Non-interactive (t, n)-Incidence Counting from Differentially Private Indicator Vectors

Mohammad Alaggan, Mathieu Cunche, Marine Minier

Non-interactive (t, n)-Incidence Counting from Differentially Private Indicator Vectors*

Mohammad Alaggan[†1], Mathieu Cunche[‡1], and Marine Minier[§3]

[1]Univ Lyon, Inria, INSA Lyon, CITI, F-69621 Villeurbanne, France
[2]Univ Lorraine, LORIA, UMR 7503, F-54506 Vandoeuvre-lès-Nancy, France

March 8, 2017

Abstract

We present a novel non-interactive (t, n)-incidence count estimation for indicator vectors ensuring Differential Privacy [8, 1]. Given one or two differentially private indicator vectors, estimating the distinct count of elements in each [3] and their intersection cardinality (equivalently, their inner product [1]) have been studied in the literature, along with other extensions for estimating the cardinality set intersection in case the elements are hashed prior to insertion [2]. The core contribution behind all these studies was to address the problem of estimating the Hamming weight (the number of bits set to one) of a bit vector from its differentially private version, and in the case of inner product and set intersection, estimating the number of positions which are jointly set to one in both bit vectors.

We develop the most general case of estimating the number of positions which are set to one in exactly t out of n bit vectors (this quantity is denoted the (t, n)-incidence count), given access only to the differentially private version of those bit vectors. This means that if each bit vector belongs to a different owner, each can locally sanitize their bit vector prior to sharing it, hence the *non-interactive* nature of our algorithm.

Our main contribution is a novel algorithm that *simultaneously* estimates the (t, n)-incidence counts for all $t \in \{0, \ldots, n\}$. We provide upper and lower bounds to the estimation error.

Our lower bound is achieved by generalizing the limit of two-party differential privacy [11] into n-party differential privacy, which is a contribution of independent interest. In particular we prove a lower bound on the additive error that must be incurred by any n-wise inner product of n mutually differentially-private bit vectors.

Our results are very general and are not limited to differentially private bit vectors. They should apply to a large class of sanitization mechanism of bit vectors which depend on flipping the bits with a constant probability.

Some potential applications for our technique include physical mobility analytics [14], call-detail-record analysis [2], and similarity metrics computation [1].

1 Introduction

Consider a set of n bit vectors, each of size m. Let $\boldsymbol{a}$ be the vector with m components, in which $a_i \in \{0, \ldots, n\}$ is the sum of the bits in the i-th position in each of the n bit vectors. Then the (t, n)-incidence count is the number of positions i such that $a_i =$

*This work is supported by Cisco grant CG# 593780.
[†]mohammad.alaggan@inria.fr
[‡]mathieu.cunche@inria.fr
[§]marine.minier@loria.fr

t. Let the *incidence vector* $\mathbf{\Phi}$ be the vector of $n+1$ components in which Φ_t is the (t,n)-incidence count, for $t \in \{0,\ldots,n\}$. It should be noted that $\sum_t \Phi_t = m$, since all m buckets must be accounted for. $\mathbf{\Phi}$ can also be viewed as the *frequency* of elements or *histogram* of $\boldsymbol{a}$.

Now consider the vector $\tilde{\boldsymbol{a}}$ resulting from the sanitized version of those vectors, if they have been sanitized by probabilistically flipping each bit b independently with probability $0 < p < 1/2$:

$$b \mapsto b \oplus \text{Bernoulli}(p) \quad . \tag{1}$$

Then each component of $\tilde{\boldsymbol{a}}$ will be a random variable[1] defined as: $\tilde{a}_i = \text{Binomial}(a_i, 1-p) + \text{Binomial}(n - a_i, p)$. This is because (1) can be rewritten as: $b \mapsto \text{Bernoulli}(p)$ if $b = 0$ and $b \mapsto \text{Bernoulli}(1-p)$ if $b = 1$, and there are a_i bits whose value is one, and $n - a_i$ bits whose value is zero, and the sum of identical Bernoulli random variables is a Binomial random variable.

Finally, define $\mathbf{\Psi}$ to be the histogram of $\tilde{\boldsymbol{a}}$, similarly to $\mathbf{\Phi}$. To understand $\mathbf{\Psi}$ consider entry i of $\mathbf{\Phi}$, which is the number Φ_i of buckets containing i ones out of n. Take one such bucket; there is a probability that the i ones in that bucket be turned into any of $j = 0, 1, \ldots, n$. The vector describing such probabilistic transformation follows a multinomial distribution. This is visually illustrated in Figure 1, by virtue of an example on two bit vectors.

The main contribution of this paper is a novel algorithm to estimate the true incidence vector $\mathbf{\Phi}$ given the sanitized incidence vector $\mathbf{\Psi}$ and p.

This model captures perturbed Linear Counting Sketches (similar to [9] which is *not* a flipping model), and BLIP [1, 2] (a differentially-private Bloom filter).

In [1], Alaggan, Gambs, and Kermarrec showed that when the flipping probability satisfies $(1-p)/p = \exp(\epsilon)$ for $\epsilon > 0$, then this flipping mechanism will satisfy ϵ-differential privacy (*cf.* Definition 2.1). This means that the underlying bit vectors will be protected with non-interactive randomized-response ϵ-differential privacy in which $\epsilon = \ln((1-p)/p)$.

[1]Which is a special case of the Poisson binomial distribution, where there are only two distinct means for the underlying Bernoulli distributions. The mean and variance of $\tilde{a}_i$ are defined as the sums of the means and variances of the two underlying binomial distribution, because they are independent.

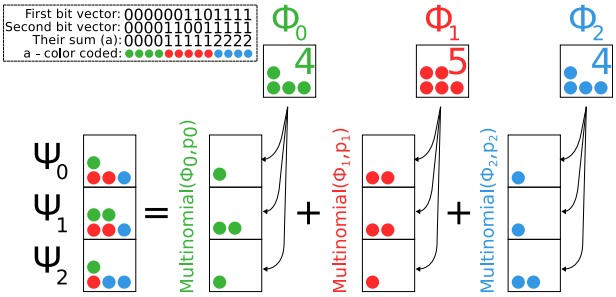

Figure 1: An example to our model. There are two bit vectors and $\boldsymbol{a}$ represents the number of bits set to one in each adjacent position, while $\mathbf{\Phi}$ represents the histogram of $\boldsymbol{a}$. For example Φ_1 is the number of entries in $\boldsymbol{a}$ which are equal to 1 (shown in red). The rest of the diagram shows what happens to entries of $\boldsymbol{a}$ if the bit vectors are sanitized by randomly and independently flipping each of their bits with probably $p < 1/2$, and how the histogram consequently changes to the random variable $\mathbf{\Psi}$. In particular, that Φ_t is probabilistically transformed into a vector-valued Multinomial random variable.

1.1 Summary of Our Results

We find that our results are best presented in terms of another parameter $0 < \eta < 1$ instead of p. Let η be such that the flipping probably $p = 1/2 - \eta/2$. We will not reference p again in this paper.

In our presentation and through the entirety of this paper, both η (which we will reference as "the privacy parameter") and ϵ (which will be referenced as "the differential privacy parameter") are completely interchangeable; since one fully determines the other through the relation

$$\epsilon = \ln\left(\frac{1+\eta}{1-\eta}\right) \quad . \tag{2}$$

However, our theoretical results will be presented in terms of η for the sake of simplicity of presentation. On the other hand, the experimental evaluation will be presented in terms of ϵ; since ϵ is the differential

privacy parameter and it will provide more intuition to the reader about the privacy guarantees provided for the reported utility (additive error). In a practical application, one may decide the value of ϵ first to suit their privacy and utility needs and then compute the resulting η value that is then given to our algorithm. A discussion on how to choose ϵ is provided in [1], which may also aid the reader with having an intuition to the value of ϵ used in our experimental evaluation and why we decided to use those values.

(t, n)-Incidence Estimation. In the following we describe upper U and lower bounds L to the additive error. That is, $\max_i |\Psi_i - \Phi'_i| \leqslant U$ and $L \leqslant \min_i |\Psi_i - \Phi'_i|$, in which $\mathbf{\Phi}'$ is the estimate output by our algorithm (for the upper bound) or the estimate output by *any* algorithm (for the lower bound). L and U may depend on m, the size of the bit vectors, n, the number of bit vectors, η, the privacy parameter, and β, the probability that the bounds fails for at least one i.

