Sketch-Flip-Merge: Mergeable Sketches for Private Distinct Counting

Jonathan Hehir [1 2] **Daniel Ting** [3] **Graham Cormode** [3 4]

Abstract

Data sketching is a critical tool for distinct counting, enabling multisets to be represented by compact summaries that admit fast cardinality estimates. Because sketches may be merged to summarize multiset unions, they are a basic building block in data warehouses. Although many practical sketches for cardinality estimation exist, none satisfy both privacy and mergeability constraints. We propose the first practical cardinality sketches that are simultaneously mergeable, differentially private (DP), and have low empirical errors. These introduce a novel randomized algorithm for performing logical operations on noisy bits, a tight privacy analysis, and provably optimal estimation. Our sketches dramatically outperform existing theoretical solutions in simulations and on real-world data.

1. Introduction

Many applications that model large volumes of data are based on tracking cardinalities of events or observations. Consequently, these applications make extensive use of data sketches that support fast, approximate cardinality estimation (Cormode & Yi, 2020). For instance, approximate distinct counting is supported via variants of the HyperLogLog (HLL) sketch (Flajolet et al., 2007; Heule et al., 2013) in popular data management systems including Amazon Redshift, ClickHouse, Google BigQuery, Splunk, Presto, Redis, and more. At the expense of a small estimation error, these approximate methods drastically reduce the computational cost of distinct counting to run in linear time, using only bounded memory. An additional key feature of distinct-count sketches is the ability to merge two or more sketches to obtain cardinality estimates over their union. This enables not only distributed computation, but also many rich aggregation possibilities from previously computed sketches. As a result, modern data pipelines rely extensively on the performance and functionality of such cardinality sketches.

Increasingly, privacy concerns constrain the operation of data processing. Organizations demonstrating commitments to preserving users' privacy require that data collected from individuals be subject to appropriate mitigations before being passed to downstream processing. Specifically, protections such as differential privacy are used to protect sensitive data while still giving accurate query response.

Although sketching techniques may appear to offer protection by reducing data, it is well-known that sketching alone does not automatically provide a privacy guarantee (Desfontaines et al., 2019). The summaries—or even the estimates calculated from them—can leak considerable information about whether the specific items belong to the underlying set. Recently, it has been shown that the contents of sketches do meet a privacy standard *if* the associated hash functions are not known to the observer (Choi et al., 2020; Smith et al., 2020; Dickens et al., 2022). However, it is not plausible to assume secret hash functions when the computation is shared among multiple entities in a large scale system. In particular, all participants must know the hash when working with sketches that will be merged, and using the same hash in multiple sketches generates correlated randomness that breaks the privacy guarantees. This creates an important gap to make these high-throughput systems private. Previous attempts to construct privacy-preserving sketches (Pagh & Stausholm, 2021) do not offer practical mergeable sketches as the errors are too large (Section 6).

In this work, we present the Sketch-Flip-Merge (SFM) summaries, a practical, mergeable, and provably private approach to distinct-count sketching. In particular, we produce summaries that satisfy the strong definition of ε–differential privacy (DP) (Dwork et al., 2006; Dwork, 2008) even when the hash function is known publicly. By attaching the privacy guarantee to the summary itself—not just the cardinality estimate—we may safely release summaries corresponding to sensitive multisets, enabling safe cardinality estimation over any union of such sets using the privacy-preserving summaries in lieu of the original sensitive data.

The key to our approach is to adapt the sketch of Flajolet

[1]Department of Statistics, Penn State University, State College, PA, USA [2]Work performed while at Meta. [3]Meta, Seattle, WA, USA [4]University of Warwick, Coventry, UK. Correspondence to: Daniel Ting <dting@meta.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

& Martin (1985), which is often referred to as either *FM85* or *probabilistic counting with stochastic averaging (PCSA)*. Although subsequent sketches such as HLL (Durand & Flajolet, 2003; Flajolet et al., 2007; Heule et al., 2013) further optimized the space usage, squeezing the space makes them less amenable to privacy protection. In contrast to PCSA where the simple, partitioned binary structure limits the sensitivity to a bit flip, these sketches store extremal hash values where small changes to the input can cause big changes in the summary, requiring more noise and yielding less accurate results. Furthermore, our methods generalize to any bitmap based sketch.

Related Work. Privacy-preserving cardinality sketches are the subject of several earlier works. Mir et al. (2011) offer an early theoretical solution under DP that lacks practicality due to reliance on an inefficient exponential mechanism and absence of a merge operation. Recent efforts provide DP guarantees for practical HLL-like sketches (Smith et al., 2020; Dickens et al., 2022), but these rely on random, secret hash functions that preclude the ability to merge sketches. Using a fixed, public hash, Choi et al. (2020) obtain a DP cardinality estimate from a LogLog sketch by adding noise to the cardinality estimator, but the sketch itself remains sensitive and unsafe for release or sharing.

One line of work extends PCSA with randomized response and subsampling of items to achieve privacy (Tschorsch & Scheuermann, 2013; Nuñez von Voigt & Tschorsch, 2019). However, Tschorsch & Scheuermann (2013) fails to achieve a DP guarantee, and Nuñez von Voigt & Tschorsch (2019) does not address merging sketches. Kreuter et al. (2020) design two sketches, including one based on PCSA. While their DP sketches cannot be merged to form a single sketch, multiple sketches may be used to estimate the union's cardinality if all sketches use the same privacy parameters. The PCSA-based sketch of Pagh & Stausholm (2021, Section 6) achieves DP and supports merging but is impractical. In our experiments, their estimator frequently failed to produce an estimate and returned impractically large errors.

Private cardinality estimation is achieved via perturbed Bloom filters in Stanojevic et al. (2017) and Alaggan et al. (2018). As in Kreuter et al. (2020), neither of these solutions produce mergeable sketches, but multiple sketches may be used to estimate set union cardinality in certain scenarios. In particular, the solution of Stanojevic et al. (2017) does not scale beyond two sketches, while Alaggan et al. (2018) requires all sketches to use identical privacy parameters.

Finally, Desfontaines et al. (2019) give an impossibility result where both privacy and high accuracy are impossible, but only when many sketches are merged, which is consistent with our results.

Contributions. We propose two practical methods for constructing *mergeable* DP cardinality sketches and obtaining cardinality estimates. The first uses a deterministic bit-merging operation used by Pagh & Stausholm (2021). We prove this merge requires a suboptimal form of randomized response, even after exponential improvement to the prior privacy analysis (Corollary 4.5). Our main methodological contribution is a novel randomized merge allowing for up to a further 75% variance reduction over the optimized deterministic merge. We generalize our randomized merge to perform to arbitrary bitwise operations on binary data that may be of independent interest. We also develop a composite likelihood-based estimator for cardinality and prove this estimator is asymptotically optimal for both private and non-private sketches based on PCSA.

Outline. Sections 2 and 3 review background material, first on non-private distinct counting sketches and the PCSA sketch that underlies our private sketch, then on differential privacy and the naive application of randomized response to non-mergeable sketches. Section 4 describes our main contribution to private sketch construction: designing mergeable sketches with randomized response. Section 5 shows how to extract cardinality estimates using likelihood based estimators. These enable us to obtain the first private sketches with practical sizes and errors. We further analyze their asymptotic error and prove their optimality. Section 6 empirically compares our methods with private and non-private alternatives. All proofs are deferred to the appendices.

Notation. We write $[m] = \{1, \ldots, m\}$. $\otimes$ denotes the Kronecker product. Logical operations are denoted $\vee$ (or), $\wedge$ (and), $\veebar$ (xor), and $\neg$ (not). We use the natural logarithm $\log = \log_e$. Equality in distribution is denoted $\overset{D}{=}$. The cardinality of a set D is denoted $|D|$.

2. Background and Problem Setup

Let $D \in \mathcal{X}^N$ denote a multiset of N items from some universe $\mathcal{X}$ of objects. The *count-distinct problem* is the task of estimating the number of unique elements in D. That is, if $\texttt{set}(D)$ denotes the support set of items in D, the count-distinct problem aims to approximate $n = |\texttt{set}(D)|$ with a data sketch in bounded memory in a single pass over the data. We consider the *private* count-distinct problem for *mergeable* sketches where the information in a sketch satisfies differential privacy (DP) and sketches can be merged to obtain a sketch of the union of underlying datasets.

