crooked; data analysis may be outsourced to far away countries where people and laws are less stringent; or the contents of the registers may be subpoenaed by law or security officials. How can distinct count query processing be done securely, as well as with privacy? Maintaning a list of IDs on-site will not work, since it compromises all such IDs when a breach occurs. A natural idea is to hash (or encrypt) IDs into a new space that hides the identity. On a closer look, this too will not work since a breach will reveal the hash function or the encrypting key, and the intruder can exhaustively enumerate potential visitors to a site and determine the identity of all visitors currently on-site; this is known as a *dictionary attack*. (Notice that we are not limiting the intruder to have any computational constraints; however, even for computationally bounded adversaries, no cryptographic guarantees are known when the adversary has full access to the private key).

Maintaning a random sample of the IDs too will not work since it compromises the sampled IDs, and further, sample-based solutions are not known for estimating D with dy-

namic data when visitors arrive and depart, only for *partly dynamic* case when departure of visitors is not recorded. One can be principled and use well-known *sketches* since they only keep aggregate information (like counts, projections), rather than explicit IDs, and therefore afford natural obfuscation. While such solutions approximate distinct count well with dynamic data, they also do not work because they rely on hash functions to aggregate IDs: during the breach, the intruder obtains access to the hash functions, and can carry out a dictionary attack, compromising some of the IDs.

This example illustrates the issues involved when one seeks privacy and security simultaneously: even if we rely on cryptography and use exponential space or time to process the dynamic data, there are no known methods for even simple queries like distinct count. Of course, in reality, the dynamic data may have more attributes and many queries are of interest from estimating statistics like averages, to data mining tasks like finding heavy hitters, anomalies and others.

In this paper, we address such problems and develop both algorithmic and lower bound techniques. In order to do that, we need to formalize security and privacy. Typically, this is done by defining the limitations of an adversary and proving methods to be secure against one. In contrast, we are inspired by the information-theoretic approach of *differential privacy* and its recent extension *pan-privacy*, where the adversary has no computational or storage limitations.

We consider the *fully dynamic* setting in which for each user, represented by an ID i, (drawn from a universe $\mathcal{U}$), we maintain a *state* a_i, which consists of cumulative updates to i until time t. At each time step, the state of a single user is modified by incrementing or decrementing updates (in arbitrary integral values). In *partly dynamic* data, only increments are allowed. In addition, we call this *fully* or *partly streaming*, respectively, if the algorithms use sublinear space (typically, space polylogarithmic in various parameters).

We adopt the notion of differential privacy [5]. Informally, a randomized function f is *differentially private* with respect to the IDs if the probability distribution on the range of f is not sensitive to changing the state of any single user ID. To add security to privacy, Dwork et al. [7, 4] formalized the notion of *pan-privacy*. Informally, both the distribution of the internal states of the algorithm and the distribution of outputs should be insensitive to changing the state of a single user. This addresses privacy even in the case when there is one unannounced memory breach by the adversary. We study this model henceforth, and later, comment on variants of the model. Without some "secret state" (such as a secret set of hash or cryptographic keys), it might seem impossible to estimate statistics privately, but, surprisingly, Dwork et al. [7] showed that several interesting statistics on streams can be estimated accurately on *partly dynamic data*. Their algorithms are based on the technique of randomized response [16] and sampling.

1.1 Our Contributions

We design the first known pan-private algorithms for *distinct count*, *cropped first moment* and *heavy hitters count* for fully dynamic data. Our algorithms rely on *sketches* widely useful in streaming: in some cases, we add suitable noise using a novel approach of calibrating noise to the underlying problem structure and the projection matrix of the sketch; in other cases, we maintain certain statistics on sketches, and in yet others, we define novel sketches. In what follows, m is the size of the universe of IDs. These statistics, in one form or the other, have a long history, and are considered basic in data analysis tasks over dynamic data in the past few decades and different streaming solutions are known for these problems:

1. *Distinct Count* D: Given a sequence of updates, D is the number of user IDs with nonzero state: $D = |\{i \in \mathcal{U} : a_i \neq 0\}|$. We present an algorithm that is ε-pan private and outputs an estimate $(1+\alpha)D \pm$polylog with probability at least $1-\delta$, where polylog is a polylogarithmic function of various input parameters and m is the size of the universe. It directly uses a sketch known before based on stable distributions for estimating distinct count [1, 10], but maintains noisy versions based on a new method of adding noise tailored to the sketch and the underlying problem. What is surprising is that without the constraints of pan-privacy, this approach yields approximations for higher frequency moments $F_k = \sum_i a_i^k$ for $k = 1$ and 2 [10], but while we are able to derive pan-private distinct counts (related to F_0) using the same approach, it does not work for pan-private cropped moments such as $T_1(\tau) = \sum_i \min(a_i, \tau)$ or $T_2(\tau) = \sum_i \min(a_i, \tau)^2$ or other cropped moments.
   We complement this result by showing lower bounds. Let $\mathcal{A}$ be an online (not necessarily streaming) algorithm that outputs $D \pm o(\sqrt{m})$ with small constant probability. Then we show that $\mathcal{A}$ is not ε-pan private for any constant ε. This is the first known lower bound explicitly for pan-private algorithms and the best such bound for the distinct count problem. In fact, we develop an approach to showing lower bounds that takes a copy of the memory by breaching the algorithm once, and then simulating the algorithm with random inputs in parallel with this seed memory like noisy decoding [3]. Our lower bound holds irrespective of the memory used by $\mathcal{A}$—even if the memory is $\Omega(m)$. Further, we show a lower bound of $(1+\alpha)D \pm \text{polylog}(1/\delta)$ for algorithms that succeed with probability $1-\delta$, essentially tight up to additive polylog terms with our pan-private algorithm.
2. *Cropped first moment* $T_1(\tau)$: We present a fully dynamic algorithm that is ϵ-pan private and outputs an estimate in $[1/2T_1(\tau) - O(\tau\sqrt{m}/\epsilon), 2T_1(\tau) + O(\tau\sqrt{m}/\epsilon]$. Using a prior technique, this guarantee can be improved to an estimate in $[1/2(1+\alpha)T_1 - O(\tau \log m/\epsilon), 2(1+\alpha)T_1 + O(\tau \log m/\epsilon)]$. Our solution is a new sketch for this problem that is linear *modulo* 2τ, an approach that is unusual in the streaming literature but helps reduce the error of our pan-private algorithm.
   The lower bounds for distinct counts above imply that no ϵ-pan private algorithm can estimate $T_1(\tau)$ to within $o(\tau\sqrt{m})$ additive error with small constant probability.
3. *Heavy Hitters Count* $\text{HH}(k)$ is defined as $\text{HH}(k) = |\{i : a_i \geq F_1/k\}|$. It is the number of IDs that have state that is at least a $1/k$ fraction of the total state over all IDs. We present a fully dynamic pan-private algorithm that returns an estimate in $[(1-\alpha)\,\text{HH}(k) - O(\sqrt{k}), \text{HH}(O(k^2)) + O(\sqrt{k})]$ (that is no worse than $O(k)$ approximation, up to additive errors). We obtain this algorithm by first observing that using our T_1 estimator and with $O(m)$ space, we can provide an

estimate $\mathrm{HH}(k) \pm O(\sqrt{m})$, and then using this on the space of all buckets in the Count-Min sketch [2] which uses much smaller space.

Once again a reduction from distinct counts establishes that no ϵ-pan private algorithm can estimate the k-heavy hitters count to within $o(\sqrt{k})$ additive error, even if it is allowed to output the count of arbitrarily light IDs with nonzero state.

We emphasize that all our algorithms work on fully dynamic data which has not been considered in pan-privacy before. Dwork et al. [7] provide pan-private algorithms for problems (1)-(3) for partly dynamic data. Our definitions of the problems we consider differ slightly from those in [7]: we consider distinct count instead of density, cropped sum instead of cropped mean, and a more standard definition of heavy hitters count. In all cases our definitions specify problems that are at least as hard to approximate as those in [7].

The algorithms presented in [7] are based on sampling and randomized response and do not work with fully dynamic data. This is why we had to develop alternative techniques based on maintaining statistics over sketches. Surprisingly, for both distinct counts and cropped sums, our algorithms provide estimates for fully dynamic data that match the best bounds from [7] for partly dynamic data (up to additive polylog factors for distinct counts, and multiplicative factor 2 for cropped sum). The hashing technique used in [7] to obtain a constant multiplicative approximation for distinct count and cropped sum has an implicit additive factor of $O(\log m)$ because of adding Laplacian noise linear in $\log m$, giving an approximation of $(1 \pm \alpha)D \pm O(\log m)$. In fact a pure multiplicative approximation of $1 \pm \alpha$, for any constant α, is prohibited by our lower bounds on distinct counts.

The pan-private estimation of heavy hitters count in [7] outputs an estimate $[\mathrm{HH}((1+\rho)k) - \alpha m, \mathrm{HH}(k/(1+\rho)) + \alpha m]$. The hashing technique in [7] discussed above cannot be directly applied to this problem because it could both decrease or increase the number of heavy hitters in different hash levels. In fact, our algorithm is based on a precise analysis of how hashing affects the heavy hitters count. We thus give the first constant additive error approximation for heavy hitters count for either partly or fully dynamic data.