Upper Bound. Theorem 4.4 states that there exist an algorithm that is ϵ-differentially private that, with probability at least $1 - \beta$, *simultaneously* estimates Φ_i for all i with additive error no more than

$$\sqrt{2m} \cdot O(\eta^{-n}) \cdot \sqrt{\ln\left(\frac{1}{\beta}\right)} \cdot \sqrt{\ln(n+1)} \ .$$

Note that this is not a trivial bound since it is a bound on estimating $n > 2$ *simultaneous* n-wise inner products. Additionally, in relation to the literature on communication complexity [11], we consider the *number-in-hand* rather than *number-on-forehead* communication model, which is more strict.

The $O(\eta^{-n})$ factor is formally proven, but in practice the actual value is much smaller, as explained in Section 6.1. A discussion of the practicality of this bound given the exponential dependence on n is given in Section 4.2.

Lower Bound. In Theorem 5.10 we generalize the results of [11] to multiple bit vectors and obtain the lower bound that for all i, any ϵ-differentially private algorithm for approximating Φ_i must incur additive error

$$\Omega\left(\frac{\sqrt{m}}{\log_2(m)} \cdot \beta \cdot \left(\frac{1-\eta}{1+\eta}\right)\right) \ ,$$

with probability at least $1 - \beta$ over randomness of the bit vectors *and* the randomness of the perturbation. It is worth noting that the upper bounds hold for all values of ϵ, but the lower bound is only shown for $\epsilon < 1$. Also notice that this lower bound does not depend on n.

The result also presents a lower bound on the additive error that must be incurred by any such algorithm for estimating n-wise inner product. The relation between the n-wise inner product and (t, n)-incidence is made explicit in the proof of Theorem 5.10.

In Section 2, we start by presenting differential privacy, after which we discuss the related work in Section 3, then in Section 4 we describe the the (t, n)-incidence counting algorithm and prove its upper bounds. The lower bound on n-wise inner product is then presented in Section 5. Finally, we finish by validating our algorithm and bounds on a real dataset in Section 6 before concluding in Section 7.

2 Background

2.1 Differential Privacy

The notion of privacy we are interested is *Differential Privacy* [8]. It is considered a strong definition of privacy since it is a condition on the sanitization mechanism that holds equally well for any instance of the data to be protected. Furthermore, it makes no assumptions about the adversary. That is, the adversary may be computationally unbounded and has access to arbitrary auxiliary information. To achieve this, any differentially private mechanism must be randomized. In fact, the definition itself is a statement about a probabilistic event where the probability is taken only over the coin tosses of such mechanism. The intuition behind differential privacy is that the distribution of the output of the mechanism should not change much (as quantified by a parameter ϵ) when an individual is added or removed from the input. Therefore, the output does not reveal much information about that individual nor even about the very fact whether they were in the input or not.

Definition 2.1 (ϵ-Differential Privacy [8])**.** A randomized function $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ is ϵ-differentially private, if for all vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t} \in \{0,1\}^n$:

$$\Pr[\mathcal{F}(\boldsymbol{x}) = \boldsymbol{t}] \leqslant \exp(\epsilon \cdot \|\boldsymbol{x} - \boldsymbol{y}\|_H) \Pr[\mathcal{F}(\boldsymbol{y}) = \boldsymbol{t}] \ , \quad (3)$$

in which $\|\boldsymbol{x} - \boldsymbol{y}\|_H$ is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, that is, the number of positions at which they differ. The probability is taken over all the coin tosses of $\mathcal{F}$.

The parameter ϵ is typically small and is usually thought of as being less than one. The smaller its value the less information is revealed and more private the mechanism is. However, it also means less estimation accuracy and higher estimation error. Therefore the choice of a value to use for ϵ is a trade-off between privacy and utility. To the best of our knowledge there is no consensus on a method to decide what this value should be. In some of the literature relevant to differentially private bit vectors [1], an attack-based approach was adopted as a way to choose the largest ϵ (and thus highest utility) possible such that the attacks fail. Given the attacks from [1] we can choose ϵ up to three without great risk.

3 Related Work

Incidence counting has been studied in the streaming literature as well as in the privacy-preserving algorithms literature under the names: t-incidence counting [9], occurrence frequency estimation [6, 7], or distinct counting [13]. We use these terms interchangeably to mean an accurate estimate of the distinct count, not an upper or lower bound on it.

There are several algorithms in the streaming literature that estimate the occurrence frequency of different items or find the most frequent items [9, 7, 6]. The problem of occurrence frequency estimation is related to that of incidence counting in the following manner: they are basically the same thing except the former reports normalized relative values. Our algorithm, instead, reports all the occurrence frequencies, not just the most frequent ones. We face the additional challenging that we are given a *privacy-preserving* version of the input instead of its raw value, but since in our application (indicator vectors) usually $m \gg n$, we use linear space in n, rather than logarithmic space like most streaming algorithms.

The closest to our work is the t-incidence count estimator of Dwork, Naor, Pitassi, Rothblum, and Yekhanin [9]. Their differentially private algorithm takes the private stream elements a_i before sanitation and sanitizes them. To the contrary, our algorithm takes the elements a_i *after* they have already been sanitized. An example inspired by [2] is that of call detail records stored by cell towers. Each cell tower stores the set of caller/callee IDs making calls for every time slot (an hour or a day for instance), as an indicator vector. After the time slot ends, the resulting indicator vector is submitted to a central facility for further analysis that involves multiple cell towers. Our work allows this central facility to be untrusted, which is not supported by [9].

In subsequent work, Mir, Muthukrishnan, Nikolov, and Wright [13] propose a p-stable distribution-based sketching technique for differentially private distinct count. Their approach also supports deletions (*i.e.* a_i may be negative), which we do not support. However, to reduce the noise, they employ the exponential mechanism [12], which is known to be computationally inefficient. Their algorithm also faces the same limitations than the ones of [9].

4 Upper Bounds

The algorithm we present and the upper bounds thereof depend on the probabilistic linear mapping A' between the observed random variable $\boldsymbol{\Psi}$ and the unknown $\boldsymbol{\Phi}$ which we want to estimate. In fact, A' and its expected value $A = \mathbb{E}[A']$ are the primary objects of analysis of this section. Therefore we begin by characterizing them.

Recall that $\boldsymbol{\Psi}$ is the histogram of $\tilde{\boldsymbol{a}}$ (*cf.* Figure 1) and that the distribution of $\tilde{a}_i$ is $Z(n, p, a_i)$ in which

$$Z(n, p, j) = \text{Binomial}(j, 1-p) + \text{Binomial}(n-j, p) \ , \quad (4)$$

and $p < 1/2$. The probability mass function of $Z(n, p, j)$ is presented in Appendix A.

In what follows we drop the n and p parameters

of $Z(n,p,j)$ since they are always implied from context. We will also denote $\boldsymbol{P}(Z(j))$ be the probability vector characterizing $Z(j)$: $(\Pr[Z(j)=0], \Pr[Z(j)=1], \ldots, \Pr[Z(j)=n])$. Finally, $\boldsymbol{e}_i$ will denote the ith basis vector. That is, the vector whose components are zero except the ith component which is set to one.

The following proposition defines the probabilistic linear mapping A' between $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$.

Proposition 4.1. *Let A' be a matrix random variable whose jth column independently follows the multinomial distribution* $\mathrm{Multinomial}(\Phi_j, \boldsymbol{P}(Z(j)))$. *Then the histogram of $\tilde{\boldsymbol{a}}$ is the sum of the columns of A': $\boldsymbol{\Psi} = A'\mathbf{1}$ in which $\mathbf{1} = (1,1,\ldots,1)$, and thus $\boldsymbol{\Psi} = \sum_j \mathrm{Multinomial}(\Phi_j, \boldsymbol{P}(Z(j)))$.*

Proof. Since $\boldsymbol{\Psi}$ is the histogram of $\tilde{a}$, it is thus can be written as $\boldsymbol{\Psi} = \sum_i \boldsymbol{e}_{\tilde{a}_i} = \sum_j \sum_{i\in\{k|a_k=j\}} \boldsymbol{e}_{\tilde{a}_i}$. Then since 1) the sum of k independent and identical copies of $\mathrm{Multinomial}(1,\boldsymbol{p})$, for any $\boldsymbol{p}$, has distribution $\mathrm{Multinomial}(k,\boldsymbol{p})$, and 2) $|\{k \mid a_k = j\}| = \Phi_j$ by definition, and 3) $\boldsymbol{e}_{\tilde{a}_i}$ is a random variable whose distribution is $\mathrm{Multinomial}(1, \boldsymbol{P}(Z(a_i)))$, then the result follows; because $\sum_{i\in\{k|a_k=j\}} \boldsymbol{e}_{\tilde{a}_i}$ has distribution $\mathrm{Multinomial}(\Phi_j, \boldsymbol{P}(Z(j)))$. □

The following corollary defines the matrix A, which is the the expected value of A'.