We focus on solutions to the count-distinct problem in which sketches form a binary vector, subject to merge operations performed through element-wise logical operations (e.g., *or*). The class of sketches to which our methods apply include PCSA, linear counting (Whang et al., 1990), Bloom filters (Broder & Mitzenmacher, 2004), and Liquid Legions

(Kreuter et al., 2020). These are particularly amenable to privacy enhancement through the application of randomized response (Warner, 1965) but require careful design of merge operations for randomized bits. Although this excludes other commonly used sketches such as HyperLogLog and the k-minimum value sketch (Bar-Yossef et al., 2002; Giroire, 2009), the richer set of values stored in these sketches make them less suitable for privatization due to their high sensitivity and, hence, higher noise required for privacy. In the remainder of this paper, we focus on the PCSA sketch of Flajolet & Martin (1985), noting that the results for constructing and merging private sketches in Sections 3 and 4 apply to related sketches through direct application or simple extensions.

The classical PCSA sketch takes the form of a matrix $S = \mathcal{S}(D) \in \{0,1\}^{B\times P}$ with B buckets and precision parameter P. Given two independent, universal hash functions, $h_1(x) \sim \text{Uniform}([B]), h_2(x) \sim \text{Geometric}(1/2)$, let $\texttt{bucket}(x) = h_1(x), \texttt{value}(x) = \min\{P, h_2(x)\}$. Then each bit S_{ij} is equal to 1 iff there exists $x \in D$ such that $\texttt{bucket}(x) = i, \texttt{value}(x) = j$. Some desirable properties of S are immediate. First, S relies only on the set of hashed values $\{h_1(x)\}_{x\in D}$ and $\{h_2(x)\}_{x\in D}$. Hence, it is invariant both to repetitions in D and to the order in which the elements of D are processed. Additionally, two sketches $\mathcal{S}(D_1)$ and $\mathcal{S}(D_2)$ may be merged via a simple bitwise-*or*, $\mathcal{S}(D_1) \vee \mathcal{S}(D_2) = \mathcal{S}(D_1 \cup D_2)$, as each entry S_{ij} in the merged sketch is equal to 1 iff there is an item x in at least one of D_1, D_2 for which $\texttt{bucket}(x) = i, \texttt{value}(x) = j$.

Importantly, when an adversary knows h_1 and h_2, the sketch $\mathcal{S}(D)$ reveals information about elements in D. For example, any $x \in \mathcal{X}$ for which $S_{\texttt{bucket}(x),\texttt{value}(x)} = 0$ cannot belong to D. In what follows, we extend the PCSA sketch to minimize this sort of privacy leakage.

3. Adding Privacy to Sketches

Differential privacy (DP) (Dwork et al., 2006; Dwork, 2008) offers a strong and quantifiable notion of privacy. DP mandates that algorithms (*privacy mechanisms*) acting on a dataset D must be randomized—typically through the addition of some carefully tuned noise—so that the distribution of a privacy mechanism's output cannot be significantly influenced by a single input record. As a result, the ability to reverse-engineer information about a single record is limited, and any analysis performed using only the output of the algorithm also satisfies DP. The strength of the DP guarantee is quantified by the parameter $\varepsilon > 0$, often called the privacy budget, with smaller ε offering stronger privacy.

Definition 3.1 (Dwork et al. (2006))**.** A randomized algorithm $\mathcal{M}$ is said to satisfy **ε–differential privacy (DP)** if for any two neighboring databases D, D' and any set of outputs $E \subseteq \text{Range}(\mathcal{M})$, we have:

$$\Pr(\mathcal{M}(D) \in E) \leq e^{\varepsilon} \Pr(\mathcal{M}(D') \in E).$$

In the count-distinct problem, we say two multisets D, D' are neighbors if D' can be obtained by adding or removing one unique item to D. Given two neighboring multisets D, D' and their corresponding PCSA sketches $\mathcal{S}(D), \mathcal{S}(D')$, it follows from the definition of PCSA that these sketches must agree on all but at most one bit. To create DP sketches from $\mathcal{S}(D)$, then, we consider general DP mechanisms applied to vectors of $\{0,1\}$ bits, where two vectors $x, x' \in \{0,1\}^d$ neighbor if they differ on at most one bit (i.e., have *sensitivity* 1).

When restricting our attention to mechanisms whose input and output are both a single bit, every DP mechanism can be viewed as an instance of randomized response (RR) (Warner, 1965). We describe a generalized form of RR as follows. Let $\mathcal{F}_{p,q}$ denote a general bit-flipping algorithm, parameterized by two probabilities p and q, and where classical RR (in the style of Warner) is recovered when $q = 1 - p$:

$$\mathcal{F}_{p,q}(x) \sim \text{Bernoulli}(p \text{ if } x = 1 \text{ else } q).$$

Theorem 3.2. *Assume $q \leq 1/2 \leq p$. Applied to vectors with sensitivity 1, the algorithm $\mathcal{M}_{p,q} : \{0,1\}^d \to \{0,1\}^d$ that independently applies $\mathcal{F}_{p,q}$ to each element of its input is ε-DP if and only if:*

$$p, q \in (0,1) \quad \textit{and} \quad \max\left\{\frac{p}{q}, \frac{1-q}{1-p}\right\} \leq e^{\varepsilon}. \tag{1}$$

Theorem 3.2 provides an entire family of privacy mechanisms, any $\mathcal{M}_{p,q}$ with p, q satisfying Eq. 1, that can be applied to a PCSA sketch or any bit vector to make its output ε-DP. Our contribution is then to address several important questions: How can we merge two sketches if their bits have been perturbed via $\mathcal{F}_{p,q}$? How can we estimate cardinality from noisy sketches? And how should we choose p and q?

4. Designing Mergeable Sketches

While a randomized response mechanism $\mathcal{M}_{p,q}$ converts a PCSA sketch to a private equivalent, this breaks PCSA's merge operation. For ordinary PCSA and multisets D_1, D_2, the bitwise-*or* $\mathcal{S}(D_1) \vee \mathcal{S}(D_2) = \mathcal{S}(D_1 \cup D_2)$ defines a merge operation on sketches that yields the same sketch that would be obtained by first taking the union. However, the same operation on noisy sketches does not satisfy this desirable property. We develop merge operations on noisy sketches and identify under what conditions they exist. In particular, Theorem 4.1 shows that if a merge operation is deterministic, then *xor* ($\veebar$) is the only possible merge on noisy sketches, and it only works for certain choices of the mechanism $\mathcal{M}_{p,q}$. We show these choices imply that, at

best, such a noisy sketch's cardinality estimates have $4\times$ worse variance than that for regular PCSA on the same sized sketch, even if the privacy budget is near-infinite. Our main contribution is to provide a novel *randomized* merge operation that adds less noise to the sketch. Furthermore, we generalize this operation to perform arbitrary boolean operations on noisy bit vectors.

4.1. Deterministic Merging

Applying the standard randomized response mechanism to a PCSA sketch breaks mergeability. PCSA merges sketches using bitwise-*or*, and in the presence of RR noise, the *or* operation results in non-RR noise that biases bits towards 1. Pagh & Stausholm (2021) address this by replacing bitwise-*or* ($\vee$) with bitwise-*xor* ($\veebar$) operations whenever the sketch is updated or merged. However, the *xor* operation destroys cardinality information. In particular, the *xor* of a PCSA sketch with itself is the empty sketch. Rather than ensuring sketches are invariant to duplicates, they ensure the *distribution* of (merged) sketches are invariant. They do this by subsampling items (including duplicates) independently with probability $1/2$. This effectively encodes bits that were 1 in PCSA as $\mathrm{Bernoulli}(1/2)$ values, while 0 bits remain 0. Unfortunately, this subsampling operation introduces a lot of noise. Figure 2 shows that even for large ε the resulting cardinality estimates have 4 times the variance.

We show that this penalty on the accuracy is inherent for any *deterministic* merge. Theorem 4.1 shows *xor* is, in fact, the only possible way to merge deterministically, so that randomized merges are the only way to improve merging. Our analysis also improves the Pagh & Stausholm (2021) sketch by significantly reducing the noise required for an ε-DP privacy guarantee and demonstrates how to merge sketches with different privacy budgets.