No explicit lower bounds were previously known for pan-private algorithms with a single intrusion. Independent of our work, [12] study lower bounds for two-party differential privacy, where two parties performing an analysis on their joint data, want to keep each party's view of the protocol a differentially private function of the other's input. In this model, they show a lower bound of $\Omega(\sqrt{n}/\log n)$ for computing the Hamming distance of two n-bit vectors. This lower bound implies a lower bound on multi-pass pan-private algorithms for distinct count (as well as for related statistics), allowing a single intrusion in each of the multiple passes over the data. Developed independently of their work, our lower bounds use different methods, hold for algorithms for fully dynamic data that may be thought of as single pass algorithms with just one intrusion, and are stronger than the bounds implied by their work for the single pass scenario.

Finally, we make an intriguing observation. Pan-privacy does not require algorithms to have any computational or storage constraints; it only requires differential privacy and security against intrusion. In fact, our lower bounds hold against algorithms that can use unbounded storage and perform arbitrary computations per update. On the other hand, the pan-private algorithms for distinct count and heavy hitters we present here are actually streaming algorithms that use only polylogarithmic time per update and polylogarithmic space. This may be an artifact of the techniques we use. We leave it open to find problems for which pan-private algorithms exist that necessarily use large (say polynomial) space.

We start in Section 2 by introducing relevant definitions and notation. In Section 3, we present our pan-private algorithms by keeping statistics on sketches. In Section 4, we present our lower bounds. We conclude with additional discussion in Section 5.

2. BACKGROUND

In this section we introduce notation and definitions and recapitulate earlier work that we build on.

2.1 Model and Notation

We are given a universe $\mathcal{U}$, where $|\mathcal{U}| = m$. An *update* is defined as an ordered pair $(i, d) \in \mathcal{U} \times \mathbb{Z}$. Consider a semi-infinite sequence of updates $(i_1, d_1), (i_2, d_2), \ldots$; the *input* for all our algorithms consists of the first t updates, denoted $S_t = (i_1, d_1), \ldots, (i_t, d_t)$. The *state vector* after t updates is an m-dimensional vector $\mathbf{a}^{(t)}$, indexed by the elements in $\mathcal{U}$. (We omit the superscript when it is clear from the context.) The elements of the vector state vector $\mathbf{a} = \mathbf{a}^{(t)}$, store the cumulative updates to i: $a_i = \sum_{j:i_j=i} d_j$. Each a_i is referred to as the state of ID i. In the *partly dynamic* model, all updates are positive, i.e. $\forall j : d_j \geq 0$; in the *fully dynamic* model, updates can be both positive (*inserts*), i.e. $d_j \geq 0$, and negative (*deletes*), i.e. $d_j < 0$, but at any time, $a_i \geq 0$ (since deletes cannot exceed inserts). We assume an upper bound Z on the maximum absolute value of the state of any $i \in \mathcal{U}$, i.e. $a_i \leq Z$ at any time step.

2.2 Differential Privacy

Dwork et al. [5] define the notion of differential privacy that provides a guarantee that the probability distribution on the outputs of a mechanism is "almost the same," irrespective of whether or not an individual is present in the data set. Such a guarantee incentivizes participation of individuals in a database by assuring them of incurring very little risk by such a participation. To capture the notion of a user opting in or out, the sameness condition is defined to hold with respect to a neighbor relation; intuitively, two inputs are neighbors if they differ only in the participation of a single individual. For example, Dwork et al. defined datasets to be neighbors if they differ in a single row. Formally,

DEFINITION 1. *[5] A randomized function f provides ε-differential privacy with respect to a binary neighbor relation $\sim$, if for input data sets D, D' such that $D \sim D'$, and for all $Y \subseteq Range(f)$, $\Pr[f(D) \in Y] \leq \exp(\varepsilon) \times \Pr[f(D') \in Y]$.*

One mechanism that Dwork et al. [5] use to provide differential privacy is the *Laplacian noise method* which depends on the *global sensitivity* of a function:

DEFINITION 2. *[5] For $f : \mathcal{D} \to \mathbb{R}^d$, the global sensitivity of f is $GS_f = \max_{D \sim D'} ||f(D) - f(D')||_1$.*

THEOREM 1. *[5] For $f : \mathcal{D} \to \mathbb{R}$, mechanism $\mathcal{M}$ that adds independently generated noise drawn from $Lap(GS_f/\varepsilon)$ to the output preserves ε-differential privacy.*

Another, more general (though, not always computationally efficient) method of providing differential privacy is the so called *exponential mechanism* proposed by McSherry and Talwar [13]. This mechanism is parametrized by a "quality function" $q(\mathbf{x}, r)$ that maps a pair of an input data set $\mathbf{x}$ (a vector over some arbitrary real-valued domain) and candidate output r (again over an arbitrary range R) to a real valued "score." The mechanism selects an output with exponential bias in favor of high scoring outputs by sampling from the following *exponential distribution*: $\mu_\varepsilon(r) \propto \exp(\varepsilon q(\mathbf{x}, r))$. For discrete ranges, Gupta et al. [9] provide an analog of a theorem of McSherry and Talwar [13]. Let $\Delta_q = \max_{\mathbf{x}\sim\mathbf{y}, r} |q(\mathbf{x}, r) - q(\mathbf{y}, r)|$.

THEOREM 2. *[13] The exponential mechanism, when used to select an output $r \in R$, gives $2\varepsilon\Delta_q$-differential privacy. Let R^* be the subset of R achieving $q(\mathbf{x}, r) = \max_r q(\mathbf{x}, r)$, and $\mathcal{E}_q^\varepsilon$ be a value drawn from the exponential mechanism, then:*

$$\Pr[q(\mathbf{x}, \mathcal{E}_q^\varepsilon) < \max_r q(\mathbf{x}, r) - \ln(|R| / |R^*|)/\varepsilon - t/\varepsilon] \leq e^{-t}$$

We will use the exponential mechanism to design a new mechanism, which will help us derive pan-private estimates of norms from sketches (see Section 3.1).

Part of the usefulness of differential privacy lies in its resilience to various notions of composition. Next we present two composition theorems due to Dwork et al. [5] that will be useful in the remainder of this paper. The first composition result concerns the privacy of computing multiple differentially private functions of the same input.

THEOREM 3. *[5] Given mechanisms $\mathcal{M}_i$, $i \in [r]$ each of which provide ε_i-differential privacy, then the overall mechanism $\mathcal{M}$ that executes these r mechanisms with independent randomness and outputs the vector of their outputs, provides $\left(\sum_{i\in[r]} \varepsilon_i\right)$-differential privacy.*

The second composition result concerns composition of the neighbor relation. First we define the notion of ℓ-neighborhood, which is a binary relation induced by the neighbor relation.

DEFINITION 3. *Given a neighbor relation $\sim$, the ℓ-neighbor relation $\sim_\ell$ is defined as follows. Two input datasets D, D' are said to be* 1-neighbors*—i.e. $D \sim_1 D'$, if $D \sim D'$. For a natural number $\ell > 1$, D, D' are said to be ℓ-*neighbors*—i.e. $D \sim_\ell D'$ if $D \sim_{\ell-1} D'$ or there exists a dataset $D'' \sim D'$ such that $D'' \sim_{\ell-1} D$.*

Another way to think of ℓ-neighbors is as inputs that are linked by a path of length at most ℓ in the graph induced by the neighbor relation. Next we present a theorem of Dwork et al. formally showing that differential privacy is resilient to composition of the neighbor relation.

THEOREM 4. *[5] If a function f provides ε-differential privacy with respect to $\sim$, then f provides $\ell\varepsilon$-differential privacy with respect to $\sim_\ell$.*

2.3 Pan privacy

Pan privacy guarantees a participant that his/her risk of being identified by participating in a data set is very little even if there is an external intrusion on the internal state of the analyzing algorithm. Formally, consider two online sequences of updates $S = \big((i_1, d_1), \ldots, (i_t, d_t)\big)$ and $S' = \big((i'_1, d'_1), \ldots, (i'_{t'}, d'_{t'})\big)$ associated with state vectors $\mathbf{a}$ and $\mathbf{a}'$ respectively.

DEFINITION 4 (USER-LEVEL NEIGBORS). *S and S' are said to be (user-level)* neighbors *if there exists a (multi)set of updates in S indexed by $K \subseteq [t]$ that update the same ID $i \in \mathcal{U}$, and there exists a (multi)set of updates in S' indexed by $K' \subseteq [t']$ that updates some $j(\neq i) \in \mathcal{U}$ such that $\sum_{k\in K} d_k = \sum_{k\in K'} d'_k$ and for all other updates in S and S' indexed by $Q = [t] - K$ and $Q' = [t'] - K'$ respectively,*

$$\forall i \in \mathcal{U} \quad \sum_{k\in Q, s.t.\ i_k = i} d_k = \sum_{k\in Q', s.t.\ i'_k = i} d'_k.$$

Notice that in the definition above t and t' do not have to be equal because we allow the d_i's to be integers. The definition ensures that two inputs are neighbors if some of the occurrences of an ID in S is replaced by some other ID in S' and everything else stays the same except (a) the order may be arbitrarily different and (b) the updates can be arbitrarily broken up since they are not constrained to be 1's. The neighbor relation preserves the first frequency moment of the sequence of updates, considered to be public information. Also, the graph induced by the neighbor relation on any set of sequences with the same first frequency moment is connected.