Corollary 4.2. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be the matrix whose jth column is $\boldsymbol{P}(Z(j))$. Then $\mathbb{E}\boldsymbol{\Psi} = A\boldsymbol{\Phi}$.*

Proof. Follows from the mean of the multinomial distribution: $\mathbb{E}[\mathrm{Multinomial}(\Phi_j, \boldsymbol{P}(Z(j)))] = \Phi_j \boldsymbol{P}(Z(j))$. □

It is also worth noting that due to the symmetry in (4), we have that

$$A_{ij} = \Pr[Z(j)=i] = \Pr[Z_{n-j} = n-i] = A_{n-i,n-j} \ . \tag{5}$$

For the rest of the paper we will be working exclusively with ℓ_1-normalized versions of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. That is, the normalized versions will sum to one. Since they both originally sum to m, dividing both of them by m will yield a vector that sums to one. The following corollary extends the results of this section to the case when $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are normalized to sum to one. In the following, $\mathrm{diag}(\boldsymbol{x})$ is the diagonal matrix whose off-diagonal entries are zero and whose diagonal equals $\boldsymbol{x}$.

Corollary 4.3. $\big(\boldsymbol{\Psi} = A'\mathbf{1} = A'\mathrm{diag}(1/\boldsymbol{\Phi})\boldsymbol{\Phi}\big) \implies \big(\boldsymbol{\Psi}/m = A'\mathrm{diag}(1/\boldsymbol{\Phi})(\boldsymbol{\Phi}/m)\big)$, *and consequently* $\big(\mathbb{E}\boldsymbol{\Psi} = A\boldsymbol{\Phi}\big) \implies \big(\mathbb{E}\boldsymbol{\Psi}/m = A\boldsymbol{\Phi}/m\big)$.

4.1 The Estimation Algorithm

Let $\hat{\boldsymbol{\Phi}} = \boldsymbol{\Phi}/m$ and $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}/m$ be the ℓ_1-normalized versions of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$.

Intuition. The first step in our algorithm is to establish a *confidence interval*[2] of diameter $f(\delta)/2$ around the perturbed incidence vector $\hat{\boldsymbol{\Psi}}$, such that, with probability at least $1-\beta$, its *expected value* $\boldsymbol{x} \stackrel{\text{def}}{=} A\hat{\boldsymbol{\Phi}}$ is within this interval. Note that this confidence interval depends only of public parameters such as η, m, and n, but not on the specific $\hat{\boldsymbol{\Psi}}$ vector. Afterwards, we use linear programming to find a valid incidence vector within this interval that could be the preimage of $\hat{\boldsymbol{\Psi}}$, yielding the vector $\boldsymbol{y} \stackrel{\text{def}}{=} A\hat{\boldsymbol{\Phi}}'$. Since $\boldsymbol{x}$ is within this interval with probability at least $1-\beta$ then the linear program has a solution with probability at least $1-\beta$. Consequently, $\boldsymbol{x}$ and $\boldsymbol{y}$ are within ℓ_∞ distance $f(\delta)$ from each other, with probability at least $1-\beta$. It remains to establish, given this fact, the ℓ_∞ distance between the true $\hat{\boldsymbol{\Phi}}$ and the estimated $\hat{\boldsymbol{\Psi}}'$, which is an upper bound to the additive error of the estimate. The details are provided later in Section 4.2.

Our estimation algorithm will take $\hat{\boldsymbol{\Psi}}$ and A as input and will produce an estimate $\hat{\boldsymbol{\Phi}}'$ to $\hat{\boldsymbol{\Phi}}$. It will basically use linear programming to guarantee that

$$\left\|\hat{\boldsymbol{\Psi}} - A\hat{\boldsymbol{\Phi}}'\right\|_\infty \leqslant f(\delta)/2 \ . \tag{6}$$

The notation $\|\boldsymbol{x}\|_\infty$ is the *max norm* or ℓ_∞ *norm* and is equal to $\max_i |x_i|$. Suitable constraints to guarantee that $\hat{\boldsymbol{\Phi}}'$ is a valid frequency vector (that its components are nonnegative and sum to 1) are employed. These constraints cannot be enforced in case the naïve

[2]The word "interval" is inappropriate here since the random variable is a vector. Technically, "ℓ_∞-ball" would be more appropriate.

unbiased estimator $A^{-1}\hat{\mathbf{\Psi}}$ is used (it would be unbiased because of Corollary 4.3). This linear program is shown in Algorithm 1.

The objective function of the linear program. The set of constraints of the linear program specify a finite convex polytope with the guarantee that, with probability $1-\beta$, the polytope contain the true solution, and that all points in this polytope are within a bounded distance from the true solution. We are then simply using the linear program as a linear constraint solver that computes an arbitrary point within this polytope. In particular, we are *not* using the linear program as an optimization mechanism. Hence, the reader should not be confused by observing that the objective function which the linear program would normally minimize is simply a constant (zero) which is independent of the LP solution.

From a practical point of view, however, it matters which point inside the polytope gets chosen. In particular, the polytope represents the probabilistically-bounded preimage of the perturbed observation. It is unlikely that the true solution lies exactly on or close to the boundary of such polytope, and is rather expected, probabilistically speaking, to exist closer to the centroid of the polytope that to its boundary. We have experimentally validated that, for low n, the centroid of the polytope is at least twice as close to the true solution than the output of the linear program (using the interior point method) which is reported in Section 6. Unfortunately, it is computationally intensive to compute the centroid for high n and thus we were not able to experimentally validate this claim in these cases. This also means that the centroid method is not practical enough. Instead, we recomment the use of the *interior point* algorithm for linear programming which is more likely to report a point from the interior of the polytope that the *simplex* algorithm which always reports points exactly on the boundary. We have also experimentally validated that the former always produces better estimates than the latter, even though both of them do satisfy our upper bound (which is independent of the LP algorithm used). An alternative theoretical analysis which provides an formal error bound for the centroid method could be the topic of future work. In Section 6 we only report results using the interior method algorithm.

Parameter Selection. The remainder of this section and our main result will proceed to show sufficient conditions that, with high probability, make (6) imply $\left\|\hat{\mathbf{\Phi}}' - \hat{\mathbf{\Phi}}\right\|_\infty \leqslant \delta$ for user-specified accuracy requirement δ. These conditions will dictate that either one of δ, ϵ, or m depend on the other two. Typically the user will choose the two that matter to him most and let our upper bounds decide the third. For example, if the user wants m to be small for efficiency and δ also be small for accuracy, then she will have to settle for a probably large value of ϵ which sacrifices privacy. Sometimes the resulting combination may be unfeasible or uninteresting. For instance, maybe m is required to be too large to fit in memory or secondary storage. Or perhaps δ will be required to be greater than one, which means that the result will be completely useless. In these cases the user will have to either refine his choice of parameters or consider whether his task is privately computable in the randomized response model. It may also be the case that a tighter analysis may solve this problem, since some parts of our analysis are somewhat loose bounds and there may be room for improvement. The probability $1-\beta$ that the bound holds can be part of the trade-off as well.

Algorithm 1 Linear Program

Given $\hat{\mathbf{\Psi}}$ and η, solve the following linear program for the variable $\hat{\mathbf{\Phi}}'$, in which $f(\delta)/2 = \|A^{-1}\|_\infty \sqrt{2\ln(1/\beta)\ln(n+1)/m}$.

$$\begin{array}{ll} \text{minimize} & 0\ , \\ \text{s.t. } \forall i & -f(\delta)/2 \leqslant \sum_j \hat{\Psi}_j - A_{ij}\hat{\Phi}'_j \leqslant f(\delta)/2\ , \\ \text{and } \forall i & \hat{\Phi}'_i \geqslant 0\ , \\ \text{and} & \sum_i \hat{\Phi}'_i = 1\ . \end{array}$$

Then output $\mathbf{\Phi}' = m\hat{\mathbf{\Phi}}'$ as the estimate of $\mathbf{\Phi}$.