Theorem 4.1. *Let $f_1 = \mathcal{F}_{p_1,q_1}, f_2 = \mathcal{F}_{p_2,q_2}, f_3 = \mathcal{F}_{p_3,q_3}$, and let $\circ : \{0,1\}^2 \to \{0,1\}$ denote a deterministic and symmetric operation. The following conditions may only be satisfied simultaneously if $\circ = \veebar$ and $p_1 = p_2 = 1/2$:*

1. *f_1, f_2 are ε_1-DP and ε_2-DP for $\varepsilon_1, \varepsilon_2 < \infty$.*
2. *$f_1(x) \circ f_2(y) \overset{D}{=} f_3(x \vee y)$.*
3. *$f_i(0) \overset{D}{\neq} f_i(1)$ for $i = 1, 2, 3$.*

Using our general family of RR mechanisms, we define a mechanism that adds noise to a PCSA sketch to provide privacy (Lemma 4.3) while preserving mergeability (Theorem 4.4). Corollary 4.5 shows our privacy analysis is much tighter than that of Pagh & Stausholm (2021).

Definition 4.2. For $\varepsilon > 0$, let $\mathcal{M}_\varepsilon^{\mathrm{xor}} : \{0,1\}^d \to \{0,1\}^d$ denote the mechanism that independently applies an asymmetric random response $\mathcal{F}_{p,q}$ to each element of its input with $p = 1/2, q = 1/(2e^\varepsilon)$.

Lemma 4.3. *$\mathcal{M}_\varepsilon^{\mathrm{xor}}$ is ε–differentially private.*

Theorem 4.4 (Deterministic Mergeability)**.** *$\mathcal{M}_{\varepsilon_1}^{\mathrm{xor}}(x) \veebar \mathcal{M}_{\varepsilon_2}^{\mathrm{xor}}(y) \overset{D}{=} \mathcal{M}_{\varepsilon^*}^{\mathrm{xor}}(x \vee y)$, where $\varepsilon^* = -\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$.*

Corollary 4.5. *Let $\mathcal{M}_\varepsilon^{\mathrm{PS}}$ denote the ε-DP privacy mechanism of Pagh & Stausholm (2021, Section 6). Then $\mathcal{M}_\varepsilon^{\mathrm{xor}} = \mathcal{M}_{2(\exp(\varepsilon)-1)}^{\mathrm{PS}}$.*

This tighter privacy analysis[1] dramatically reduces noise added to achieve the privacy guarantee, effectively increasing the privacy budget by at least a factor of 2. Pragmatically, Figure 2 shows that error increases exponentially as $\varepsilon \to 0$.

4.2. Randomized Merging

Theorem 4.1 showed that a deterministic merge is only possible if the 1-bits in a PCSA sketch are randomized to $\mathrm{Bernoulli}(1/2)$ values. Thus, even if the privacy budget is nearly infinite, the mergeable DP sketch must add significant noise to the base PCSA sketch. We show that by moving randomness from the base sketch to the merge procedure, we can achieve lower overall noise while using the standard random response mechanism (Definition 4.6).

Definition 4.6. For $\varepsilon > 0$, we denote by $\mathcal{M}_\varepsilon^{\mathrm{sym}} : \{0,1\}^d \to \{0,1\}^d$ the mechanism that independently applies the standard RR mechanism $\mathcal{F}_{p,1-p}$ to each element of its input with $p = e^\varepsilon/(e^\varepsilon + 1)$.

Lemma 4.7. *$\mathcal{M}_\varepsilon^{\mathrm{sym}}$ is ε–differentially private.*

A merge is a randomized algorithm $g_{\varepsilon_1,\varepsilon_2} : \{0,1\}^2 \to \{0,1\}$ that commutes with $\vee$ in the following sense: $g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}_{\varepsilon_1}^{\mathrm{sym}}(x), \mathcal{M}_{\varepsilon_2}^{\mathrm{sym}}(y)) \overset{D}{=} \mathcal{M}_{\varepsilon^*}^{\mathrm{sym}}(x \vee y)$. Since $g_{\varepsilon_1,\varepsilon_2}$ is a random mapping from pairs of bits to single bits, we can represent it as a 4×2 Markov transition matrix. A valid merge operation is the solution of the resulting matrix equality, with ε^* a free parameter. We obtain a optimal randomized merge operation for $\mathcal{M}^{\mathrm{sym}}$ by solving for the largest ε^* that generates a valid solution, which is given by:

Theorem 4.8. *Assume $\varepsilon_1, \varepsilon_2 > 0$. Let $q(\varepsilon) = (e^\varepsilon + 1)^{-1}$,*

$$\varepsilon^* = -\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)}),\ q^* = q(\varepsilon^*),$$

$$K_i = \begin{bmatrix} 1 - q(\varepsilon_i) & q(\varepsilon_i) \\ q(\varepsilon_i) & 1 - q(\varepsilon_i) \end{bmatrix} \text{ for } i \in \{1, 2\}, \text{ and}$$

$$v^* = (q^*, 1 - q^*, 1 - q^*, 1 - q^*)^T,$$

Letting $\otimes$ denote the Kronecker product, define:

$$(t_{00}, t_{01}, t_{10}, t_{11})^T = (K_1^{-1} \otimes K_2^{-1})\, v^*, \text{ and}$$

[1] It is proven in the appendix of Pagh & Stausholm (2021) that $q = 1/(e^\varepsilon + 1)$ satisfies ε-DP, although the recommendation and main results in the paper rely on the choice of $q = 1/(2 + \varepsilon)$. Our recommendation of $q = 1/(2e^\varepsilon)$ is optimal under DP constraints.

$$g_{\varepsilon_1,\varepsilon_2}(a,b) \sim \text{Bernoulli}(t_{ab}), \quad a,b \in \{0,1\}.$$

Then $g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x), \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x \vee y)$*, where* g *is taken bitwise and independently.*

When the original vectors x and y are visible in addition to the merged vector, the ε^* parameter of Theorems 4.4 and 4.8 is best interpreted as a measure of utility in the merged sketch, rather than a privacy budget, since by the post-processing invariance of DP (Dwork et al., 2006), no additional privacy leakage occurs from the release of the merged vector. It is for this reason we seek the maximal ε^* in merging. Noting that Theorems 4.4 and 4.8 produce identical ε^* and that $\mathcal{M}^{\text{sym}}_{\varepsilon}$ is less noisy than $\mathcal{M}^{\text{xor}}_{\varepsilon}$, $\mathcal{M}^{\text{sym}}$ remains the preferred mechanism after merging.

Remark 4.9. By induction, the merges prescribed in Theorems 4.4 and 4.8 allow any k bit vectors of equal length $x_1, \ldots, x_k$ privatized using $\varepsilon_1, \ldots, \varepsilon_k$ to be merged, resulting in a vector $v = (x_1 \vee \cdots \vee x_k)$ privatized with

$$\varepsilon^* = -\log\left(1 - \prod_{i=1}^{k}(1 - e^{-\varepsilon_i})\right).$$

A natural question is whether there exists a randomized merge algorithm $g_{\varepsilon_1,\ldots,\varepsilon_k} : \{0,1\}^k \to \{0,1\}$ that satisfies a property like Theorem 4.8 with a larger ε^* than given by induction over the pairwise merges. In Appendix A, we prove a more general form of Theorem 4.8 (Theorem A.2), which answers this question in the negative.

4.3. General Boolean Operations

We briefly switch focus from distinct counting to present a generalization to Boolean operations under randomized response that may be of foundational interest, e.g., in designing intersection operations. In distinct-count sketches, set unions correspond to bitwise-*or* operations, and the challenge posed by privacy is performing an equivalent operation over noisy bits. PCSA, like other mergeable sketches, defines a homomorphism $\mathcal{S}$ from multisets to sketches. The commutative diagram below illustrates this mergeability property, since it does not matter which path one takes from D_1, D_2 to $v_1 \vee v_2$. Likewise, our merge operation g from Theorem 4.8 ensures that the privacy mechanism $\mathcal{M}$ makes the diagram commute. By preserving the structure of the union operation, inferences about the cardinality of the union can be made from merged, private sketches.

$$\begin{tikzcd}
\text{Sets} & \text{Bit Vectors} \arrow[r, "\mathcal{M}_\varepsilon"] & \text{DP Bit Vectors} \\
D_1, D_2 \arrow[r, "\mathcal{S}"] \arrow[d, "\cup"'] & v_1, v_2 \arrow[r, "\mathcal{M}_\varepsilon"] \arrow[d, "\vee"'] & \mathcal{M}_\varepsilon(v_1), \mathcal{M}_\varepsilon(v_2) \arrow[d, "g"'] \\
D_1 \cup D_2 \arrow[r, "\mathcal{S}"] & v_1 \vee v_2 \arrow[r, "\mathcal{M}_{\varepsilon^*}"] & \mathcal{M}_{\varepsilon^*}(v_1 \vee v_2)
\end{tikzcd}$$

Here, we generalize the *or* ($\vee$) merge under $\mathcal{M}^{\text{sym}}$ to any logical operation. In particular, Corollary 4.10 shows a simple change in our target probabilities v^* yields the appropriate randomized merge for *and* ($\wedge$), while Lemmas B.1 and B.2 in Appendix B demonstrate a merge for *xor* ($\veebar$) and show that *not* ($\neg$) commutes with $\mathcal{M}^{\text{sym}}$.