Our notion of neighborhood is slightly different the definition of Dwork et al. [7] definition, where any two data streams S and S' are neighbors if they differ only in the presence or absence of any number of occurrences of any element $i \in \mathcal{U}$ (i.e. $\mathbf{a}$ and $\mathbf{a}'$ have hammind distance at most 1). Our definition ensures that two neighboring sequences of updates are of the same "length," in the sense that the sum of the updates over all items is the same for both S and S', that is, $\sum_{i=1}^{t} d_k = \sum_{i=1}^{t'} d'_k$. For this purpose, we constrain the sum of the updates of the occurrences of item i in S to be conserved when they are replaced by item j in S'. In our definition, the total weight of updates is public, but, still, an adversary cannot distinguish between appearances of ID i or ID j, even if the adversary knows all other appearances of all other IDs. This modified definition of neighborhood (with its modified notion of privacy) is necessary to make the sensitivity of the k-heavy hitters count bounded. Such a modification was not necessary in [7], as they used a non-standard (and easier to approximate) notion of k-heavy hitters. We emphasize that except for our heavy hitters algorithm, all our other algorithms are private both according to the definition of Dwork et al. and according to our definition of neighborhood.

We comment on the composability of our definition of neighborhood. Applying definition 3, we see that two sequences S and S' will be ℓ-neighbors if there exist (possibly multi) sets of ID's of cardinality ℓ: $\{i_1, i_2 \ldots i_\ell\}$ and $\{j_1, j_2, \ldots, j_\ell\}$ all from $\mathcal{U}$, such that some occurrences of each $i_k, 1 \leq k \leq \ell$ in S are replaced by some occurrences of $j_k \neq i_k, 1 \leq k \leq \ell$ in S'. There is no other restriction on the $j'_k s$; they may be all equal, different or any subset of these may be equal. Hence Theorem 4 is applicable to our definition of ℓ-neighbors.

DEFINITION 5 (USER-LEVEL PAN PRIVACY [7]).
Let **Alg** *be an algorithm. Let I denote the set of internal states of the algorithm, and let σ the set of possible output*

sequences. Then algorithm **Alg** *mapping input prefixes to the range $I \times \sigma$, is* pan-private (against a single intrusion) *if for all sets $I' \subseteq I$ and $\sigma' \subseteq \sigma$, and for all pairs of user-level neighboring data stream prefixes S and S'*

$$\Pr[\mathbf{Alg}(S) \in (I', \sigma')] \leq e^{\varepsilon} \Pr[\mathbf{Alg}(S') \in (I', \sigma')]$$

where the probability spaces are over the coin flips of the algorithm **Alg**.

2.4 Sketches and Stable Distributions

In this section we discuss previous work in sketch-based streaming.

DEFINITION 6. *[15] A distribution $\mathcal{S}(\mathrm{p})$ over $\mathbb{R}$ is said to be p-stable if there exists $p \geq 0$ such that for any n real numbers $b_1, \ldots, b_m$ and i.i.d. variables $Y_1, \ldots, Y_m$ with distribution $\mathcal{S}(\mathrm{p})$, the random variable $\sum_i b_i Y_i$ has the same distribution as the random variable $(\sum_i |b_i|^p)^{1/p} Y$, where Y is a random variable with distribution $\mathcal{S}(\mathrm{p})$.*

Examples of p-stable distributions are the Gaussian distribution, which is 2-stable, and the Cauchy distribution, which is 1-stable. Stable distributions have been used to compute the L_p norms of vectors ($L_p = (\sum_i a_i^p)^{1/p}$) in the streaming model [10, 1].

Let X be a matrix of random values of dimension $m \times r$, where each entry of the matrix $X_{i,j}$, $1 \leq i \leq m$, and $1 \leq j \leq r$, is drawn independently from $\mathcal{S}(\mathrm{p})$, with p as small as possible. The *sketch vector* $\mathrm{sk}(\mathbf{a})$ is defined as the dot product of matrix X^T with $\mathbf{a}$, so $\mathrm{sk}(\mathbf{a}) = X^T \cdot \mathbf{a}$. From the property of stable distributions we know that each entry of $\mathrm{sk}(\mathbf{a})$ is distributed as $(\sum_i |a_i|^p)^{1/p} X_0$, where X_0 is a random variable chosen from a p-stable distribution. The sketch is used to compute $\sum_i |a_i|^p$ for $0 < p < \alpha / \log Z$, from which we can approximate $D^{(t)}$ up to a $(1+\alpha)$ factor(See [1] for details). By construction, any $\mathrm{sk}(\mathbf{a})_j$ can be used to estimate L_p^p. Cormode et al. [1] and Indyk [10] obtain a low-space good estimator for $(\sum_i |a_i|^p)$ by taking the median of all entries $\left|\mathrm{sk}(\mathbf{a})_j\right|^p$ over j:

THEOREM 5. *If the continuous stable distribution is approximated by discretizing it to a grid of size $(\frac{mZ}{\alpha\delta})^{O(1)}$, the support of the distribution $\mathcal{S}(\mathrm{p})$ from which the values $X_{i,j}$ are drawn is truncated beyond $(mZ)^{O(1)}$, and $r = O(1/\alpha^2 \cdot \log(1/\delta))$, then with probability $1-\delta$,*

$$(1-\alpha)^p \,\mathrm{median}_j \left|\mathrm{sk}(\mathbf{a})_j\right|^p \leq \mathrm{median}\,|X_0|^p \,(\sum_i |a_i|^p) \leq (1+\alpha)^p \,\mathrm{median}_j \left|\mathrm{sk}(\mathbf{a})_j\right|^p$$

where $\mathrm{median}\,|X_0|^p$ *is the median of absolute values (raised to the power p) from a (truncated, discretized) p-stable distribution.*

We will use these details in Algorithm 1 in Section 3.1 to propose a pan-private algorithm for distinct counts.

3. PAN-PRIVATE ALGORITHMS FOR FULLY DYNAMIC DATA

In this section we present our pan-private algorithms that work for fully dynamic data. Our algorithms follow the outline:

- initialize a sketch to a noisy vector chosen from an appropriate distribution;
- update the sketch linearly (linearity may be over the real field, or modulo a real number); and
- compute a global statistic of the sketch.

The fact that, for all the algorithms, the state of the algorithm is a linear function of its input and the noisy initialization allows us to characterize the distribution of the state of the algorithm at any time step; this property is essential to both the privacy and utility analyses of our algorithms. While particular entries in the sketches may not be accurate approximations of the states of the user IDs, the global statistic computed at the end can be shown to be an accurate estimate of the desired value.

3.1 Distinct count

We use sketching based on stable distributions outlined in Section 2.4 to design an algorithm for pan-private estimation of the distinct count statistic $D^{(t)}$. We exploit the linearity property of the sketches by maintaining a noisy version of the sketch in order to achieve pan-privacy. Because the sketch is a linear function of the state vector, it is enough to add an initial noise vector drawn from the appropriate distribution. To do so without adding too much noise, we develop a new technique of adding noise calibrated to the underlying random projection matrix and the nature of the statistic we are computing, using the exponential mechanism of McSherry and Talwar [13]. As a consequence, while this mechanism, in general, is not computationally efficient, it provides us with a new framework for adding noise that is not "function oblivious." The established Laplace mechanism [5], that adds noise calibrated to the *global sensitivity* of the function, beyond being aware of the global sensitivity of the function is oblivious of the underlying structure of the problem. This is important for our application as the sensitivity of the stable distribution sketch can be very high due to the heavy tails of p-stable distributions for small p.

Next we describe the mechanism we use to draw the noise vector.

An initializing noise vector: We use the exponential distribution to generate a random noise vector that initializes the sketch. The sketch vector has dimension r; let us denote the i-th row of X as X_{i*} and the j-th column of X as X_{*j}.

We use the exponential mechanism of McSherry and Talwar with the following quality function q. If the true sketch vector is $\mathrm{sk}(\mathbf{a})$, then

$$q\left(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}\right) = -d\left(\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}\right),$$

where d is defined as:

$$d(\mathbf{z}) := \min \|\mathbf{c}\|_0 \quad \text{s.t.}$$
$$\mathbf{z} = X^T \mathbf{c}$$
$$\forall i \in [m] : c_i \in [-2Z, 2Z].$$

If the above program is infeasible, then $d(\mathbf{z}) = \infty$.

Given sketch vector $\mathrm{sk}(\mathbf{a})$, the mechanism picks a sketch $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$ from a distribution, μ_ϵ given by

$$\mu_\epsilon(\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}) \propto \exp\left(\frac{\epsilon}{4} q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}})\right).$$

Intuitively, the distance function d roughly measures the minimum number of items in the state vector $\mathbf{a}$, whose

entries need to be changed in order to get from $\mathrm{sk}(\mathbf{a})$ to $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$. This is used in the utility analysis.

Next, we need to compute the sensitivity Δ_q of q defined as $\Delta_q = \max_{\mathbf{x}\sim\mathbf{z},\mathbf{y}} |d(\mathbf{x},\mathbf{y}) - d(\mathbf{z},\mathbf{y})|$.