4.2 Upper Bounding the Additive Error

As explained earlier, the first step is to find an ℓ_∞ ball of confidence around the the expected value of the perturbed incidence vector. This is is provided

by Theorem B.1 through a series of approximations and convergences between probability distributions, which are detailed in two lemmas, all in the appendix. The high level flow and the end result is shown in the following theorem and is meant only to be indicative. For details or exact definitions of particular symbols, kindly refer to Appendix B.

Theorem 4.4. *The component-wise additive error between the estimated incidence vector output by Algorithm 1 and the true incidence vector is* $\left\|\hat{\mathbf{\Phi}} - \mathbf{\Phi}'\right\|_\infty \leqslant \sqrt{2m} \cdot O(\eta^{-n}) \cdot \sqrt{\ln\left(\frac{1}{\beta}\right)} \cdot \sqrt{\ln(n+1)}$.

Proof. Assuming the matrix A is nonsingular, the matrix norm (of A^{-1}) induced by the max norm is, by definition: $\left\|A^{-1}\right\|_\infty = \sup_{\boldsymbol{x}\neq\mathbf{0}} \left\{\left\|A^{-1}\boldsymbol{x}\right\|_\infty / \left\|\boldsymbol{x}\right\|_\infty\right\}$, and since A is nonsingular we can substitute $\boldsymbol{x} = A\boldsymbol{y}$ in the quantifier: $\left\|A^{-1}\right\|_\infty = \sup_{\boldsymbol{y}\neq\mathbf{0}} \left\{\left\|A^{-1}A\boldsymbol{y}\right\|_\infty / \left\|A\boldsymbol{y}\right\|_\infty\right\}$ without loss of generality, yeilding $\sup_{\boldsymbol{y}\neq\mathbf{0}} \left\{\left\|\boldsymbol{y}\right\|_\infty / \left\|A\boldsymbol{y}\right\|_\infty\right\}$. Thus for all $\boldsymbol{y} \neq 0$, $\left\|A^{-1}\right\|_\infty \geqslant \frac{\|\boldsymbol{y}\|_\infty}{\|A\boldsymbol{y}\|_\infty}$. If we multiply both sides by $\|A\boldsymbol{y}\|_\infty / \|A^{-1}\|_\infty$ (which is positive), we get: $\left\|A^{-1}\right\|_\infty^{-1} \|\boldsymbol{y}\|_\infty \leqslant \|A\boldsymbol{y}\|_\infty$. In the following, we let $\boldsymbol{y} = \hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}'$. The rest of the proof begins by upper bounding the following expression using the preceding derivation:

$$
\begin{aligned}
&\left\|A^{-1}\right\|_\infty^{-1} \left\|\hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}'\right\|_\infty \leqslant \left\|A(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}')\right\|_\infty \\
&= \left\|A\hat{\mathbf{\Phi}} - A\hat{\mathbf{\Phi}}'\right\|_\infty = \left\|A\hat{\mathbf{\Phi}} + \left(\hat{\mathbf{\Psi}} - \hat{\mathbf{\Psi}}\right) - A\hat{\mathbf{\Phi}}'\right\|_\infty \\
&\leqslant \left\|A\hat{\mathbf{\Phi}} - \hat{\mathbf{\Psi}}\right\|_\infty + \left\|A\hat{\mathbf{\Phi}}' - \hat{\mathbf{\Psi}}\right\|_\infty \\
&\leqslant 2\left\|A\hat{\mathbf{\Phi}} - \hat{\mathbf{\Psi}}\right\|_\infty \text{ (LP constraint)} \\
&\leqslant \frac{2}{m} \mathrm{CDF}^{-1}_{G(a\mathcal{R}+\mathcal{M},b\mathcal{R})}(1-\beta) \text{ (By Lemma B.3;}\Phi_j \uparrow, n \uparrow) \\
&= \frac{2}{m}\left(\mathcal{M} + \mathcal{R}\beta''\right) \\
&\to \sqrt{\frac{2}{m}}\left(E_2 + (E_3 - E_1)\beta''\right) \text{ (By Lemma B.2;}\eta \downarrow) \\
&\to \sqrt{2\ln(1/\beta)\ln(n+1)/m} \text{ (By Theorem B.1;}n \uparrow)
\end{aligned}
$$

in which G is the Gumbel distribution, $\beta'' = a - b\ln(-\ln(1-\beta))$, $\mathcal{R}, \mathcal{M}$ which depend n, η and a, b which are absolute constants, are all defined in Lemma B.3 and subsequently approximated in Lemma B.2. It remains to show that $\|A^{-1}\|_\infty = O(\eta^{-n})$, which holds since η^{-n} is the largest eigenvalue of A^{-1}. The increase of Φ_j may either be justified or quantified in probability by Lemma C.1. □

Practicality of the bound. The factor $O(\eta^{-n})$ grows exponentially with n since $\eta < 1$. Therefore, if the bound is used in this form it may be useful for parameter selection only for very small n. In practice, however, the $O(\eta^{-n})$ factor is a over-estimation and its effective value is asymptotically sub-exponential. We discuss this issue and propose a practical solution in Section 6.1.

5 Lower Bounds

In this section we generalize the results of [11] to multiple bit strings and obtain the lower bound on approximating Φ_i. In the rest of this section we use $\lg(x)$ to denote the logarithm to base 2, and we let $\mu_0 = 1/2 - \eta/2$ and $\mu_1 = 1/2 + \eta/2$.

Definition 5.1. ***(Strongly α-unpredictable bit source)*** [11, Definition 3.2] For $\alpha \in [0,1]$, a random variable $X = (X_1, \ldots, X_m)$ taking values in $\{0,1\}^m$ is a *strongly α-unpredictable bit source* if for every $i \in \{1, \ldots, m\}$, we have

$$\alpha \leqslant \frac{\Pr[X_i=0 | X_1=x_1,\ldots,X_{i-1}=x_{i-1},X_{i+1}=x_{i+1},\ldots,X_m=x_m]}{\Pr[X_i=1 | X_1=x_1,\ldots,X_{i-1}=x_{i-1},X_{i+1}=x_{i+1},\ldots,X_m=x_m]} \leqslant 1/\alpha \ ,$$

for every $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}^{m-1}$.

Definition 5.2 (**β-closeness**)**.** Two random variables X and Y are β-close if the *statistical distance* between their distributions is at most β: $\sum_v 2^{-1} \left|\Pr[X=v] - \Pr[Y=v]\right| \leqslant \beta$, where the sum is over the set $\mathrm{supp}(X) \cup \mathrm{supp}(Y)$.

Definition 5.3 (**Min-entropy**)**.** Min-entropy of a random variable X is $H_\infty(X) = \inf_{x \in \mathrm{Supp}(X)} \lg\left(\frac{1}{\Pr[X=x]}\right)$.

Proposition 5.4. (Min-entropy of strongly α-unpredictable bit sources) *If X is a strongly α-unpredictable bit source, then X has min-entropy at least $m\lg(1+\alpha)$.*

Proof. Let

$$p = \Pr[X_i = 1 \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_m = x_m]$$

for any $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}^{m-1}$. Then we know that $\alpha \leqslant (1-p)/p \leqslant 1/\alpha$, and thus $p \leqslant 1/(1+\alpha)$. We can then verify that no string in the support of X has probability greater than $1/(1+\alpha)^m$. Thus X has min-entropy at least βm, in which $\beta = \lg(1+\alpha) \geqslant \alpha$. □

Lemma 5.5. (A uniformly random bit string conditioned on its sanitized version is an unpredictable bit source) *Let X be a uniform random variable on bit strings of length m, and let X' be a perturbed version of X, such that $X'_i = \text{Bernoulli}(\mu_0)$ if $X_i = 0$ and $\text{Bernoulli}(\mu_1)$ otherwise. Then X* conditioned on *X' is a strongly $\frac{1-\eta}{1+\eta}$-unpredictable bit source.*