Table 1. Boolean Operations on Bit Vectors under $\mathcal{M}^{\text{sym}}$

Op. $\square$	DP Op.	$\varepsilon^*_\square(\varepsilon_1, \varepsilon_2)$
$\neg$	Lem. B.2	—
$\vee$	Thm. 4.8	$-\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$
$\wedge$	Cor. 4.10	$-\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$
$\veebar$	Lem. B.1	$\log\left(\frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}\right)$

Corollary 4.10. *Assume the setting of Theorem 4.8, but set* $v^* = (q^*, q^*, q^*, 1-q^*)^T$*. Then* $g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x), \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x \wedge y)$*, where* g *is taken bitwise and independently.*

Formally, for each binary logical operator $\square$, there is a function $\varepsilon^*_\square(\varepsilon_1, \varepsilon_2)$ combining two privacy budgets that endows pairs of bit vectors and privacy budgets with the semigroup structure $(v_1, \varepsilon_1) \cdot (v_2, \varepsilon_2) := (v_1 \,\square\, v_2, \varepsilon^*_\square(\varepsilon_1, \varepsilon_2))$. The privacy mechanism $\mathcal{M}^{\text{sym}}$ then defines a mapping $\phi(v, \varepsilon) := (\mathcal{M}^{\text{sym}}_\varepsilon(v), \varepsilon)$ that is a homomorphism from this semigroup to its noisy counterpart. These operations are summarized in Table 1.

Our randomized merging technique is crucial in supporting general Boolean operations. Unlike our randomized merge operations, Corollary 4.11 shows no *deterministic* operations $\circ$ and $\bullet$ can define merge operations for both *or* ($\vee$) and *and* ($\wedge$) under the same RR mechanism.

Corollary 4.11. *Assume the setting of Theorem 4.1. Let* $\bullet : \{0,1\}^2 \to \{0,1\}$ *denote a deterministic and symmetric operation. It is impossible to satisfy conditions (1)–(3) of Theorem 4.1 in addition to the following:*

4. $f_1(x) \bullet f_2(y) \overset{D}{=} f_3(x \wedge y)$.

5. Cardinality Estimation from Sketches

The Sketch-Flip-Merge method developed so far satisfies privacy and mergeability requirements, but it remains to show how SFM summaries may be used to estimate cardinalities. We develop a composite likelihood–based estimator that is consistent and asymptotically optimal. We give an analytic estimator of the error that closely matches the true error in our experiments.

Likelihood-based approaches to cardinality estimation have been used in the non-private setting (Clifford & Cosma, 2012; Lang, 2017; Ertl, 2017; Ting, 2019), where they have demonstrated greater accuracy than competing estimators for PCSA. While true maximum likelihood estimation of n given a sketch is computationally infeasible due to non-

independence of bits in the sketch (Ting, 2019; Ertl, 2017), the marginal likelihood for any bit is easy to derive. Similar to Ting (2019), we derive a composite marginal likelihood estimator (Lindsay, 1988; Varin et al., 2011) for n.

Let C_{ij} denote the number of unique items $x \in D$ mapped to bucket i and value j in the sketch and ρ_{ij} be the probability an item is mapped to that location. Then, assuming the use of universal random hashes, the following generative process describes the SFM summary $T = \mathcal{M}_{p,q}(\mathcal{S}(D))$.

$$(C_{11}, \ldots, C_{BP}) \sim \text{Multinomial}(n, \rho_{11}, \ldots, \rho_{BP})$$
$$T_{ij} \mid C_{ij} \sim \mathcal{F}_{p,q}(1(C_{ij} > 0))$$

While the joint distribution of the C_{ij}'s, and hence the observed T_{ij}'s, involves an intractable sum over integer partitions of n into at most BP parts, the marginal distribution of a single bit T_{ij} is easy to compute. Note that cell C_{ij}'s probability of occupancy ρ_{ij} depends only on its level j, with $\rho_{ij} = 2^{-\min\{j, P-1\}}/B$. Then $\Pr(C_{ij} = 0 \mid n) = \gamma_j^n$, where $\gamma_j = 1 - \rho_{ij}$, and

$$T_{ij} \sim \text{Bernoulli}\left(p(1-\gamma_j^n) + q\gamma_j^n\right).$$

The composite marginal log-likelihood (Lindsay, 1988; Varin et al., 2011) replaces the true log-likelihood by the surrogate $\ell_{p,q}(n; t)$ that sums over marginal log-probabilities,

$$\ell_{p,q}(n;t) = \textstyle\sum_{ij}(1-t_{ij}) \log\left(1 - p + (p-q)\gamma_j^n\right) + \sum_{ij} t_{ij} \log\left(p - (p-q)\gamma_j^n\right),$$

where $t = \mathcal{M}_{p,q}(\mathcal{S}(D))$ denotes a realized SFM summary. The corresponding composite maximum likelihood estimator is $\hat{n} = \max_n \ell_{p,q}(n; t)$ and can be optimized by Newton's method. The required derivatives of $\ell_{p,q}$ are

$$\ell'_{p,q}(n;t) = \textstyle\sum_{ij}(1-t_{ij}) \frac{(p-q)\gamma_j^n \log(\gamma_j)}{1-p+(p-q)\gamma_j^n} - \sum_{ij} t_{ij} \frac{(p-q)\gamma_j^n \log(\gamma_j)}{p-(p-q)\gamma_j^n},$$
$$\ell''_{p,q}(n;t) = \textstyle\sum_{ij}(1-t_{ij}) \frac{(1-p)(p-q)(\log\gamma_j)^2\gamma_j^n}{(1-p+(p-q)\gamma_j^n)^2} - \sum_{ij} t_{ij} \frac{p(p-q)(\log\gamma_j)^2\gamma_j^n}{(p-(p-q)\gamma_j^n)^2}.$$

In the absence of privacy (i.e., $p = 1, q = 0$), $\ell_{p,q}(\cdot; t)$ is strictly concave. While this is not true in the private case, Theorem 5.1 states that the expectation of $\ell_{p,q}$ remains concave over the interval $(0, n + \Delta)$ for some $\Delta > 0$.

Theorem 5.1. *Let D be a multiset such that $|\texttt{set}(D)| = n$. Let $T = \mathcal{M}_{p,q}(\mathcal{S}(D))$. Let $f(\hat{n}) = \mathbb{E}[\ell_{p,q}(\hat{n}; T)]$, where the expectation is taken over the randomness of the hash functions h_1, h_2 and the privacy mechanism $\mathcal{M}_{p,q}$. Then $f(\hat{n})$ attains a global maximum at $\hat{n} = n$ and is concave on an interval containing $(0, n]$ in its interior.*

5.1. Theoretical Results

We choose to use composite marginal likelihood due to its attractive theoretical properties. In particular, the use of a true likelihood, even if incomplete, ensures that cardinality estimates are asymptotically consistent, and the Hessian of the composite likelihood provides an estimate of the variance (Ting, 2019). We further show the cardinality estimates are asymptotically optimal in the typical case when the cardinality is large relative to the sketch size.