LEMMA 1. *For q as defined above, $\Delta_q \leq 2$.*

PROOF. If $\mathrm{sk}(\mathbf{a})$ and $\mathrm{sk}(\mathbf{a}')$ are the true sketch vectors for neighboring sequences of updates corresponding to state vectors $\mathbf{a}$ and $\mathbf{a}'$ respectively, then for some $i, j \in \mathcal{U}$, $i \neq j$, $\mathrm{sk}(\mathbf{a}') = \mathrm{sk}(\mathbf{a}) + c_i X_{i*} + c_j X_{j*}$, for some $c_i, c_j \in [-2Z, 2Z]$. Therefore,

$$\begin{aligned}\Delta_q &\leq \max_{\mathrm{sk}(\mathbf{a}),\mathrm{sk}(\mathbf{a}'),\mathbf{y}} |d(\mathrm{sk}(\mathbf{a}) - \mathbf{y}) - d(\mathrm{sk}(\mathbf{a}') - \mathbf{y})| \\ &\leq \max_{\mathrm{sk}(\mathbf{a}),c_i,c_j,\mathbf{y}} |d(\mathrm{sk}(\mathbf{a}) - \mathbf{y}) \\ &\qquad - d(\mathrm{sk}(\mathbf{a}) - \mathbf{y} + c_i X_{i*} + c_j X_{j*})| \\ &\leq 2.\end{aligned}$$

□

Let $B = \mathbf{poly}(m, Z)$ be large enough so that: (1) Theorem 5 holds, (2) for any $c \in [-2Z, 2Z]^m$, $X^T c \in [-B, B]^r$. We pick an initializing vector $\mathbf{y}$ using the exponential distribution with quality function q from the range $\mathcal{R} = [-B, B]^r \cap \langle X_{1*}, \ldots, X_{m*} \rangle$, discretized to within $\mathbf{poly}(m, Z, 1/\alpha, 1/\delta)$ precision, again so Theorem 5 holds. Notice that $\log \mathcal{R} = O(r \cdot \log(\mathbf{poly}(m, Z, 1/\alpha, 1/\delta)))$, which implies that $\log \mathcal{R} = \mathbf{poly}(\log m, \log Z, 1/\varepsilon, 1/\alpha, \log(1/\delta))$.

The Algorithm: After initializing, we update and decode the sketch as in the non-private algorithm. Before outputting the final answer, we draw another vector using the exponential mechanism with the same parameters. The algorithm is shown below as Algorithm 1.

Since updates are linear, and $q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}})$ is a function of $\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$, initializing the sketch to a vector picked using the exponential mechanism with quality function $q(\mathbf{y}, 0) = -d(\mathbf{y})$ ensures that any state is $2\frac{\epsilon}{4}\Delta_q$-differentially private. More formally, from Theorems 2 and 3, and Definition 5:

LEMMA 2. *At any step in Algorithm 1, the state of the algorithm is a sketch and the distribution over states is given by the exponential mechanism with quality function*

$$q(\mathrm{sk}(\mathbf{a}), \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}) = -d(\mathrm{sk}(\mathbf{a}) - \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}).$$

Hence the algorithm is ϵ-pan private.

Also, by simple application of Theorem 2:

LEMMA 3. *The initializing vector $\mathbf{y}$ has $d(\mathbf{y}) \leq 4\frac{\log|\mathcal{R}|}{\epsilon} + \frac{4}{\epsilon}\log 1/\delta \leq \mathrm{polylog}(m, Z, 1/\epsilon, \log(1/\delta), 1/\alpha)$ with probability $1-\delta$. The same holds for $\mathbf{y}'$*

THEOREM 6. *With probability $1-\delta$, Algorithm 1 outputs an estimate in $(1 \pm \alpha)D^{(t)} \pm \mathbf{poly}(\log m, \log Z, \frac{1}{\varepsilon}, \frac{1}{\alpha}, \log\frac{1}{\delta})$.*

PROOF. Follows by the previous lemma, the definition of d, the fact that $\|\mathbf{a}\|_0 - \|\mathbf{c}\|_0 \leq \|\mathbf{a}+\mathbf{c}\|_0 \leq \|\mathbf{a}\|_0 + \|\mathbf{c}\|_0$, Theorem 5 and the linearity property of sketches: $\mathrm{sk}(\mathbf{a}) \pm \mathrm{sk}(\mathbf{b}) = \mathrm{sk}(\mathbf{a} \pm \mathbf{b})$. □

Algorithm 1 is a streaming algorithm since it uses space polylogarithmic in m and takes time polylogarithmic in m per new update.

Algorithm 1 Pan-private approximation of $D^{(t)}$

INPUT: privacy parameter ε, $0 < p < \alpha/Z < 1$, matrix X computed off-line See [1] for converting this to the on-line setting using seeded pseudorandom constructions., $\mathrm{sf}(p) = \mathrm{median}\,|X_0|^p$ also computed off-line numerically.

Initialize the r-dimensional sketch vector $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}$ to $\mathbf{y}$, by picking $\mathbf{y}$ from μ_ε
for all tuples (i, d_t) **do**
  for all $j = 1$ to r **do**
    $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j \leftarrow \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j + d_t * X_{ij}$
  end for
end for

OUTPUT: Draw r-dimensional vector $\mathbf{y}'$ from μ_ε, assign $\mathrm{sk}(\mathbf{a})^{\mathrm{priv}} \leftarrow \mathrm{sk}(\mathbf{a})^{\mathrm{priv}} + \mathbf{y}'$.
return $\tilde{\mathcal{D}} = \mathrm{median}_j \left(\left| \mathrm{sk}(\mathbf{a})^{\mathrm{priv}}{}_j \right|^p \right) \cdot \mathrm{sf}(p)$

Since we use the exponential mechanism, our techniques are not efficient in general. We need to sample from a space of 2^S different possible sketches, where S is the maximum bit size of a sketch. When S is polylogarithmic, we need to sample from a quasipolinomial set of objects. Note that a noise vector is only drawn during the preprocessing and postprocessing phases of the algorithm. While these phases take time 2^s, the time per update is only polylogarithmic.

A general noise-calibrating technique for sketches. The construction above gives a more general "recipe." Assume that a function f from state vectors to the reals ($f : [-Z, Z]^{\mathcal{U}} \to \mathbb{R}$) with $f(0) = 0$ can be approximated by a sketch. More precisely, the sketch is given by a linear map L and there exists a procedure that given the sketch outputs $\tilde{f}(\mathbf{a}) \in [\gamma_1 f(\mathbf{a}), \gamma_2 f(\mathbf{a})]$. Then we can use the technique above with $d(\mathbf{z}) = \min\{f(\mathbf{c}) : L\mathbf{c} = \mathbf{z}\}$, where the minimum is over valid differences of state vectors, i.e. $\mathbf{c} \in [-2Z, 2Z]^{\mathcal{U}}$. By identical proofs to the ones above, the algorithm is $\epsilon/2\Delta_q$-pan private and computes an approximation of f in $[\gamma_1 f(\mathbf{a}) - O(S), \gamma_2 f(\mathbf{a}) + O(S)]$, where S is a bound on the bitsize of a sketch, provided that $f(\mathbf{a}+\mathbf{y}) \in f(\mathbf{a}) \pm f(\mathbf{y})$. Note also that $\Delta_q = \max_{\mathbf{y}:\|\mathbf{y}\|_0=1} |f(\mathbf{y})|$, where $\mathbf{y}$ has one nonzero component, and that component is bounded in $[-2Z, 2Z]$.

In particular, a variant of Theorem 6 can be easily achieved for pan-private computation of L_1 and L_2. However, for both L_1 and L_2, this results in an additive factor that is linear in Z, the upper bound on each $|a_i|$. This is because for L_1 or L_2, the sensitivity of the quality function is $\Delta_q = 2Z$ (where d minimizes $||c||_1$ and $||c||_2$, respectively, instead of $||c||_0$) and we need to sample the noise vector from $\mu_{\varepsilon'}$, where $\varepsilon' = \varepsilon/2Z$. In turn, this results in linear dependence on Z in the bound on $d(\mathbf{y})$. The linear dependence is inherent in trying to estimate L_1 and L_2, due to their high sensitivity.

3.2 Cropped First Moment

In this section, we approximate $T_1(\tau)$ using sketches that are linear modulo an appropriately chosen parameter, to be specified later. The difficulty in approximating $T_1(\tau)$ for fully dynamic data is the apparent need to keep counters with range $[0, Z]$ for items, where Z is an upper bound on a_i (because the counters can race up to Z during intermedi-

ate stages and later get decremented to less than τ, so one has to keep track of the counter even when it goes far past τ for fully dynamic data). Such an approach results in error that scales linearly with Z, while the sensitivity of $T_1(\tau)$ is only τ, i.e. independent of Z. A natural workaround is to use modular counters, for example to use counters that estimate $a_i \bmod \tau$. However, such counters cannot distinguish between $a_i = 1$ and $a_i = \tau + 1$, and result in an estimate that is no better than a random guess. We show that if we scale a_i randomly between a_i and $2a_i$, then in expectation the modular counters provide an accurate approximation to $T_1(\tau)$. To the best of our knowledge, this approach to modular sketching is new, being motivated by the challenges of pan-private approximation in the fully dynamic data model. The modular counter technique allows us to show accuracy guarantees that are independent of Z.

For any $i \in \mathcal{U}$, let w_i be a real number independently and uniformly sampled from the interval $[1, 2]$. Define:

$$T'(\tau) = \sum_{i \in \mathcal{U}} w_i a_i \bmod 2\tau.$$

In analyzing the relation of $T'(\tau)$ to $T(tau)$, the following techical claim is useful:

CLAIM 1. *Let $a \geq \tau$ and let w be uniformly distributed in $[1, 2]$. Then $\mathbb{E}[wa \bmod 2\tau] \geq \tau/2$.*

PROOF SKETCH. Intuitively, because a is large, wa is supported on a constant fraction of the range $[0, 2\tau)$. Therefore, $\mathbb{E}[wa]$ is high. The full proof appears in Appendix A.1. □

LEMMA 4. *Assume that $\forall i \in \mathcal{U} : a_i \geq 0$. It follows that,*

$$\frac{1}{2} T_1(\tau) \leq \mathbb{E}[T'(\tau)] \leq 2T_1(\tau). \tag{1}$$

PROOF. Let us break down $T_1(\tau)$ and $T'(\tau)$ into partial sums. Define $A = \sum_{i:a_i<\tau} a_i$, $A' = \sum_{i:a_i<\tau} w_i a_i \bmod 2\tau$, $B = \sum_{i:a_i \geq \tau} \tau = |i : a_i > \tau|\tau$, and $B' = \sum_{i:a_i \geq \tau} w_i a_i \bmod 2\tau$. By definition, $T_1(\tau) = A + B$ and $T'(\tau) = A' + B'$.