Proof. Observe that since X is a uniformly random bit string then X_i and X_j are independent random variables for $i \neq j$. Since X'_i depends *only* on X_i for all i and not on any other X_j for $j \neq i$, then X'_i and X'_j are also independent random variables. Then using Bayes theorem and uniformity of X we can verify that for all $x' \in \{0,1\}^m$ and for all $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}^{m-1}$

$$\alpha \leqslant \frac{\Pr[X_i=0 | X_1=x_1,\ldots,X_{i-1}=x_{i-1},X_{i+1}=x_{i+1},\ldots,X_m=x_m,X'=x']}{\Pr[X_i=1 | X_1=x_1,\ldots,X_{i-1}=x_{i-1},X_{i+1}=x_{i+1},\ldots,X_m=x_m,X'=x']} \leqslant 1/\alpha \ ,$$

in which $\alpha = \mu_0/\mu_1 = (1-\eta)/(1+\eta)$. □

Lemma 5.6. *Let $S_1, \ldots, S_n$ be n uniform random variables on bit strings of length m, and for all $1 \leqslant i \leqslant n$ let S'_i be a perturbed version of S_i, such that for all $1 \leqslant j \leqslant m$, $S'_{ij} = \text{Bernoulli}(\mu_0)$ if $S_{ij} = 0$ and $\text{Bernoulli}(\mu_1)$ otherwise. Let Y be a vector such that $Y_j = \prod_i S_{ij}$ and Y' be a vector such that $Y'_j = \prod_i S'_{ij}$. Then Y* conditioned on *Y' is a strongly $\left(\frac{1-\eta}{1+\eta}\right)^n$-unpredictable bit source, and therefore has at least $m \lg\left(1 + \left(\frac{1-\eta}{1+\eta}\right)^n\right) \geqslant m\left(\frac{1-\eta}{1+\eta}\right)^n$ min-entropy.*

Proof. Follows the same line of the proof of Lemma 5.5. □

Theorem 5.7. *[11, Theorem 3.4] There is a universal constant c such that the following holds. Let X be an α-unpredictable bit source on $\{0,1\}^m$, let Y be a source on $\{0,1\}^m$ with min-entropy γm (independent from X), and let $Z = X \cdot Y \bmod k$ for some $k \in \mathbb{N}$, be the inner product of X and Y mod k. Then for every $\beta \in [0,1]$, the random variable (Y, Z) is β-close to (Y, U) where U is uniform on Z_k and independent of Y, provided that $m \geqslant c \cdot \frac{k^2}{\alpha\gamma} \cdot \lg\left(\frac{k}{\gamma}\right) \cdot \lg\left(\frac{k}{\beta}\right)$.*

Theorem 5.8. *[11, Theorem 3.9] Let $P(x,y)$ be a randomized protocol which takes as input two uniformly random bit vectors x, y of length m and outputs a real number. Let P be $\ln(\frac{1+\eta}{1-\eta})$-differentially private and let $\beta \geqslant 0$. Then with probability at least $1-\beta$ over the inputs $x, y \leftarrow \{0,1\}^m$ and the coin tosses of P, the output differs from $x^T y$ by at least $\Omega\left(\frac{\sqrt{m}}{\lg(m)} \cdot \beta \cdot \frac{1-\eta}{1+\eta}\right)$.*

Theorem 5.9. *Let $P(S_1, \ldots, S_n) = \sum_j \prod_i S_{ij}$ be the n-wise inner product of the vectors $S_1, \ldots, S_n$. If for all i, S_i is a uniform random variable on $\{0,1\}^m$, and S'_i is the perturbed version of S_i, such that $S'_{ij} = \text{Bernoulli}(\mu_0)$ if $S_{ij} = 0$ and $\text{Bernoulli}(\mu_1)$ otherwise, then with probability at least $1-\beta$ the output of any algorithm taking $S'_1, \ldots, S'_n$ as inputs will differ from $P(S_1, \ldots, S_n)$ by at least $\Omega\left(\frac{\sqrt{m}}{\lg(m)} \cdot \beta \cdot \frac{1-\eta}{1+\eta}\right)$.*

Proof. Without loss of generality take S_1 to be one vector and Y with $Y_j = \prod_{i=2}^n S_{ij}$ to be the other vector. Then we will use Theorem 5.8 to bound $S_1^T Y$. To use Theorem 5.8, we first highlight that S'_i is a $\ln(\frac{1+\eta}{1-\eta})$-differentially private version of S_i. Then since Theorem 5.8 depends on Theorem 5.7, we will show that S_1 and Y satisfies the condition of the latter theorem. Theorem 5.7 concerns inner product between two bit sources, one is an unpredictable bit source while the other has linear min-entropy. Lemma 5.5 shows that S_1 conditioned on its sanitized version S'_1 is an α-unpredictable bit source and Lemma 5.6 shows that Y has linear min-entropy (assuming n is constant in m). □

Theorem 5.10. *Let $S_1, \ldots, S_n$ be uniformly random binary strings of length m and let S'_i be a perturbed version of S_i, such that $S'_{ij} = \text{Bernoulli}(\mu_0)$ if $S_{ij} = 0$ and $\text{Bernoulli}(\mu_1)$ otherwise. Then let the vectors v, v' of length m be such that $v_i = \sum_j S_{ji}$ and $v'_i = \sum_j S'_{ji}$, and the vector $\mathbf{\Phi} = (\Phi_0, \ldots, \Phi_n)$ in which $\Phi_i = |\{j :$*

$v_j = i\}|$ *is the frequency of* i *in* v*, and similarly for* $\mathbf{\Phi}'$ *the frequency in* v'*. Then with probability at least* $1 - \beta$ *the output of an algorithm taking* S' *differs from* Φ_i *for all* i *by at least* $\Omega\left(\frac{\sqrt{m}}{\lg(m)} \cdot \beta \cdot \frac{1-\eta}{1+\eta}\right)$.

Proof. We will proceed by reducing n-wise inner product to frequency estimation. Since Theorem 5.9 forbids the former, then the theorem follows. The reduction is as follows. Let $P(j, A) = \prod_{i \in A} S_{ij}$ be the product of the bits in a particular position j across a subset A of the binary strings. Observe that $\sum_j P(j, [n])$, with $[n] = \{1, \ldots, n\}$, is the n-wise inner product of all the binary string. Similarly, let $\overline{P}(j, A) = \prod_{i \in A}(1 - S_{ij})$, be the product of the negated bits. Finally, denote $Q(A) = \sum_j P(j, A)\overline{P}(j, A^C)$, in which $A^C = [n] \setminus A$ is the complement of the set A. Now we claim that $\Phi_k = \sum_{A \subseteq [n], |A|=k} Q(A)$, in which the sum is over all subsets of $[n]$ of size k. This can be seen since for a set A of size k, $P(j, A)\overline{P}(j, A^C)$ is one only if $a_i = \sum_i S_{ij} = k$. Since there may be several sets A of the same size k, we can therefore conclude that the sum over all such sets $\sum_{A \subseteq [n], |A|=k} P(j, A)\overline{P}(j, A^C)$ is one *if and only if* $a_i = \sum_i S_{ij} = k$, and thus the sum (over all j) of the former quantity is the count (frequency) of the latter.

We will then show why the result follows first for Φ_0 and Φ_n then for $\Phi_1, \Phi_2, \ldots, \Phi_{n-1}$. According to this reduction, Φ_0 (resp. Φ_n) is equivalent to the n-wise inner product of $\{1 - S_1, 1 - S_2, \ldots, 1 - S_n\}$ (resp. $\{S_1, S_2, \ldots, S_n\}$) and thus if one was able to compute Φ_0 (resp. Φ_n) within error γ they would have also been able to compute those two n-wise inner products within error γ. Then we employ the lower bound on the n-wise inner product from Theorem 5.9 to lower bound γ for Φ_0 and Φ_n. For Φ_i for $i \notin \{0, n\}$, Φ_i is equivalent to the sum of $\binom{n}{i}$ n-wise inner products. In the case all but one of those n-wise inner products are zero, an estimate of Φ_i within error γ gives an estimate for a particular n-wise inner product within error γ as well, in which we can invoke Theorem 5.9 again to lower bound γ for Φ_i. □

6 Experimental Evaluation

We use the Sapienza dataset [4] to evaluate our method. It is a real-life dataset composed of wireless probe requests sent by mobile devices in various locations and settings in Rome, Italy. We only use the MAC address part of the dataset, as typical physical analytics systems do [14]. It covers a university campus and as city-wide national and international events. The data was collected for three months between February and May 2013, and contains around 11 million probes sent by 162305 different devices (different MAC addresses), therefore this is the size (m) of our indicator vectors. The released data is anonymized. The dataset contains 8 setting called POLITICS1, POLITICS2, VATICAN1, VATICAN2, UNIVERSITY, TRAINSTATION, THEMALL, and OTHERS. Each setting is composed of several files. Files are labeled according to the day of capture and files within the same setting occurring in the same day are numbered sequentially. In our experiments we set the parameter $n \in \{1, 2, \ldots, 21\}$, indicating the number of sets we want to experiment on. Then we pick n random files from all settings and proceed to estimate their t-incidence according to our algorithm. We add 1 to all incidence counts to reduce the computational overhead necessary to find a combination of files with non-zero incidence for all t for large n, so that the t-incidence for this random subset is nonzero for all t. This is unlikely to affect the results since the additive error will be much larger than 1 (about $O(\sqrt{m})$) anyway.