Theorem 5.2. *Let S_n denote a PCSA summary of n distinct items with $B(n)$ buckets and $P = \infty$ levels. Let $\overline{S}_n$ denote a modified PCSA summary of $\text{Poisson}(n)$ distinct items, and $\tilde{S}_n$ denote one where the composite marginal likelihood is the true likelihood. If $B \log B = o(\sqrt{n})$, then there exists modified PCSA summaries $\tilde{S}_n, \overline{S}_n$ where*

$$\Pr(S_n = \tilde{S}_n = \overline{S}_n) \to 1 \quad \textit{as } n \to \infty.$$

Corollary 5.3. *The composite likelihood estimator of the SFM sketch's cardinality is asymptotically efficient in the asymptotic regime in Theorem 5.2.*

We outline the proofs and provide details in Section A.5 of Appendix A. The main difficulty is that entries in an SFM summary are dependent, since each item can only be allocated to one cell. By constructing a coupling between sketches $S_n, \overline{S}_n$ with dependent and independent entries, we show they are asymptotically equal. In these coupled processes, the bucket with maximum difference in item allocations can only differ by only a small amount, $O_p(\sqrt{n}/B \log B)$. By showing an item updates its bucket's sketch values with probability $O(1/v_i)$ (where v_i is the number of items in bucket i), we conclude the coupled sketches are, in fact, equal with probability going to 1 when the average bucket allocation $v_i \approx n/B$ grows fast enough to make the small differences in item allocation irrelevant. Since $\overline{S}_n, \tilde{S}_n$ have independent bits, we couple them via the inverse CDF method and directly bound the probability that they differ. Since the sketches are the same asymptotically, applying the exact same RR noise to them implies their privatized versions are the same, and any estimator on them has the same asymptotic sampling distribution. Therefore, the asymptotically efficient MLE for the independent-entry sketch, i.e., the composite likelihood estimator, is also asymptotically efficient for the true SFM summary S_n.

Remark 5.4. This result also proves the MLE derived under the approximation that each bin has $\text{Poisson}(n/B)$ items is asymptotically efficient. This can be extended to HyperLogLog and other sketches to show pseudo-likelihood based estimators (Ertl, 2017; Ting, 2019) that have good empirical properties are, in fact, asymptotically optimal.

Error Estimation. Like the Fisher information matrix for MLE's, the inverse Godambe (or sandwich) information

provides a consistent estimate of the estimator's variance. The Godambe information is $G(n) = H(n)J^{-1}(n)H(n)$ where $-J(n)$ is the Hessian of the expected log composite likelihood at n and $H(n) = \mathrm{Var}(\ell'_{p,q}(n, T))$ is the variance of the composite score functions. In the non-private case, Ting (2019) demonstrated that composite marginal likelihood variance estimates for HyperLogLog based on Fisher information and Godambe information are nearly identical for large cardinalities and that the Fisher information overestimates the variance at small cardinalities due to the negative dependence of buckets. Figure 1b shows this overestimation is much less pronounced when independent randomized response noise is added. Thus, we use only the Hessian to define the *estimated standard error* as

$$\begin{aligned}\widehat{\mathrm{SE}}_{p,q}(B, P, n) &= \sqrt{1/\mathbb{E}[-\ell''_{p,q}(n; T)]} \qquad (2)\\ &= \Big[B(p-q)\textstyle\sum_{j=1}^{P}(\log\gamma_j)^2\gamma_j^n \\ &\quad \times \left(\tfrac{p}{p-(p-q)\gamma_j^n} - \tfrac{1-p}{1-p+(p-q)\gamma_j^n}\right)\Big]^{-1/2},\end{aligned}$$

where $T = \mathcal{M}_{p,q}(S)$ for a random PCSA sketch S of size $B \times P$ and cardinality n. Figure 2 in Section 6 demonstrates empirically that our error estimates $\widehat{\mathrm{SE}}_{p,q}(B, P, n)$ are a good approximation for the error.

6. Evaluation

We evaluate our methods on both real-world and synthetic datasets. We demonstrate empirically that the SFM summary is the first mergeable ε-DP distinct counting sketch with practical performance, since the errors for the Pagh & Stausholm (2021) sketch are impractically large. Among private sketches, our novel randomized-merge sketch construction dominates the deterministic-merge sketches. Thus, our improvements on the construction, estimation, and privacy analysis yield practical gains. Moreover, our theoretical error closely approximates empirical error.

6.1. Experiment Setup

We consider four different private distinct counting sketches in our experiments. Among our methods, *SFM (sym)* pairs $\mathcal{M}_\varepsilon^{\mathrm{sym}}$ with our randomized merge procedure, while *SFM (xor)* pairs $\mathcal{M}_\varepsilon^{\mathrm{xor}}$ with the deterministic *xor* merge. Both SFM methods use the estimator of Section 5. We compare these methods against the sketch and estimator of Pagh & Stausholm (2021) implemented two ways: *PS (loose)* constructs sketches using $\mathcal{M}_\varepsilon^{\mathrm{PS}} = \mathcal{M}_{p,q}$ with $p = 1/2, q = 1/(2+\varepsilon)$ as prescribed in Pagh & Stausholm (2021), while *PS (tight)* uses the tightened $\mathcal{M}_\varepsilon^{\mathrm{xor}}$ (Definition 4.2). By Corollary 4.5, the sketch construction of *PS (tight)* at ε is equivalent to that of *PS (loose)* at $\varepsilon' = 2(e^\varepsilon - 1)$.

We also consider two non-private sketches as baseline comparisons in our final experiment, noting that we should not expect the accuracy of a private sketch to be as strong as a non-private one. In what follows, *FM85* denotes non-private PCSA using the estimator of Flajolet & Martin (1985), and *HLL* denotes HyperLogLog (Flajolet et al., 2007). We compare HLL sketches against PCSA sketches at equal bucket counts, noting that the HLL sketches are smaller per bucket than the corresponding PCSA sketches.

We measure estimation error primarily in the form of *relative root mean squared error (RRMSE)*, defined as

$$\mathrm{RRMSE}(\hat{n}_1, \ldots, \hat{n}_m; n) = \tfrac{1}{n}\sqrt{\tfrac{1}{m}\textstyle\sum_{i=1}^{m}(\hat{n}_i - n)^2}.$$

We also measure the *relative efficiency* of two methods as the ratio of their mean squared error,

$$\mathrm{RE}(\hat{n}^{(1)}, \hat{n}^{(2)}) = \mathrm{MSE}(\hat{n}^{(2)}) \,/\, \mathrm{MSE}(\hat{n}^{(1)}).$$

If two sketches have unbiased estimators, a relative efficiency of r indicates that the less efficient sketch must asymptotically use r times more buckets to get the same accuracy. This is because the asymptotic MSE (variance) decreases proportionally to $1/B$ for these estimators.

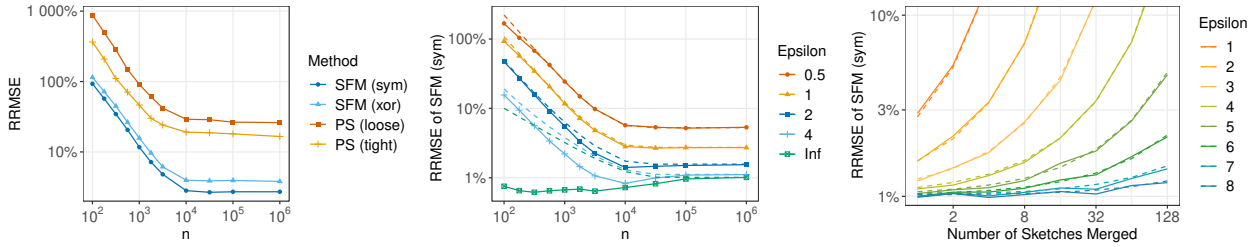
The simulations use sketches with dimensions $B = 4096$, $P = 24$ by default, using the xxHash64 (Collet, 2022) hash function, averaged over $m = 1000$ trials.

Modification to Pagh & Stausholm (2021). In our experiments, their original estimator frequently failed to produce an estimate. For a desired error tolerance β, the method computes intervals for all P levels of the sketch and intersects them to produce an estimate. This intersection was frequently empty for small β. To patch this, we search for the smallest β resulting in a non-trivial intersection. We use the midpoint of this interval to estimate n.

Data Sources. Our experiments use both synthetic and real data. Synthetic data consist of random sets of integers with a fixed cardinality. Real data is taken from the BitcoinHeist paper (Akcora et al., 2020), which provides a database of Bitcoin transactions to $n = 2{,}631{,}095$ unique addresses. The dataset is available in the UCI repository under the CC BY 4.0 license. We note that distinct-count sketches are insensitive to the value distribution of inputs since values are hashed as part of the processing.