Note that for $0 \leq a_i < \tau$, $w_i a_i \bmod 2\tau = w_i a_i$ since $w_i a_i < 2\tau$. Therefore, $\mathbb{E}[A'] = 3/2A$.

Next we compute $\mathbb{E}[B']$ and compare it to B. Notice first that $w_i a_i \bmod 2\tau < 2\tau$, and, therefore, $B' \leq 2B$.

Claim 1 provides a lower bound B' in terms of B. In particular, Claim 1 implies that $\mathbb{E}[B'] \geq 1/2B$.

The lemma follows from the bounds on $\mathbb{E}[A']$ and $\mathbb{E}[B']$.

□

Since $T'(\tau)$ is the sum of bounded independent random variables, Hoeffding's bound can be used to show that $T'(\tau) = (1 \pm \frac{1}{2})T_1(\tau) \pm O(\tau\sqrt{m})$ with high constant probability.

The next step is to estimate $T'(\tau)$ pan-privately. First, for technical reasons related to the noise distribution, we need to prove a variation of Lemma 4.

LEMMA 5.

$$\mathbb{E}\left[T'(\tau) - \sum_{w_i a_i \bmod 2\tau > 2\tau - 1} w_i a_i \bmod 2\tau\right] \geq \left(\frac{1}{2} - \frac{1}{\tau}\right) T_1(\tau).$$

PROOF SKETCH. We show that only the contribution of items i with $a_i \geq \tau$ is reduced and we bound the reduction. □

We are now ready to describe and analyze the algorithm. We first describe the noise distribution we use. Let $\mathcal{N}$ be the distribution given by the following density function:

$$f(x) = \begin{cases} \frac{e^\varepsilon}{2\tau - 1 + e^\varepsilon} & x \in [0, 1] \\ \frac{1}{2\tau - 1 + e^\varepsilon} & x \in (1, 2\tau) \end{cases} \tag{2}$$

This distribution corresponds to the following experiment: with probability $e^\varepsilon/(2\tau - 1 + e^\varepsilon)$ pick a uniform random value from $[0, 1]$; with probability $(2\tau - 1)/(2\tau - 1 + e^\varepsilon)$ pick a uniform random value from $(1, 2\tau)$.

The algorithm is shown as Algorithm 2.

Algorithm 2 Pan-private approximation of $T_1(\tau)$

INPUT: privacy parameter α, cropping parameter τ

for all $i \in \mathcal{U}$ **do**
  pick c_i independently from $\mathcal{N}$
  pick w_i independently and uniformly from $[1, 2]$
end for

for all tuples (i, d_t) **do**
  set $c_{i_t} := (c_{i_t} + w_{i_t} d_t) \bmod 2\tau$
end for

OUTPUT:
Set $\sigma = \sum_i c_i$, and $\tilde{\sigma} = \sigma + \mathrm{Lap}(2\tau/\epsilon)$
Set:

$$\tilde{T_1}(\tau) := \left(\tilde{\sigma} - \frac{\tau^2 m}{2\tau - 1 + e^\varepsilon}\right) \frac{2\tau - 1 + e^\varepsilon}{e^\varepsilon - 1} - \frac{m}{2}$$

return $\tilde{T_1}(\tau)$

THEOREM 7. *Algorithm 2 is 2ϵ-pan private. Further, with probability at least $2/3$,*

$$\left(\frac{1}{2} - \frac{1}{\tau}\right) T_1 - O(\tau\sqrt{m}/\varepsilon) \leq \tilde{T_1}(\tau) \leq 2T_1 + O(\tau\sqrt{m}\varepsilon).$$

PROOF SKETCH. Since the state of Algorithm 2 is a linear function of the input and the initial noise, to prove both privacy and utility, it is enough to consider the noise distribution when the state vector at query time is $\mathbf{a}$. Namely, observe that for any i with state a_i, c_i is distributed as $(\mathcal{N} + w_i a_i) \bmod 2\tau$. It can be shown that this distribution provides pan-privacy and has expectation bounded above by $T'(\tau)$ and below by $T'(\tau) - \sum_{w_i a_i \bmod 2\tau > 2\tau - 1} w_i a_i \bmod 2\tau$. Then Lemma 4, Lemma 5, and a Hoeffding bound finish the proof. □

Using the technique of multiple levels of hashing [7], the additive error can be reduced to $\tau\,\mathbf{poly}(\log m, \frac{1}{\varepsilon})$ at the cost of slightly increasing the multiplicative approximation factor. Also, Algorithm 2 can be used to approximate distinct counts by setting τ to a small constant greater than 2. However, this method provides a worse approximation than the one we achieve using stable distribution sketches.

3.3 Heavy Hitter Counts

We present a pan-private algorithm for heavy hitter count estimation with fully dynamic data by using the T_1 statistic over a structure inspired by the well known CM sketches [2]. We use the T_1 algorithm from Subsection 3.2 as a black box; in fact, any T_1 estimator that works with fully dynamic data suffices.

Recall that our T_1 estimator incurs a multiplicative approximation factor of 2 and an additive error $O(\sqrt{m}/\epsilon)$. As $\mathrm{HH}(k)$ is bounded by k, which can be assumed to be constant, the additive error term is prohibitive. The key step in our algorithm is to project the input S onto S' over a much smaller universe, so that S' has approximately the same k-heavy hitters count. In fact, we are able to reduce the universe size to a constant that depends only on k and the desired approximation guarantee. The reduced universe size directly implies a more accurate T_1 estimate and, hence, a more accurate estimate of the number of k-heavy hitters. Next we present our algorithm.

Assume the value $F_1 = F_1^{(t_0)}$, where t_0 is the time step when the algorithm is queried, is known ahead of time (this value is public by our definition of neighborhood). Assume also oracle access to a random function $f : [m] \to [h]$. Given a sequence of updates S, let $f(S)$ be the sequence $(f(i_1), d_1), \ldots, (f(i_t), d_t)$, and let $T_k(\tau|f)$ and $\tilde{T}_k(\tau|f)$ be, respectively, $T_k(\tau)$ and $\tilde{T}_k(\tau)$ computed on the stream $f(S)$. Note that $f(S)$ is a stream over the universe $[h]$ and can easily be simulated online given the oracle for f. Our algorithm is shown as Algorithm 3.

Algorithm 3 Pan-private approximation of $\mathrm{HH}(k)$

INPUT: privacy parameter ε, parameter k

Choose a random function $f : \mathcal{U} \to [h]$
Compute $x_1 = \tilde{T}_1(F_1/k|f)$ and $x_2 = \tilde{T}_1(F_1/ck|f)$ using Algorithm 2

OUTPUT: return

$$\tilde{\mathrm{HH}}(k) := (x_1 - x_2)\left(\frac{F_1}{k} - \frac{F_1}{ck}\right)^{-1}$$

Algorithm 3 is accurate provided that the function f approximately preserves the number of heavy hitters. Lemma 6 shows that a random f satisfies this condition with high probability.

LEMMA 6. *Let $f : \mathcal{U} \to [h]$ be a random function. Also, let $\tilde{k} = |\{j : \exists i \in h^{-1}(j) \text{ s.t. } a_i \geq t/k\}|$. With probability $1 - \delta$,*

$$\frac{\tilde{k}}{\mathrm{HH}(k)} \geq 1 - \frac{k}{\delta h}.$$

PROOF SKETCH. The proof is a standard balls-and-bins analysis. □

Lemma 7 shows that we can project the universe onto a significantly smaller universe without creating "new" heavy hitters.

LEMMA 7. *Let $A \subseteq \mathcal{U}$ be set of items s.t. $\forall i \in A : a_i \leq F_1\delta/2k^2$. Also, let $f : \mathcal{U} \to [h]$ be a pairwise-independent hash function. There exists an $h_0 = \Theta(k)$, s.t. for any $h \geq h_0$ with probability at least $1 - \delta$*

$$\forall j \in [h] : \sum_{i \in A \cap f^{-1}(j)} a_i \leq F_1/k.$$

PROOF SKETCH. The proof follows from a variance computation and Chebyshev's bound. □

We are now ready to analyze $\tilde{\mathrm{HH}}(k)$.

THEOREM 8. *$\tilde{\mathrm{HH}}(k)$ can be computed while satisfying 2ϵ-pan privacy. Moreover, for large enough $h = \Omega(k)$, with constant probability the following holds:*

$$\frac{1-\beta}{2}\,\mathrm{HH}(k) - O(\sqrt{k}/\varepsilon) \leq 2\tilde{\mathrm{HH}}(O(k^2)) + O(\sqrt{k}/\varepsilon)$$

PROOF. The privacy guarantee follows by the ϵ-pan privacy of the cropped sum estimators and the composition theorem. Next we analyze utility.