The additive error reported is the maximum additive error across all t. In real-life datasets, the additive error would be a problem only for low values of t (closer to the "intersection"), since the true value may be smaller than the additive error. However, for high t (closer to the "union"), high additive error is unlikely to be damaging to utility. This is a property of most real-world datasets since they are likely to follow a Zipf distribution. If this is the case it may be useful to consider employing the estimated union (or high t) to compute the intersection (or low t) via the inclusion-exclusion principle instead.

6.1 Calibrating to the Dataset

In our experiments we observe that the value of $\|A^{-1}\|_\infty$ may be too high for small ϵ, making it useless as an upper bound in this case. This is due to the definition of the induced norm, which takes the maximum over all vectors whose max norm is 1. This maximum is achieved for vectors in $\{-1,1\}^{n+1}$. However, in reality it is unlikely that the error vector will be this large and thus it may never actually reach this upper bound (as confirmed by the experiments). Instead, we consider the maximum over $\Gamma = \{-\gamma,\gamma\}^{n+1}$ for $\gamma < 1$ and use the fact that linearity implies $\max_{\boldsymbol{x}\in\Gamma} \|A^{-1}\boldsymbol{x}\|_\infty / \|\boldsymbol{x}\|_\infty = \gamma\|A^{-1}\|_\infty$. We empirically estimate γ by estimating, from the dataset, the multinomial distribution of $\boldsymbol{\Phi}$ for each n and each ϵ, then we sample vectors from this distribution and run our algorithm on them, and then compute γ from the the resulting error vector. We stress that this calibration process thus does not use any aspects of the dataset other than the distribution of $\boldsymbol{\Phi}$ and that γ depends only on n and ϵ and not on the actual incidence vector. Therefore, in real-life situation where there is no dataset prior to deployment to run this calibration on, it suffices to have prior knowledge (or expectation) to the distribution of the incidence vectors. For most applications it should follow a power-law distribution.

If Figure 2, all the lines represent the $1-\beta$ quantile. For instance, in the Sapienza line, the $1-\beta$ quantile (over 1000 runs) is shown. For the other line, the upper bound value was computed to hold with probability at least $1-\beta$. The value of β we used is 0.1. The corresponding values for the lower bound are independent from n and are $\{1.3\times10^{-5}, 8.7\times10^{-6}, 5.3\times10^{-6}, 3.2\times10^{-6}, 1.9\times10^{-6}, 1.2\times10^{-6}, 7.1\times10^{-7}\}$, respective to the x-axis. We observe that the upper bound is validated by the experiments as it is very close to the observed additive error. In addition, the additive error itself resulting from our algorithm is very small even for ϵ as small as 0.5. For $\epsilon = 0.1$ the additive error increase is unavoidable since such relatively high error may be necessary to protect the high privacy standard in this case.

7 Conclusion

We have presented a novel algorithm for estimating incidence counts of sanitized indicator vectors. It can also be used to estimate the n-wise inner product of sanitized bit vectors as the relationship is described in the proof of Theorem 5.10. We provided a theoretical upper bound that is validated by experiments on real-life datasets to be very accurate. Moreover, we extended a previous lower bound on 2-wise inner product to n-wise inner product. Finally, we evaluated our algorithm on a real-world dataset and validated the accuracy, the general upper bound and the lower bound.

References

[1] M. Alaggan, S. Gambs, and A.-M. Kermarrec. BLIP: Non-Interactive Differentially-Private Similarity Computation on Bloom Filters. In *Proceedings of the 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS12)*, Toronto, Canada, October, 2012. to appear.

[2] M. Alaggan, S. Gambs, S. Matwin, and M. Tuhin. Sanitization of Call Detail Records via Differentially-Private Bloom Filters. In P. Samarati, editor, *Data and Applications Security and Privacy XXIX - 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, volume 9149 of *Lecture Notes in Computer Science*, pages 223–230. Springer, 2015.

[3] R. Balu, T. Furon, and S. Gambs. Challenging Differential Privacy: The Case of Non-Interactive Mechanisms. In *ESORICS*, pages 146–164, 2014.

[4] M. V. Barbera, A. Epasto, A. Mei, S. Kosta, V. C. Perta, and J. Stefa. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Downloaded from http://crawdad.org/sapienza/probe-requests/20130910, Sept. 2013.

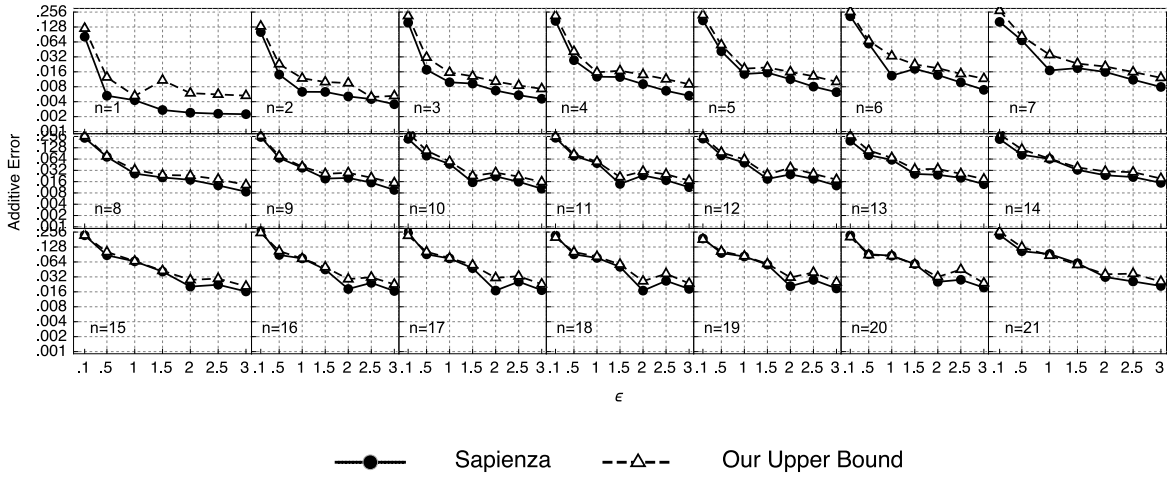

Figure 2: The additive error $\left\|\hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}'\right\|_\infty$ (on the y-axis), is plotted against the differential privacy parameter ϵ (on the x-axis), and the number of vectors n (in different subplots). The y-axis is in logarithmic scale while the x-axis is in linear.

[5] S. Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag London, 2001.

[6] G. Cormode and M. Hadjieleftheriou. Finding the Frequent Items in Streams of Data. *Commun. ACM*, 52(10):97–105, Oct. 2009.

[7] M. Datar and S. Muthukrishnan. Estimating Rarity and Similarity over Data Stream Windows. In R. H. Möhring and R. Raman, editors, *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'02)*, volume 2461 of *Lecture Notes in Computer Science*, pages 323–334, Rome, Italy, Sept. 2002. Springer.

[8] C. Dwork. Differential Privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06), Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12, Venice, Italy, 2006. Springer.

[9] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin. Pan-Private Streaming Algorithms. In A. C. Yao, editor, *Proceedings of the 1st Symposium on Innovations in Computer Science (ICS'10)*, pages 66–80, Tsinghua University, Beijing, China, 2010. Tsinghua University Press.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[11] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan. The Limits of Two-Party Differential Privacy. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:106, 2011.

[12] F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. In *Proceedings o the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103, Providence, RI, USA, Oct. 2007. IEEE Computer Society.

[13] D. J. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright. Pan-Private Algorithms via Statistics on Sketches. In M. Lenzerini and T. Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 37–48. ACM, 2011.

[14] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.

[15] R. P. Stanley. *Enumerative Combinatorics: Volume 1*. Cambridge University Press, New York, NY, USA, 2nd edition, 2011.