6.2. Results

Figure 1a compares the accuracy of the four private methods on synthetic data as the cardinality n ranges from 10^2 to 10^6 given a fixed privacy budget of $\varepsilon = 1$. For large cardinalities, RRMSE tends toward a fixed constant for each method. Our methods have error that is an order of magnitude better than Pagh & Stausholm (2021) even after we tighten their privacy analysis. For small cardinalities, the relative error increases as the cardinality decreases, which is expected



(a) RRMSE at $\varepsilon = 1$ (SFM vs. PS) (b) RRMSE at various ε (SFM) (c) RRMSE at $n = 10^6$ (SFM)

Figure 1. (a) RRMSE vs. n compared across the four private methods. (b) RRMSE for *SFM (sym)* at various ε. (c) RRMSE after merging a given number of SFM summaries, each with a given privacy budget ε. Across plots, dashed lines show estimated relative error, $\widehat{\mathrm{SE}}/n$.

for differentially private methods. Figure 1b compares the accuracy for multiple values of ε but only for the best sketch, *SFM (sym)*. It also shows that the RRMSE stabilizes as $n \to \infty$ regardless of the choice of ε. In contrast to DP methods, the SFM summary with infinite privacy budget, which is the same as PCSA, yields especially accurate estimates at small n. Figure 1b further shows that the estimated relative error $\widehat{\mathrm{SE}}/n$ (Eq. 2) is an upper bound on the empirical error and yields a good estimate of RRMSE, especially for large cardinalities or small ε.

Figure 1c demonstrates the tradeoff between merging and privacy in SFM summaries at large cardinality ($n = 10^6$). Merge operations result in an accumulation of noise, requiring the use of larger privacy budgets to accommodate greater merge counts. The estimated relative error here is calculated according to Remark 4.9 and once again closely matches empirical error.

We also compare private methods against the real-world BitcoinHeist data over a variety of privacy budgets ε, ranging from 0.25 to 4. The left panel of Figure 2 shows the relative efficiency of *SFM (sym)* as compared with the other private methods. *SFM (sym)* is uniformly more efficient than the PS estimators by at least an order of magnitude. The $100\times$ better efficiency compared to *PS (loose)* implies *SFM (sym)* would require just 1% of the space to achieve the same error. Moreover, *SFM (sym)* outperforms *SFM (xor)* with relative efficiency tending toward 4 for larger ε, indicating that for larger privacy budgets, the randomized-merge *SFM (sym)* can achieve comparable accuracy to *SFM (xor)* in as little as one fourth the space. The estimation error from this experiment is depicted in absolute terms in the right panel, where we again see that the estimated relative error for SFM is a good approximation for RRMSE.

Surprisingly, Figure 2 shows that for larger privacy budgets, *PS (loose)* outperforms *PS (tight)*, with RRMSE of *PS (tight)* actually increasing for large ε. This reflects a trend we observed that the PS estimator performs poorly in low-noise regimes, frequently failing to return estimates except at high error tolerance. By contrast, the SFM estimator thrives even

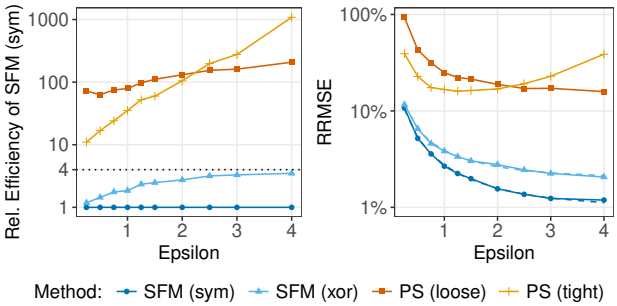


Figure 2. (left) Relative efficiency of *SFM (sym)* vs. other methods on BitcoinHeist ($n \approx 2.6\mathrm{M}$). SFM is far more efficient than PS. *SFM (sym)* is 4x more efficient than *SFM (xor)* for larger privacy budgets. *(right)* RRMSE vs. ε. For SFM, dashed lines show estimated relative error match the true error.

in the complete absence of noise (Figure 1b).

Finally, we compare SFM to popular non-private alternatives and show that error similarly decreases with the bucket count. Using synthetic data with cardinality $n = 10^6$, we construct sketches of varying bucket count B, using a privacy budget of $\varepsilon = 2$ for SFM. Figure 3 shows RRMSE as a function of B for each method. Like the familiar non-private distinct counting sketches, our RRMSE decreases with $B^{-0.5}$. Thus, like non-private sketches, the RRMSE of our DP summaries can be easily characterized by a simple formula $c_\varepsilon/\sqrt{B}$ at large cardinalities, where c_ε is a constant specific to a method and privacy budget.

7. Discussion and Conclusion

The Sketch-Flip-Merge summaries demonstrate dramatic improvement over the current state-of-the-art mergeable and differentially private distinct-count sketches. This is achieved through novel merge algorithms (Theorem 4.8 and Section 4.3), asymptotically optimal estimation (Section 5), and an improved privacy analysis (Corollary 4.5).

An important limitation in mergeable private summaries is the inherent tension between privacy and mergeability.

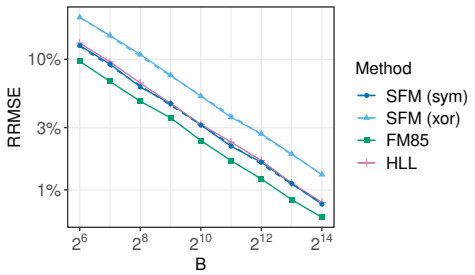


Figure 3. RRMSE at $n = 10^6$ vs. bucket count B for private sketches at $\varepsilon = 2$ vs. common non-private alternatives (on log–log axes). For SFM, dashed lines show estimated relative error. For all methods, RRMSE scales with $B^{-0.5}$.

While both are attainable, repeated merging in the private setting degrades accuracy. This tradeoff, argued in the general distinct-count setting by Desfontaines et al. (2019), is explicitly quantified for SFM in Remark 4.9 and Figure 1c.

Finally, we note the generality of some of our findings. In particular, our methods for aggregating noisy binary data provide fundamental machinery and a quantification of the noise-compounding effects of bitwise operations under randomized response that apply to a wide array of problems, particularly in the privacy-preserving space.

Acknowledgements. We thank Chen-Kuei Lee for helpful conversations, support, and guidance during this project. This work is supported in part by the UKRI Prosperity Partnership Scheme (FAIR) under the EPSRC Grant EP/V056883/1, and the Alan Turing Institute.

References

Akcora, C. G., Li, Y., Gel, Y. R., and Kantarcioglu, M. Bitcoinheist: Topological data analysis for ransomware prediction on the bitcoin blockchain. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 4439–4445, 7 2020. Special Track on AI in FinTech.

Alaggan, M., Cunche, M., and Gambs, S. Privacy-preserving wi-fi analytics. *Proceedings on Privacy Enhancing Technologies*, 2018(2):4–26, 2018.

Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D., and Trevisan, L. Counting distinct elements in a data stream. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pp. 1–10. Springer, 2002.

Broder, A. and Mitzenmacher, M. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4): 485–509, 2004.

Choi, S. G., Dachman-Soled, D., Kulkarni, M., and Yerukhimovich, A. Differentially-private multi-party sketching for large-scale statistics. *Cryptology ePrint Archive*, 2020.

Clifford, P. and Cosma, I. A. A statistical analysis of probabilistic counting algorithms. *Scandinavian Journal of Statistics*, 39(1):1–14, 2012.

Collet, Y. xxHash - extremely fast non-cryptographic hash algorithm. https://cyan4973.github.io/xxHash/, 2022. Accessed: 2022-11-16.

Cormode, G. and Yi, K. *Small summaries for big data*. Cambridge University Press, 2020.

Desfontaines, D., Lochbihler, A., and Basin, D. Cardinality estimators do not preserve privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(2):26–46, apr 2019. doi: 10.2478/popets-2019-0018.

Dickens, C., Thaler, J., and Ting, D. Order-invariant cardinality estimators are differentially private. In *Advances in Neural Information Processing Systems*, 2022.

Durand, M. and Flajolet, P. Loglog counting of large cardinalities. In *European Symposium on Algorithms*, pp. 605–617. Springer, 2003.

Dwork, C. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pp. 1–19. Springer, 2008.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Erlingsson, Ú., Pihur, V., and Korolova, A. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067. ACM, 2014. doi: 10.1145/2660267.2660348. URL https://doi.org/10.1145/2660267.2660348.

Ertl, O. New cardinality estimation algorithms for hyperloglog sketches. *arXiv preprint arXiv:1702.01284*, 2017.

Flajolet, P. and Martin, G. N. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31(2):182–209, 1985.

Flajolet, P., Fusy, É., Gandouet, O., and Meunier, F. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science*, pp. 137–156. Discrete Mathematics and Theoretical Computer Science, 2007.