Let $N_j = \sum_{i \in f^{-1}(j)} a_i$. Computing $T_1(\tau)$ at two levels of τ provides an approximation of the number of heavy hitters:

$$\begin{aligned} &T_1(F_1/k) - T_1(F_1/ck) \\ &\quad= \sum_{j:N_j \geq F_1/ck} \min(N_j, F_1/k) - F_1/ck \\ &\quad= \sum_{j:N_j \geq F_1/k} (F_1/k - F_1/ck) \\ &\quad+ \sum_{j:F_1/ck \geq N_j \geq F_1/k} (N_j - F_1/ck) \end{aligned}$$

It immediately follows that $|\{j : N_j \geq F_1/k\}| < \mathbb{E}[\tilde{\mathrm{HH}}(k)] \leq |\{j : N_j \geq F_1/ck\}|$. By Lemma 6, $|\{j : N_j \geq F_1/k\}| \geq (1 - \beta)\,\mathrm{HH}(k)$ except with probability δ. Lemma 7 can be applied with $A = \{i : a_i \leq F_1\delta/2c^2k^2\}$. By the lemma, for every $j \in [h]$, it holds that $N_j \geq F_1/ck \Rightarrow \exists i \in f^{-1}(j)$ s.t. $a_i \geq F_1\delta/(2c^2k^2)$, except with probability δ. Therefore, $|\{j : N_j \geq F_1/ck\}| \leq \mathrm{HH}(c^2k^2/\delta)$. The proof then follows by the guarantees for $\tilde{T}_1$. □

As described, Algorithm 3 keeps only constantly many counters. However, the space complexity is at least linear, as the algorithm needs to keep $O(n \log k)$ random bits in order to evaluate a truly random function f. The number of random bits can be decreased by picking a function f from a family of bounded independence. Kane et al. [11] show that if f is picked from an r-wise independent family, where $r = \Omega(\log(k/\beta)/\log\log(k/\beta))$, then the lower bound on the ratio $\tilde{k}/\,\mathrm{HH}(k)$ from Lemma 6 decreases by at most a multiplicative factor of $1 - \beta$, with high probability. Notice also that for Lemma 7 only pairwise independence is required. Since a function from $\mathcal{U}$ to $[h]$ from an r-wise independent family can be represented by $O(r \log m + r \log k)$ bits and r only needs to be logarithmic in k, Algorithm 3 can be implemented using $O(k + \log k \log m/\log\log k)$ words.

4. LOWER BOUNDS

We present the first known lower bounds against pan-private algorithms that allow a single intrusion. We emphasize that these are the first lower bounds explicitly for pan-privacy with a single intrusion. The lower bounds were developed independently of the work of McGregor et al. [12] and use different methods. The lower bounds of McGregor

et al. imply lower bounds for pan-private algorithms that make multiple passes over the data and allow one intrusion in each pass. We consider only single pass algorithms, but our lower bounds are stronger than those of McGregor et al. in the single pass model. We present a more concrete comparison at the end of this section. Our lower bounds hold even for partly dynamic data, and therefore also for fully dynamic data. We show that if only an additive approximation is allowed, the full space extension of prior work [7] for distinct count estimation is optimal. Thus, the multiplicative approximation factor in the analysis of our algorithm is necessary. Furthermore, our new noisy decoding theorem shows that our sketching algorithm gives an almost optimal bi-approximation guarantee. Interestingly, our lower bounds make no assumptions on the space or time complexity of the algorithm, and yet the (almost) optimal algorithm biapproximation happens to use polylogarithmic space.

Dinur-Nissim Style Decoding. Our lower bounds utilize a decoding algorithm of the style introduced in a privacy context by Dinur and Nissim [3]. Informally, we argue that the (private) state of an accurate pan-private algorithm can be thought of as a sanitization of the part of the input that was already processed. We then employ the noisy decoding results to show that if this sanitization is very accurate, then most of the input of the algorithm can be recovered by an adversary.

Next, we introduce the decoding results we will use.

THEOREM 9 ([3]). *Let $\mathbf{x} \in \{0,1\}^n$. For any ϵ and $n \geq n_\epsilon$, the following holds. Given $O(n \log^2 n)$ random strings $\mathbf{q}_1, \ldots, \mathbf{q}_t \in_R \{0,1\}^n$, and approximate answers $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_t$ s.t. $\forall i \in [t] : |\mathbf{x} \cdot \mathbf{q}_i - \tilde{\mathbf{a}}_i| = o(\sqrt{n})$, there exists an algorithm that outputs a string $\tilde{\mathbf{x}} \in \{0,1\}^n$ and except with negligible probability $||\mathbf{x} - \tilde{\mathbf{x}}||_0 \leq \epsilon n$.*

In follow up work, [6] strengthened the above and showed that decoding is possible even when a constant fraction of the queries are inaccurate.

THEOREM 10 ([6]). *Given $\rho < \rho^*$, where $\rho*$ is a constant approximately equal to 0.239, there exists a constant ϵ s.t. the following holds. Let $\mathbf{x} \in \{0,1\}^n$. There exists a matrix $A \in \{-1,1\}^{n \times m}$ for some $m = O(n)$ and an efficient algorithm $\mathcal{A}$, s.t. on input $\tilde{b} \in \mathbb{N}^m$, satisfying $|\{i : |(A\mathbf{x} - \tilde{b})_i| > \alpha\}| \leq \rho$, $\mathcal{A}$ outputs $\tilde{\mathbf{x}} \in \{0,1\}^n$ and with probability $1 - e^{-O(m)}$, $||\mathbf{x} - \tilde{\mathbf{x}}||_0 \leq \epsilon\alpha^2$*

Next, we prove a result that is similar to Dinur and Nissim's but uses "union queries" rather than dot product queries.

THEOREM 11. *Let $\mathbf{x} \in \{0,1\}^n$, $||\mathbf{x}||_0 \leq L$ for some $L = L(n)$. For any ϵ and $n \geq n_\epsilon$, there exist $n^{O(L)}$ binary strings $\mathbf{q}_1, \ldots, \mathbf{q}_t \in \{0,1\}^n$ and an algorithm $\mathcal{A}$ such that given answers $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_1$ satisfying*

$$\forall i : (1-\alpha_1)||\mathbf{x} + \mathbf{q}_i||_0 - \alpha_2 \leq \tilde{\mathbf{a}}_i \leq (1+\alpha_1)||\mathbf{x} + \mathbf{q}_i||_0 + \alpha_2$$

for $\alpha_2 = o(L)$, $\mathcal{A}$ outputs $\tilde{\mathbf{x}}$ with $||\mathbf{x} - \tilde{\mathbf{x}}||_0 \leq \frac{16(\alpha_1+\epsilon)}{1-\alpha_1} L$.

PROOF SKETCH. The proof follows the outline of Dinur and Nissim's arguments for lower bounds against exponentially many queries [3]. □

Lower Bounds from Noisy Decoding. Our approach is to consider pan-private algorithms as sanitization algorithms that respect specific restrictions. The private state revealed to the adversary at the time of intrusion can be thought of as a sanitization of the part of the input that was processed before the time of intrusion. The adversary is then allowed to ask any query that can be encoded as adding more input and asking for the final answer of the function computed by the algorithm on the concatenated inputs. Using noisy decoding results, we can give noise lower bounds for this sanitization setting which then imply lower bounds for pan-private algorithms.

Another point of view is that the sanitization setting with restricted queries described above can be seen as a one-way two-party differentially private protocol, i.e. a one-way restriction of the model defined by Mironov et al. [14]. Then our lower bounds can be thought of as lower bounds for one-way two-party differential privacy. Since pan-private algorithms give one-way two-party differentially private protocols in the same way in which streaming algorithms give one-way communication protocols in the communication complexity setting, the lower bounds for pan-private algorithms follow.

We introduce our approach using the most direct argument first: a lower bound for the dot product problem.

PROBLEM 1. *Input is a sequence S_t of updates followed by a sequence S'_t.*

Output: Let $\mathbf{a}$ be the state of sequence S_t, and let $\mathbf{a}'$ be the state of S'_t. Output $\mathbf{a} \cdot \mathbf{a}' \pm \alpha = \sum_{i \in \mathcal{U}} a_i a'_i \pm \alpha$, where α is an approximation factor.

THEOREM 12. *Let $\mathcal{A}$ be a streaming algorithm that on input streams S_t, S'_t outputs $\mathbf{a} \cdot \mathbf{a}' \pm o(\sqrt{m})$ with probability at least $1 - O(m^{-2})$. Then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ.*

PROOF. Fix an input sequence S_t s.t. $\forall i \in \mathcal{U} : a_i \in \{0,1\}$. Let X be the internal state of the algorithm $\mathcal{A}$ after processing S_t. By the definition of pan privacy, X is ϵ-differentially private with respect to S_t. Fix some constants δ and η. We will show that for all large enough m, any algorithm $\mathcal{Q}$ that takes as input X and a stream S'_t and outputs $\mathbf{a} \cdot \mathbf{a}' \pm o(\sqrt{m})$ with probability at least $1 - O(m^{-2})$ can be used to compute a vector $\tilde{\mathbf{a}}$ such that $\tilde{a_i} = a_i$ for all but an η fraction of $i \in \mathcal{U}$ with probability $1 - \delta$. Therefore, the existence of such an algorithm $\mathcal{Q}$ implies that X cannot be ϵ-differentially private for any fixed ϵ. Indeed, assume for the sake of contradiction that an algorithm with the given properties exists and X is ϵ-differentially private. Since $\mathcal{Q}$ depends only on X and not on S_t, the output of $\mathcal{Q}$ is also ϵ-differentially private. This is a contradiction, since the output of $\mathcal{Q}$ can be used to guess a bit of the binary vector $\mathbf{a}$ accurately with probability at least $(1 - \delta - \eta)$, where δ and η can be chosen arbitrarily small. More formally, choose a component i of $\mathbf{a}$ uniformly at random. The event $E = \{\tilde{a_i} = a_i\}$ happens with probability at least $1 - \delta - \eta$ by a union bound. Now consider an input sequence S''_t which is a neighbor of S_t and $a''_i \neq a_i$. If S''_t were the input to $\mathcal{A}$, the event E would happen only if the vector $\tilde{\mathbf{a}}''$ computed from the output of $\mathcal{Q}$ disagrees with $\mathbf{a}''$ on i. By a union bound this happens with probability at most $\eta + \delta$. We get a contradiction with pan privacy as long as $(1 - \delta - \eta)/(\delta + \eta) \geq e^\epsilon$.