A The Probability Mass Function (PMF) of $Z(j)$

Since $Z(j) = \text{Binomial}(j, 1-p) + \text{Binomial}(n-j, p)$ then its PMF is equivalent to the convolution: $\Pr[Z(j) = i] = \sum_\ell \Pr[\text{Binomial}(j, 1-p) = \ell] \Pr[\text{Binomial}(n-j, p) = i-\ell]$. Consider one term in the summation, t_ℓ, which equals $\left[(1-p)^\ell p^{j-\ell}\binom{j}{\ell}\right]\left[p^{i-\ell}(1-p)^{-i-j+n+\ell}\binom{n-j}{i-\ell}\right]$. Since the ratio $\frac{t_{\ell+1}}{t_\ell} = \frac{(\ell-i)(\ell-j)}{(\ell-i-j+n+1)(\ell+1)}\left(\frac{p-1}{p}\right)^2$ is a rational function in ℓ then the summation over ℓ can be represented as a hypergeometric function: $\Pr[Z(j) = i] = t_0 \cdot {}_2F_1\left(\begin{smallmatrix}-i;\ -j\\ -i-j+n+1\end{smallmatrix} \mid \left(\frac{p-1}{p}\right)^2\right)$, in which $t_0 = (1-p)^{n-i-j}p^{i+j}\binom{n-j}{i}$, given that $i+j \leqslant n$. The case of $i+j > n$ is computed by symmetry as in (5). The notation ${}_2F_1$ denotes the Gauss hypergeometric function ${}_2F_1\left(\begin{smallmatrix}a;\ b\\ c\end{smallmatrix} \mid z\right) = \sum_{k\geqslant 0}\frac{a^{\overline{k}}b^{\overline{k}}}{c^{\overline{k}}}\frac{z^k}{k!}$, in which $x^{\overline{k}} = x(x+1)\cdots(x+k-1)$ is the rising factorial notation, also known as the Pochhammer symbol $(x)_k$.

B Bounding Deviation of A'' from its Mean

Theorem B.1 (Bounding deviation of A'')**.** *Let $\alpha = f(\delta)/2$ and β be positive real numbers less than one. Then with probability at least $1-\beta$, if $m = \Omega\left(-\ln(\beta)\ln(n)\alpha^{-2}\right)$, then $\left\|A\hat{\mathbf{\Phi}} - A''\hat{\mathbf{\Phi}}\right\|_\infty \leqslant \alpha$.*

Proof. Using Lemma B.2 and the fact that $E(n,x)$ approaches $\sqrt{Z - \ln(\pi Z - \pi\ln(\pi))}/\sqrt{2}$, as n approaches ∞ (according to its expansion at $n\to\infty$), in which $Z(x) = \ln(2n^2/\ln^2(4/x))$. This is a good approximation even for $n \geqslant 1$ except for $x = 1$ it becomes a good approximation for $n \geqslant 4$. Let $E_x = E(n,x)$ and $C = C(x)$, then using Lemma B.2, $m = \Omega(\alpha^{-2}\{E_2 + C[E_3 - E_1]\}^2) = \Omega(\alpha^{-2}\{E_2^2 + C^2[E_3 - E_1]^2\}) = \Omega(\alpha^{-2}\{E_2^2 + C^2[E_3^2 - E_1^2]\}) = \Omega(\alpha^{-2}C^2E_3^2) = \Omega(\alpha^{-2}\ln(1/\beta)\ln(n))$. □

Lemma B.2. *Let $\alpha, \beta < 1$ be positive real numbers. Then with probability at least $1-\beta$, if $2m \geqslant \left(\alpha^{-1}D(n,\beta)\right)^2$, then $\left\|A\hat{\mathbf{\Phi}} - A''\hat{\mathbf{\Phi}}\right\|_\infty \leqslant \alpha$, in which $D(n,\beta) = E(n,2) + C(1-\beta)(E(n,3) - E(n,1))$, $E(n,x) = \text{erf}^{-1}((4/x)^{-1/(n+1)})$, $C(x) = c_0\ln\log_2(1/x)$, and $c_0 = 1/\ln\log_4(4/3)$.*

Proof. Using Lemma B.3. Since A goes to a rank-1 matrix as fast as η^n (its smallest eigenvalue), we see that for every i, $A_{ij}(1-A_{ij})$ approaches a value that does not depend on j, call it p_i. Therefore, $\sum_j 2\Phi_j A_{ij}(1-A_{ij})$ approaches $2p_i\sum_j\Phi_j = 2p_i m$, a value which does not depend on the particular, unknown, composition. Since thus the choice of the weak $(n+1)$-composition Φ of m does not matter, we set $\Phi_j = m/(n+1)$ in the statement of Lemma B.3 and proceed[3]. Therefore $F(x)$ approaches $F_{\eta\downarrow}(x) \stackrel{\text{def}}{=} \prod_i \text{erf}\left(x/\sqrt{2mp_i}\right)$. We could compute the limit p_i if we require dependence on η for fine tuning. However, we will instead use the

[3] If η is not small enough, then the probability matrix A approaches the identity matrix I and Ψ, a known quantity, becomes close to the unknown quantity Φ. Hence, we can substitute it instead. For practical purposes, if this is the case, then we would not need this lemma and could using Lemma B.3 directly. In special cases where $\mathbf{\Phi}$ is close to uniform, we may quantify the probability of setting $\Phi_j = m/(n+1)$ by Lemma C.1.

bound $p_i = A_{ij}(1 - A_{ij}) \leqslant 1/4$ (for any j, since in the limit $A_{ij}(1 - A_{ij}) = A_{ik}(1 - A_{ik})$ for all j, k). The bound holds since A_{ij} is a probability value in $(0, 1)$ and the maximum of the polynomial $x(1 - x)$ is $1/4$. Consequently, $F_{\eta\downarrow}(x) \geqslant \operatorname{erf}\left(x\sqrt{2/m}\right)^{n+1}$. Therefore $F^{-1}_{\eta\downarrow}(q) = (\sqrt{m/2})\operatorname{erf}^{-1}(q^{1/(n+1)})$. Hence, $\mathcal{M} + \mathcal{R}C(1 - \beta) = (\sqrt{m/2})D(n, \beta)$. □

Lemma B.3. *Let* $F(x) = \prod_i \operatorname{erf}\left(x/\sqrt{\sum_j 2\Phi_j A_{ij}(1 - A_{ij})}\right)$ *be a cumulative distribution function (CDF) and let* $\mathcal{M} = F^{-1}(1/2)$ *and* $\mathcal{R} = F^{-1}(3/4) - F^{-1}(1/4)$ *be the median and the interquantile range of the distribution represented by* F*, respectively. Additionally, let* $C(x) = c_0 \ln\log_2(1/x)$*, in which* $c_0 = 1/\ln\log_4(4/3)$*. Then, if* $\alpha < 1$ *is a positive real number, then with probability at least* $1 - \beta$ *we have that* $\left\|A\hat{\mathbf{\Phi}} - A''\hat{\mathbf{\Phi}}\right\|_\infty \leqslant m^{-1}(\mathcal{M} + \mathcal{R}C(1 - \beta))$.

Proof. Consider the following transformation of the random variable A'':

$$
\begin{aligned}
&m\left\|A\hat{\mathbf{\Phi}} - A''\hat{\mathbf{\Phi}}\right\|_\infty = m\left\|(A - A'')(\mathbf{\Phi}/m)\right\|_\infty \\
&= m\left\|(A - A'\mathrm{diag}(1/\mathbf{\Phi}))(\mathbf{\Phi}/m)\right\|_\infty \\
&= m\left\|(A\mathrm{diag}(\mathbf{\Phi}) - A')\mathrm{diag}(1/\mathbf{\Phi})(\mathbf{\Phi}/m)\right\|_\infty \\
&= m\left\|(A\mathrm{diag}(\mathbf{\Phi}) - A')\mathbf{1}/m\right\|_\infty \\
&= \left\|(A\mathrm{diag}(\mathbf{\Phi}) - A')\mathbf{1}\right\|_\infty \\
&= \left\|A\mathbf{\Phi} - A'\mathbf{1}\right\|_\infty = \max_i\left\{|A_{i\bullet}\mathbf{\Phi} - A'_{i\bullet}|\right\} \\
&= \max_i\left\{\left|\sum_j A_{ij}\Phi_j - A'_{ij}\right|\right\} \\
&\lesssim \max_i\left\{\left|\sum_j A_{ij}\Phi_j - \mathrm{Binomial}(\Phi_j, A_{ij})\right|\right\} \quad ,
\end{aligned}
$$

since the marginal distribution of a multinomial random variable is the binomial distribution, which in turn converges in distribution to the normal distribution, by the centrali limit theorem, as $\min_j \Phi_j$ grows (which is justified in Lemma C.1),