Giroire, F. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2): 406–427, 2009.

Heule, S., Nunkesser, M., and Hall, A. Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings of the 16th International Conference on Extending Database Technology*, pp. 683–692, 2013.

Kolchin, V., Sevast'janov, B., Christ'Yakov, V., Čistjakov, V., and Balakrishnan, A. *Random Allocations*. A Halsted Press book. V. H. Winston, 1978. ISBN 9780470993941. URL https://books.google.com/books?id=jxjvAAAAMAAJ.

Kreuter, B., Wright, C. W., Skvortsov, E. S., Mirisola, R., and Wang, Y. Privacy-preserving secure cardinality and frequency estimation. 2020.

Lang, K. J. Back to the future: an even more nearly optimal cardinality estimation algorithm. *arXiv preprint arXiv:1708.06839*, 2017.

Lindsay, B. G. Composite likelihood methods. *Contemporary mathematics*, 80(1):221–239, 1988.

Mir, D., Muthukrishnan, S., Nikolov, A., and Wright, R. N. Pan-private algorithms via statistics on sketches. In *Principles of database systems*, 2011.

Nuñez von Voigt, S. and Tschorsch, F. RRTxFM: Probabilistic counting for differentially private statistics. In *Conference on e-Business, e-Services and e-Society*, pp. 86–98. Springer, 2019.

Pagh, R. and Stausholm, N. M. Efficient differentially private F_0 linear sketching. In *24th International Conference on Database Theory (ICDT 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Smith, A., Song, S., and Guha Thakurta, A. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. *Advances in Neural Information Processing Systems*, 33:19561–19572, 2020.

Stanojevic, R., Nabeel, M., and Yu, T. Distributed cardinality estimation of set operations with differential privacy. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*, pp. 37–48. IEEE, 2017.

Ting, D. Approximate distinct counts for billions of datasets. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 69–86, 2019.

Tschorsch, F. and Scheuermann, B. An algorithm for privacy-preserving distributed user statistics. *Computer Networks*, 57(14):2775–2787, 2013.

Varin, C., Reid, N., and Firth, D. An overview of composite likelihood methods. *Statistica Sinica*, pp. 5–42, 2011.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Whang, K.-Y., Vander-Zanden, B. T., and Taylor, H. M. A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems (TODS)*, 15(2):208–229, 1990.

A. Proofs of Main Results

A.1. Additional Notation

In the proofs to follow, we use the following notation. $\mathbf{1}_k$ is the column vector of k ones, and I_k is the $k \times k$ identity matrix (with subscripts omitted when dimensions are clear). We denote by e_i the i-th elementary basis vector, i.e., the vector whose entries are all 0, except at position i, where the entry is 1. We use $\langle \cdot, \cdot \rangle$ to denote the inner product between two vectors.

A.2. Proofs of Results in Section 3

Proof of Theorem 3.2. The proof follows along the same lines as Erlingsson et al. (2014, Theorem 2). We first prove that $\mathcal{F}_{p,q}$ is ε-DP if and only if $p \in (0,1)$ and $\max\{p/q, (1-q)/(1-p)\} \leq e^{\varepsilon}$. For this to hold, we must satisfy

$$\Pr(\mathcal{F}_{p,q}(x) = y) \leq e^{\varepsilon} \Pr(\mathcal{F}_{p,q}(1-x) = y)$$

for all $x, y \in \{0,1\}$. If $p = q$, $\mathcal{F}_{p,p}$ is a mechanism that ignores its inputs and outputs a randomly chosen bit value and so this holds trivially. Assuming $p \neq q$, this holds if

$$\Pr(\mathcal{F}_{p,q}(1-x) = y) > 0 \quad \text{and} \quad \frac{\Pr(\mathcal{F}_{p,q}(x) = y)}{\Pr(\mathcal{F}_{p,q}(1-x) = y)} \leq e^{\varepsilon}.$$

The first condition is equivalent to the requirement that $p, q \in (0,1)$, and the second condition is equivalent to $\max\{p/q, (1-q)/(1-p)\} \leq e^{\varepsilon}$.

For $\mathcal{M}_{p,q}$, note that the entries of $\mathcal{M}_{p,q}$ are independent by construction, so for $x, x' \in \{0,1\}^d$ differing only on one bit $x_j = 1 - x'_j$ and some $y \in \{0,1\}^d$, we have:

$$\frac{\Pr(\mathcal{M}_{p,q}(x) = y)}{\Pr(\mathcal{M}_{p,q}(x') = y)} = \frac{\prod_i \Pr(\mathcal{F}_{p,q}(x_i) = y_i)}{\prod_i \Pr(\mathcal{F}_{p,q}(x'_i) = y_i)} = \frac{\Pr(\mathcal{F}_{p,q}(x_j) = y_j)}{\Pr(\mathcal{F}_{p,q}(1-x_j) = y_j)}.$$

As shown above, this quantity is bounded by e^{ε} under the stated conditions on p and q. □

Theorem 3.2 is used below to demonstrate that both $\mathcal{M}_{\varepsilon}^{\text{xor}}$ and $\mathcal{M}_{\varepsilon}^{\text{sym}}$ satisfy ε-DP via Corollaries 4.3 and 4.7.

A.3. Proofs of Results in Section 4.1

Before proving Theorem 4.1, we provide a helpful result:

Fact A.1. *Let $X \sim \text{Bernoulli}(p), Y \sim \text{Bernoulli}(q)$ be independent. Then:*

1. $X \wedge Y \sim \text{Bernoulli}(pq)$.
2. $X \vee Y \sim \text{Bernoulli}(p(1-q) + q(1-p) + pq)$.
3. $X \veebar Y \sim \text{Bernoulli}(p(1-q) + q(1-p))$. *Moreover, if $p = \frac{1}{2}$ or $q = \frac{1}{2}$, then $X \veebar Y \sim \text{Bernoulli}(\frac{1}{2})$.*

Proof.

$$\begin{aligned}
\Pr(X \wedge Y = 1) &= \Pr(X = 1, Y = 1) \\
&= pq, \\
\Pr(X \veebar Y = 1) &= \Pr(X = 1, Y = 0) + \Pr(X = 0, Y = 1) \\
&= p(1-q) + q(1-p), \\
\Pr(X \vee Y = 1) &= \Pr(X \wedge Y = 1) + \Pr(X \veebar Y = 1).
\end{aligned}$$

□

Proof of Theorem 4.1. We begin with some simple necessary conditions for (1)–(3) to hold. From Theorem 3.2, we know that p_1, p_2, q_1, q_2 must lie in $(0,1)$. Another necessary condition is that $p_1 \neq q_1$ and $p_2 \neq q_2$, as otherwise (3) is violated. For this reason, we can assume $p_1 \neq q_1$ and $p_2 \neq q_2$.

Modulo negation, there exist four symmetric operations $\{0,1\}^2 \to \{0,1\}$: and ($\wedge$), or ($\vee$), xor ($\veebar$), and the trivial operator that maps all inputs to 0. We will now rule out the operators other than $\veebar$.

(Trivial Operator) Let $\circ$ denote the operator $x \circ y = 0$. Then $f_1(x) \circ f_2(y) = 0$ for any x, y. If (2) holds, then $f_3(0) = f_3(1) = 0$, violating (3).

(And) Let $\circ = \wedge$, and assume (2) holds. Then we must have:

$$f_1(0) \wedge f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \wedge f_2(1).$$

By Fact A.1, this implies that:

$$q_1 p_2 = p_1 p_2.$$

Since we assumed $p_2 \neq 0$, this implies that $p_1 = q_1$, in contradiction to our assumption.

(Or) Let $\circ = \vee$, and assume (2) holds. Then we must have:

$$f_1(0) \vee f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \vee f_2(1).$$

By Fact A.1, this implies that:

$$p_1(1-q_2) + q_2(1-p_1) + p_1 q_2 = p_1(1-p_2) + p_2(1-p_1) + p_1 p_2.$$

After rearranging terms, we obtain:

$$0 = (1-p_1)(p_2 - q_2).$$

Since we assumed $p_1 \neq 1$, this implies that $p_2 = q_2$, in contradiction to our assumption.