To finish the proof we show that an algorithm $\mathcal{Q}$ with the specified properties can be used to recover all but an η fraction of $\mathbf{a}$ with probability $1 - \delta$. To see this, observe that $\mathcal{Q}$ can be used to answer queries $\mathbf{a} \cdot \mathbf{q}$ for any arbitrary

$\mathbf{q}$ to within $o(\sqrt{m})$ additive error. In particular, to answer queries $\mathbf{a}\cdot\mathbf{q}_1,\ldots,\mathbf{a}\cdot\mathbf{q}_r$, run $\mathcal{Q}(X,S_t^{(1)}),\ldots,\mathcal{Q}(X,S_t^{(r)})$ in parallel, where $S_t^{(i)}$ is a stream with state $\mathbf{q}_i$. If $r=o(n^2)$, then, by the union bound, with probability $1-\delta$ for any constant δ, $\mathcal{Q}(X,S_t^{(i)})=\mathbf{a}\cdot\mathbf{q}_i\pm o(\sqrt{m})$ for all i. By Theorem 9, there exists an algorithm that, given the output of $\mathcal{Q}(X,S_t^{(1)}),\ldots,\mathcal{Q}(X,S_t^{(r)})$, outputs $\tilde{\mathbf{a}}$ s.t. except with negligible probability $\tilde{\mathbf{a}}$ agrees with $\mathbf{a}$ on all but η fraction of the coordinates. □

For the same problem, a recent and independently proved result by McGregor et al. [12] for two-party differential privacy, when interpreted to apply to dynamic data, would imply a lower bound on the additive error of $\Omega(\sqrt{m}/\log m)$. Thus, our lower bound improves the asymptotic additive term by a factor of $\log m$, and, unlike their bound, is tight, even for partly dynamic data.

We have the following corollary.

COROLLARY 1. *Let $\mathcal{A}$ be an online algorithm that on input S_t outputs $D^{(t)}\pm o(\sqrt{m})$ with probability at least $1-O(m^{-2})$. Then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ. Moreover, the same conclusion holds for $\mathcal{A}$ that outputs $T_1(\tau)\pm\tau o(\sqrt{m})$ with probability $1-O(m^{-2})$.*

PROOF. Notice that the proof of Theorem 12 goes through if we restrict the instances to be binary, i.e. if we require that $\forall i\in\mathcal{U}: a_i,a_i'\in\{0,1\}$. The corollary follows by a reduction from this restricted dot-product problem to the distinct elements problem. Given binary streams S_t', S_t'', let $S_t=(S_t',S_t'')$ be their concatenation. By a simple application of inclusion-exclusion, $D^{(t)}=D^{(t)}(S')+D^{(t)}(S'')-\mathbf{a}\cdot\mathbf{a}'$. Therefore, an ϵ-pan private algorithm for $D^{(t)}$ that achieves additive approximation α with probability $1-\delta$ implies a 3ϵ-pan private algorithm for dot product on binary instances that achieves additive approximation 3α with probability $1-3\delta$.

The statement for T_1 holds by the same reduction, but substituting binary instances with instances for which $\forall i\in\mathcal{U}: a_i,a_i'\in\{0,\tau\}$. □

A similar corollary holds for heavy hitters. Recall that $\mathrm{HH}(k)$ is the number of items i such that for which $a_i\geq F_1/k$.

COROLLARY 2. *Let $\mathcal{A}$ be an online algorithm that on input S_t outputs an estimate $\tilde{\mathrm{HH}}(k)\in[\mathrm{HH}(k)-o(\sqrt{k}),\mathrm{HH}(k')+o(\sqrt{k})]$ for some $k'\leq F_1$ with probability at least $1-O(m^{-2})$. Then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ*

PROOF IDEA. The proof is a reduction from the distinct count problem on sequences of updates with items drawn from the universe $[k]$, to the k-Heavy Hitters problem on sequences with items drawn from $\mathcal{U}$. The full details can be found in Appendix B. □

The next theorems follow by arguments identical to the one used to prove Theorem 12, but using, respectively, Theorem 10 and Theorem 11 in place of Theorem 9.

THEOREM 13. *Let $\mathcal{A}$ be an online algorithm that on inputs S_t, S_t' outputs $\mathbf{a}\cdot\mathbf{a}'\pm o(\sqrt{m})$ with probability at least $1-\delta$. If $\delta<\rho^*/2(1+\eta)$ for any η, then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ.*

PROOF. The proof is analogous to the proof of Theorem 12. Note first that the $\{-1,1\}$ queries of Theorem 10 can be simulated as the difference of two $\{0,1\}$ queries, which gives $o(\sqrt{m})$ additive error with probability at most $1-2\delta$. In order to apply Theorem 10, we need to guarantee that at most $\rho<\rho^*$ fraction of the queries answered by $\mathcal{Q}$ have error $\Omega(\sqrt{m})$. Call such queries *inaccurate*. In expectation, there are at most 2δ inaccurate queries. Since the statement of Theorem 10 holds when the queries are independent, an application of a Chernoff bound with a large enough number of queries shows that except with negligible probability there are at most ρ^* inaccurate queries. After applying Theorem 10, the proof can be finished analogously to the proof of Theorem 12. □

COROLLARY 3. *Let $\mathcal{A}$ be an online algorithm that on input S outputs $D^{(t)}(S)\pm o(\sqrt{m})$ with probability at least $1-\delta$. If $\delta<\rho^*/6(1+\eta)$, then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ. Moreover, the same conclusion holds for $\mathcal{A}$ that outputs $T_1(\tau)\pm\tau o(\sqrt{m})$ with probability $1-\delta$, for $\delta<\rho^*/(6(1+\eta))$.*

This corollary implies the optimality of the full-space extensions of the *partly dynamic* algorithms for distinct count and $T_1(\tau)$ of Dwork et al. [7]. Furthermore, it establishes that the our fully dynamic (full space) $T_1(\tau)$ algorithm presented in Section 3.2 is almost optimal, except for a constant multiplicative factor.

The corresponding lower bound for k-Heavy Hitters follows by the reduction from distint counts problem in the proof of Corollary 2.

COROLLARY 4. *Let $\mathcal{A}$ be an online algorithm that on input S_t outputs an estimate $\tilde{\mathrm{HH}}(k)\in[\mathrm{HH}(k)-o(\sqrt{k}),\mathrm{HH}(k')+\sqrt{k}]$ for some $k'\leq F_1$ with probability at least $1-O(\delta)$. If $\delta<\rho^*/6(1+\eta)$, then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ.*

This result implies that our k-Heavy Hitters algorithm in Section 3.3 is almost optimal, up to an arbitrarily small multiplicative factor.

Using similar arguments and utilizing Theorem 11, we can show the following (proof omitted).

THEOREM 14. *Let $\mathcal{A}$ be a streaming algorithm that on input a stream S_t and any constant α outputs $(1\pm\alpha)D^{(t)}\pm o(\log\frac{1/\delta}{\log m})$ with probability at least $1-\delta$. Then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ. Moreover, the same conclusion holds for $\mathcal{A}$ that for any constant α outputs $(1\pm\alpha)T_1(\tau)\pm\tau o(\log\frac{1/\delta}{\log m})$ with probability at least $1-\delta$.*

For small enough δ (for example, $\delta<m^{\log m}$), the theorem establishes that when an arbitrarily small multiplicative approximation factor is allowed, an additive polylogarithmic error is unavoidable for the problem of estimating distinct count. In particular, our lower bound matches the logarithmic dependence of the additive error on the probability of failure of our fully streaming (i.e. fully dynamic sublinear space) algorithm for distinct count estimation. The non-private sketch based algorithm for distinct count, gives an $(1+\alpha)$ ([1]) multiplicative approximation. Hence, the theorem also implies that a pan-private algorithm for the distinct count problem necessarily incurs error larger than a small space streaming algorithm for the problem by an additive factor. This is the first known separation between the

two models, namely between pan-private algorithms (with unbounded space) and polylogarithmic space streaming algorithms.

5. DISCUSSION

We focus not only on privacy of data analysis, but also security, formulated as pan-privacy in [7]. Informally, pan-private algorithms guarantee differential privacy of data analyses even when the internal memory of the algorithm may be compromised by an unannounced intrusion. We present the first pan-private algorithms on fully dynamic data for various useful statistics (distinct count, cropped sum, and heavy hitter count), and also present matching and almost matching lower bounds for these problems—the first such lower bounds explicitly for pan-privacy.

Privacy with security is an important issue, and pan-privacy [7] is an effective and interesting formulation of this problem. A number of extensions are of interest.

Other Security Models. In the bulk of the paper, we focus on security against a single unannounced intrusion. A natural extension is to protect against multiple intrusions. If the occurence of an intrusion is announces before or immediately after the intrusion, such as in applications where they are legally mandated or are detected by the system, then our results will still hold, with the simple fix to randomize anew after each intrusion. If the intrusions are unannounced, there are extreme cases when differential privacy cannot be ensured even with partly dynamic data [7]. We leave it open to formulate a realistic model of multiple unannounced intrusions and investigate tradeoffs between privacy and accuracy guarantees.