$$
\begin{aligned}
&\xrightarrow{d} \max_i\left\{\left|\sum_j A_{ij}\Phi_j - \mathcal{N}(\Phi_j A_{ij}, \sigma^2_{ij} = \Phi_j A_{ij}(1 - A_{ij}))\right|\right\} \\
&= \max_i\left\{\left|\sum_j \mathcal{N}(0, \sigma^2_{ij})\right|\right\} = \max_i\left\{\left|\mathcal{N}(0, \sum_j \sigma^2_{ij})\right|\right\} \\
&= \max_i\left\{\mathrm{HalfNormal}\left(\theta^2_i = \frac{\pi}{2\sum_j \sigma^2_{ij}}\right)\right\} \ ,
\end{aligned}
$$

which, as n grows, converges in distribution to the Gumbel distribution, by the extreme value theorem [5]: $\xrightarrow{d} G(a\mathcal{R} + \mathcal{M}, b\mathcal{R})$, in which $\mathcal{R} = F^{-1}(3/4) - F^{-1}(1/4)$, $\mathcal{M} = F^{-1}(1/2)$, $a = -\ln(\ln 2)/\ln(\log_4(4/3))$, and $b = -1/\ln(\log_4(4/3))$, where $F(x) = \prod_i F_i(x)$ and $F_i(x)$ is the CDF of $\mathrm{HalfNormal}\left(\theta^2_i\right)$. Therefore, computing the quantile function of the Gumbel distribution at $1 - \beta$ shows that with probability at least $1 - \beta$ we have $m\left\|A\hat{\mathbf{\Phi}} - A''\hat{\mathbf{\Phi}}\right\|_\infty \leqslant \mathcal{M} + \mathcal{R}(a - b\ln(-\ln(1 - \beta))) = \mathcal{M} + \mathcal{R}\left[\frac{\ln(-\log_2(1-\beta))}{\ln(\log_4(4/3))}\right]$. The last convergence result is due the fact that maximums approach Gumbel and therefore we choose a Gumbel distribution matching the median and interquantile range of the actual distribution of the maximum of HalfNormals, whose CDF is the multiplication of their CDFs. This is done by setting a and b to be the parameters of a Gumbel distribution $G(a, b)$ with zero median and interquantile range of one, and then using the fact that Gumbel distribution belong to a location-scale family, which also implies that the Gumbel distribution is uniquely defined by its median and interquantile range (two unknowns and two equations). □

C Discrete Uniform Distribution on Φ

We treat the case of uniform streams in this section. We call the vector $\boldsymbol{a} = (a_1, \ldots, a_m)$ uniform if a_i is uniform on the range $\{0, \ldots, n\}$. Considering the *marginal distribution* of the resulting incidence vector, we observe that in this case $\mathbb{E}\Phi_i = \mathbb{E}\Phi_j$ for all i, j. However, Φ_i will be strongly concentrated around

its mean. Therefore, we consider an even *stronger* model in which $\mathbb{E}\Phi_i$ is still equal to $\mathbb{E}\Phi_j$ for all i, j, but Φ_i is marginally almost uniform on its range. An algorithm doing well in this latter case (with higher variance) can intuitively do at least as well in the former case (with less variance).

The vector $\mathbf{\Phi}$ is a vector of $n+1$ elements but only n degrees of freedom; since it has to sum to m. Therefore, we cannot consider the discrete uniform product distribution on its entries. Instead, we will consider the joint uniform distribution on all non-negative integer vectors which sum to m. All such vectors are the set of weak $(n+1)$-compositions of m [15, p. 25].

Lemma C.1. *If $m \geqslant \frac{(n+1)\delta}{1-\sqrt[n]{1-\beta}}$, then with probability at least $1-\beta$, $\min_j \Phi_j \geqslant \delta$, assuming $\mathbf{\Phi}$ are picked uniformly at random from all weak $(n+1)$-compositions of m.*

Proof. Notice that the sum of $\mathbf{\Phi}$ must be m, therefore, it has only n degrees of freedom instead of $n+1$. In fact, $\mathbf{\Phi}$ is the multivariate uniform distribution on weak $(n+1)$-compositions[4] of m. Notice that the marginal distribution of Φ_j is *not* $\mathrm{Uniform}(0, m)$, but rather lower values of Φ_j have strictly higher probability than greater ones.

Consider the compositions of m into exactly $n+1$ parts, in which each part is greater than or equals δ. There is exactly[5] $C_{n+1}(m;\delta) \stackrel{\text{def}}{=} C_{n+1}(m-(n+1)\delta)$ such compositions. Hence, the *joint* probability that all entries of $\mathbf{\Phi}$ exceed a desired threshold δ, *simultaneously*, is $\frac{C_{n+1}(m;\delta)}{C_{n+1}(m)}$.

In the rest of this proof we will use $\left[{n \atop k}\right]$, the unsigned Stirling cycle number (*i.e.* Stirling numbers of the first kind), $x^{\underline{n}} = x(x-1)\cdots(x-(n-1))$ the falling factorial power, and $x^{\overline{n}} = x(x+1)\cdots(x+(n-1))$ the rising factorial power. We will also use the identity $x^{\overline{n}} = \sum_k \left[{n \atop k}\right] x^k$. All the definitions and a proof of the aforementioned identity could be found in [10, p. 264].

$$\frac{C_{n+1}(m;\delta)}{C_{n+1}(m)} = \frac{\binom{m+n-(n+1)\delta}{n}}{\binom{m+n}{n}} = \frac{(m+n-(n+1)\delta)^{\underline{n}}}{(m+n)^{\underline{n}}}$$
$$= \frac{\prod_{i=0}^{n-1}(m+n-(n+1)\delta-i)}{\prod_{i=0}^{n-1}(m+n-i)}$$
$$= \frac{\prod_{i=1}^{n}(m+n-(n+1)\delta-(n-i))}{\prod_{i=1}^{n}(m+n-(n-i))}$$
$$= \frac{(m+1-(n+1)\delta)^{\overline{n}}}{(m+1)^{\overline{n}}} \ ,$$

and then, substituting $\mu = m+1$ and $\nu = n+1$ for readability

$$\frac{C_{n+1}(m;\delta)}{C_{n+1}(m)} = \frac{(\mu-\nu\delta)^{\boldsymbol{\nu-1}}}{\mu^{\boldsymbol{\nu-1}}} \geqslant 1-\beta$$
$$\iff (\mu-\nu\delta)^{\boldsymbol{\nu-1}} \geqslant (1-\beta)\mu^{\boldsymbol{\nu-1}}$$
$$\iff \sum_k \left[{\nu-1 \atop k}\right](\mu-\nu\delta)^k \geqslant \sum_k \left[{\nu-1 \atop k}\right](1-\beta)\mu^k$$
$$\iff \sum_k \left[{\nu-1 \atop k}\right]((\mu-\nu\delta)^k - (1-\beta)\mu^k) \geqslant 0 \ ,$$

which is true when the sufficient condition $(\mu-\nu\delta)^k - (1-\beta)\mu^k \geqslant 0$ holds for all $1 \leqslant k \leqslant n$. Equivalently, when

$$(\mu-\nu\delta)^k \geqslant (1-\beta)\mu^k \iff \left(\frac{\mu-\nu\delta}{\mu}\right)^k \geqslant (1-\beta)$$
$$\iff k\ln(\frac{\mu-\nu\delta}{\mu}) \geqslant \ln(1-\beta)$$
$$\iff \frac{\mu-\nu\delta}{\mu} \geqslant \exp(\ln(1-\beta)/k)$$
$$\iff \frac{\mu-\nu\delta}{\mu} \geqslant \sqrt[k]{1-\beta} \iff 1-\frac{\nu\delta}{\mu} \geqslant \sqrt[k]{1-\beta}$$
$$\iff \frac{\nu\delta}{\mu} \leqslant 1-\sqrt[k]{1-\beta} \impliedby m \geqslant \frac{(n+1)\delta}{1-\sqrt[n]{1-\beta}} - 1 \ \square$$

[4] A weak k-composition of an integer n is a way of writing n as the sum of k non-negative integers (zero is allowed) [15, p. 25]. It is similar to integer partitions except that the order is significant. The number of such weak compositions is $C_k(n) = \binom{n+k-1}{k-1}$.

[5] There are $n+1$ urns, each already have at least δ balls. Thus $m-(n+1)\delta$ balls remain to distribute into $n+1$ urns.