In a dynamic data scenario, it is often desirable to *continuously monitor* some set of statistics in order to detect trends in the data in a timely manner [4]. Our results can also be used to provide *continual event-level pan-privacy* [8, 4]—i.e., to provide the ability to monitor the statistics we have considered while ensuring privacy and security. *Event-level pan-privacy* can be defined analogously as in Definition 5 by considering event-level neighbors instead. Two sequences S and S' are said to be event-level neighbors if some "event" (i_k, d_k) in S is replaced by some other event (j, d_k), where $j \neq i_k$ in S'. While the notion of user-level privacy offers protection to a user, event-level privacy seeks to protect an "event,"—i.e., a particular update. Continual event-level pan privacy addresses the problem of providing continual outputs over dynamic data (over time $1 \leq t \leq T$), that are event-level pan-private with respect to one intrusion. As further evidence of the utility of linear sketches (and linear measurements of data, in general), we notice that along with L_p sketches, our noise adding technique of Section 3.1 can easily be extended to provide a continual event-level pan-private data structure for computing the number of distinct elements in a dynamic stream by a simple extension of the results in [8]. They propose a counter that within a bounded time period of T provides an accurate estimate of the number of ones in a binary stream, with an additive error term scaled by $O(\log(T)^{2.5})$. A key ingredient in their construction is the linearity of the binary count operation; since operations on sketches are also linear, essentially the same construction (replaced by linear sketch updates) works for our case. The same observation can also be made for our sketches for cropped sum and k-heavy hitters.

Other Data Models. We studied the fully dynamic data where items may be inserted or deleted. In such applications, at all times, for all i, $a_i \geq 0$ since one does not delete an item or copy that was not inserted. Still, there are applications with for example, distributed data, which may be modeled by dynamic data where some a_i's may be negative. Our algorithm for distinct count from Section 3.1 still works and provides the same guarantees, but we need new algorithms for estimating cropped sum and heavy hitter count in such a data model.

Other Queries. We studied basic statistical queries in this paper. Many richer queries are of interest, including estimating the entropy of dynamic data, join size estimation for dynamic relations, graph quantities on dynamic graphs, rank and compressibility of dynamic matrices and so on.

Other Lower Bound Techniques. In the cases above, we may also need new lower bound techniques beyond the one based on noisy decoding that we introduced in this paper. Developing a collection of lower bounds for problems in the one-way two-party differential privacy model will be useful for showing lower bounds for pan-private algorithms.

We believe that there is a rich theory of pan-private algorithms that needs to be developed, inspired by recent work on differential privacy and streaming algorithms, but already quite distinct as we know from [7] and this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for many insightful comments and Kobbi Nissim and Graham Cormode for helpful discussions.

Darakhshan Mir was supported by NSF Awards No. CCF-0728937 and CCF-1018445 and U.S. DHS Award No. 2009-ST-061-CCI002, S. Muthukrishnan was supported by NSF Awards No. DMS-0354690, IIS-0414852 and CCF-916782, and U.S. DHS Award No. 2009-ST-061-CCI002. Aleksandar Nikolov was supported by NSF Award No. CCF-1018445. Rebecca N. Wright was supported by NSF Award No. CCF-1018445.

6. REFERENCES

[1] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *TKDE*, 2003.

[2] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 2005.

[3] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, 2003.

[4] C. Dwork. Differential privacy in new settings. In *SODA*, 2010.

[5] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[6] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of lp decoding. In *STOC*, 2007.

[7] C. Dwork, M. Naor, T. Pitassi, G. Rothblum, and S. Yekhanin. Pan-Private streaming algorithms. In *ICS*, 2010.

[8] C. Dwork, T. Pitassi, M. Naor, and G. Rothblum. Differential privacy under continual observation. In *STOC*, 2010.

[9] A. Gupta, K. Ligett, F. McSherry, A. Roth, and

K. Talwar. Differentially private combinatorial optimization. In *SODA*, 2010.

[10] P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. In *Journal of ACM*, 2006.

[11] D. Kane, J. Nelson, and D. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, 2010.

[12] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan. Limits of two-party differential privacy. In *FOCS*, 2010.

[13] F. Mcsherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, 2007.

[14] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In *CRYPTO*, 2009.

[15] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, 2010. In progress, Chapter 1 online at academic2.american.edu/∼jpnolan.

[16] S. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association*, 1965.

APPENDIX

A. PROOFS FOR SECTION 3

A.1 Proofs for Subsection 3.2

Here we prove the following technical claim, used in the proof of Lemma 4.

CLAIM 2. *Let $a \geq \tau$ and let w be uniformly distributed in $[1, 2]$. Then $\mathbb{E}[wa \bmod 2\tau] \geq \tau/2$.*

PROOF. Let $a \bmod 2\tau = j$. Assume that $a = 2x\tau + y$, for x, y positive integers, $y \leq 2\tau - 1$. We consider two cases.

Case I. Assume that $y < 2\tau - j$. Then conditioned on the event $w \in [1, 1 + 2x\tau/a]$, wa is uniformly distributed in $[0, 2\tau)$; conditioned on the event $w \in [1 + 2x\tau/a, 2]$, wa is uniformly distributed in $[j, j + y]$. By the law of total expectation,

$$\mathbb{E}[wa \bmod 2\tau] = \frac{2x\tau}{a}\tau + \frac{y}{a}(j + \frac{1}{2}y)$$

If $x \geq 1$, $2x\tau/a > 1/2$; then $\mathbb{E}[wa \bmod 2\tau] > \tau/2$. If $x = 0$, then $y \geq \tau$, and $\mathbb{E}[wa \bmod 2\tau] \geq j + \tau/2 \geq \tau/2$.

Case II. Assume that $y \geq 2\tau - j$. Again, conditioned on $w \in [1, 1 + 2x\tau/a]$, wa is uniformly distributed in $[0, 2\tau)$. Conditioned on $w \in [1 + 2x\tau/a, 2]$, wa is uniformly distributed in $[j, 2\tau) \cup [0, y - 2\tau + j)$. By the law of total expectation,

$$\mathbb{E}[wa \bmod 2\tau] = \frac{2x\tau}{a}\tau + \frac{2\tau - j}{a}(j + \frac{2\tau - j}{2}) + \frac{y - 2\tau + j}{a}\frac{y - 2\tau + j}{2}.$$

Once again, if $x \geq 1$, $\mathbb{E}[wa \bmod 2\tau] > \tau/2$. If $x = 0$, then $y = a \geq \tau$. Let $(2\tau - j)/a = f$. Then we have,

$$\mathbb{E}[wa \bmod 2\tau] = f(j + \frac{fa}{2}) + (1 - f)\frac{(1 - f)a}{2} \geq \frac{a}{2} \geq \tau/2.$$

□

B. PROOFS FOR SECTION 4

B.1 Noisy Decoding

B.2 Heavy Hitters Lower Bound

Next, we present the details of the reduction that establishes a lower bound for the heavy hitters problem (Corollary 2). Once again, recall that $\mathrm{HH}(k)$ is the number of items i such that $a_i \geq F_1/k$.

COROLLARY 5. *Let $\mathcal{A}$ be an online algorithm that outputs an estimate $\tilde{\mathrm{HH}}(k) \in [\mathrm{HH}(k) - o(\sqrt{k}), \mathrm{HH}(k') + o(\sqrt{k})]$ for some $k' \leq F_1$ with probability at least $1 - O(m^{-2})$. Then $\mathcal{A}$ is not ϵ-pan private for any constant ϵ*

PROOF. Given a sequence $S_t = \{(i_1, d_1), (i_2, d_2), \ldots\}$ with state vector $\mathbf{a}$ and $i_j \in [k]$ for all j, we can construct a sequence $S'_t = \{(i'_1, d'_1), (i'_2, d'_2), \ldots\}$ with state vector $\mathbf{a}'$ and $i'_j \in \mathcal{U}$ in the following way. Select an arbitrary set $J \in \binom{\mathcal{U}}{k}$, a mapping $\phi : [k] \to J$, and a real number W. If $a_i > 0$, construct S'_t so that $a'_{\phi(i)} = W/k$; otherwise, let $a'_{\phi(i)} = 0$. For our lower bound it is enough to assume that $\mathbf{a}$ is binary and to show a contradiction with the definition of differential privacy on a neighbor of S_t which is also binary, i.e. has the same hamming weight but one 1 was "moved" to a different coordinate. The reason this is sufficient for a lower bound is that for the proof of Corollary 1 it is sufficient to consider binary instances and binary neighbors. Observe then that for binary instances and binary neighbors with the same weight the following hold:

- two neighboring sequences on universe $[k]$ give rise to two neighboring sequences on universe $\mathcal{U}$; therefore, an algorithm that is ϵ-pan-private with respect to the transformed input is also ϵ-pan-private with respect to the original input;
- the number of k-heavy hitters in S'_t is equal to the distinct counts for S_t, as each item with nonzero state in S_t maps to an item with state WF_1/k in S'_t, and $F'_1 \leq WF_1$;
- since each item in S'_t is either a k-heavy hitter or has state 0, $HH(k) = HH(k')$ for any $k' \leq F'_1$.

As a consequence, an algorithm that is ε-pan private and outputs $\tilde{\mathrm{HH}}(k) \in [\mathrm{HH}(k) - o(\sqrt{k}) \ , \mathrm{HH}(k') + o(\sqrt{k})]$ with probability $1 - \delta$ can be used to approximate $D^{(t)}$ to within $o(\sqrt{k})$ additive error with probability $1 - \delta$ on sequences of updates with items drawn from $[k]$, while satisfying ϵ-pan-privacy. By Corollary 1, this is a contradiction. □