(Xor) Now we will show that when $\circ = \veebar$, we must have $p_1 = p_2 = \frac{1}{2}$. Assuming condition (2) holds, we have:

$$f_1(0) \veebar f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \veebar f_2(1),$$

which implies (by Fact A.1):

$$q_1(1-p_2) + p_2(1-q_1) = p_1(1-p_2) + p_2(1-p_1).$$

Rearranging terms yields:

$$p_2(p_1 - q_1) = (1-p_2)(p_1 - q_1).$$

Since we assumed $p_1 \neq q_1$, we must have $p_2 = \frac{1}{2}$. A similar argument shows $p_1 = \frac{1}{2}$. □

Proof of Lemma 4.3. We need to show that $p/q \leq e^{\varepsilon}$ and $(1-q)/(1-p) \leq e^{\varepsilon}$ for $p = 1/2$ and $q = 1/(2e^{\varepsilon})$. Clearly, $p/q = e^{\varepsilon}$, satisfying the first component. Next, consider the expression

$$\begin{aligned} e^{\varepsilon} - \frac{1-q}{1-p} &= e^{\varepsilon} - \frac{1 - \frac{1}{2}e^{-\varepsilon}}{\frac{1}{2}} \\ &= e^{\varepsilon} - (2 - e^{-\varepsilon}) \\ &= e^{\varepsilon} + e^{-\varepsilon} - 2 \\ &= (e^{\varepsilon/2} - e^{-\varepsilon/2})^2 \\ &\geq 0. \end{aligned}$$

Therefore, $(1-q)/(1-p) \leq e^{\varepsilon}$ as required. □

Proof of Theorem 4.4. Since the entries of $\mathcal{M}^{\text{xor}}(\cdot)$ are independent, it suffices to show this holds for $\mathcal{M}^{\text{xor}}$ applied to arbitrary $x_i, y_i \in \{0,1\}$. Observe that if $x_i = 1$ or $y_i = 1$, then $x_i \vee y_i = 1$, and so $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2})$. On the other hand, since we know $x_i = 1$ or $y_i = 1$, then $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \sim \text{Bernoulli}(\frac{1}{2})$ or $\mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$, and so by Fact A.1, $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$.

Thus all that remains to show is that $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x \vee y) \overset{D}{=} \mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i)$ when $x_i = y_i = 0$. In this case, $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_1})$, and $\mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_2})$. By Fact A.1, we have that:

$$\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(q^*),$$

where

$$\begin{aligned} q^* &= \frac{1}{2}e^{-\varepsilon_1}\left(1 - \frac{1}{2}e^{-\varepsilon_2}\right) + \frac{1}{2}e^{-\varepsilon_2}\left(1 - \frac{1}{2}e^{-\varepsilon_1}\right) \\ &= \frac{1}{2}\left(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)}\right) \\ &= \frac{1}{2}\exp\left(-\left[\underbrace{-\log\left(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)}\right)}_{\varepsilon^*}\right]\right) \\ &= \frac{1}{2}e^{-\varepsilon^*}. \end{aligned}$$

Finally, since $x_i \vee y_i = 0$, we have that $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon^*})$. □

A.4. Proofs of Results in Section 4.2

Proof of Lemma 4.7. Since $p = e^{\varepsilon}/(e^{\varepsilon}+1)$ and $q = 1 - p = 1/(e^{\varepsilon}+1)$, we have $p/q = (1-q)/(1-p) = e^{\varepsilon}$, and so the result follows immediately from Theorem 3.2. □

We now state a more general form of Theorem 4.8 for proof. Where Theorem 4.8 gives a randomized merge for 2 bits, which may be invoked repeatedly to merge $k > 2$ bits, Theorem A.2 considers a simultaneous merge of $k \geq 2$ bits. Beyond simply serving to prove the original pairwise theorem, this generalization shows that nothing is gained in ε^* (i.e., the noise level of the final sketch) by simultaneously merging k bits vs. performing repeated pairwise merges.

Theorem A.2. *Fix an integer $k \geq 2$. For $i \in [k]$, assume $\varepsilon_i > 0$. Let*

$$q(\varepsilon) = \frac{1}{e^{\varepsilon}+1},\ K_{\varepsilon} = \begin{bmatrix} 1-q(\varepsilon) & q(\varepsilon) \\ q(\varepsilon) & 1-q(\varepsilon) \end{bmatrix},$$

$$\varepsilon^* = -\log\left(1 - \prod_{i=1}^{k}(1 - e^{-\varepsilon_i})\right),\ q^* = q(\varepsilon^*),$$

and let $v^* \in \mathbb{R}^{2^k}$ be the vector whose first entry is q^* with all other entries $1 - q^*$. Let

$$t = (t_{\ldots 01}, t_{\ldots 10}, t_{\ldots 11}, \ldots)^T = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})v^*,$$

$$g(x_1, \ldots, x_k) \sim \text{Bernoulli}(t_{x_1 \ldots x_k}).$$

Then $g(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x_1), \ldots, \mathcal{M}^{\text{sym}}_{\varepsilon_k}(x_k)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x_1 \vee \cdots \vee x_k)$.

Before proving Theorem A.2, we provide another fact that will be used in the proof.

Fact A.3. *Let $A_{m_A \times n_A}, B_{m_B \times n_B}$ be matrices satisfying $A\mathbf{1} = \mathbf{1}$ and $B\mathbf{1} = \mathbf{1}$. Then $(AB)\mathbf{1} = \mathbf{1}$ and $(A \otimes B)\mathbf{1} = \mathbf{1}$. Additionally, if A^{-1} exists, then $A^{-1}\mathbf{1} = \mathbf{1}$.*

Proof. Since $A\mathbf{1} = \mathbf{1}$ and $B\mathbf{1} = \mathbf{1}$, we must have $(AB)\mathbf{1} = A(B\mathbf{1}) = A\mathbf{1} = \mathbf{1}$.

Next, write $\mathbf{1}_{m_A m_B} = \mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B}$. Then

$$\begin{aligned} (A \otimes B)\mathbf{1}_{m_A m_B} &= (A \otimes B)(\mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B}) \\ &= (A\mathbf{1}_{m_A}) \otimes (B\mathbf{1}_{m_B}) \\ &= \mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B} \\ &= \mathbf{1}_{m_A m_B}. \end{aligned}$$

Finally, in the case when A is invertible, $A^{-1}A = I$, and so we must have $A^{-1}\mathbf{1} = A^{-1}A\mathbf{1} = I\mathbf{1} = \mathbf{1}$. □

Proof of Theorem A.2. Since all operations are performed bitwise and independently, assume without loss of generality that the bit vectors x_i are scalar, i.e., $x_1, \ldots, x_k \in \{0, 1\}$.

The fundamental idea is to model the chain of operations performed on the bits x_i as a Markov chain. This involves three different types of transition probability matrices. The matrices K_ε defined in the theorem statement map the state space of a single bit in $\{0, 1\}$ to another bit in $\{0, 1\}$ via the application of $\mathcal{M}_\varepsilon^{\text{sym}}$. Next, we define K^{or} to be the $2^k \times 2$ matrix mapping k bits in $\{0, 1\}^k$ to a single bit in $\{0, 1\}$ via an *or* operation. Finally, we define K^{merge} to be the $2^k \times 2$ matrix corresponding to our desired merge operation, which maps $\{0, 1\}^k$ to $\{0, 1\}$.

Since the bit-flipping operations of K_ε are performed independently, the matrix $K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k}$ represents the $2^k \times 2^k$ matrix jointly mapping the state space of the original bits $\{x_i\}_{i=1}^k$ to $\{\mathcal{M}_{\varepsilon_i}^{\text{sym}}(x_i)\}_{i=1}^k$. Thus we wish to solve:

$$(K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k}) K_{\varepsilon'}^{\text{merge}} = K^{\text{or}} K_{\varepsilon'}, \tag{3}$$

where ε' is a free parameter that we will fix, and $K_{\varepsilon'}^{\text{merge}}$ is the unknown quantity. We proceed by solving the matrix equation above, finding the maximum ε' for which $K_{\varepsilon'}^{\text{merge}}$ represents a valid transition probability matrix.

Let $q_i = q(\varepsilon_i), q' = q(\varepsilon')$. We note that K_{ε_i} is invertible and write $K_{\varepsilon_i}^{-1}$ as follows:

$$K_{\varepsilon_i}^{-1} = \frac{1}{1 - 2q_i} \begin{bmatrix} 1 - q_i & -q_i \\ -q_i & 1 - q_i \end{bmatrix}.$$

So we may solve our matrix equation (3) by left-multiplication of $(K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k})^{-1}$:

$$K_{\varepsilon'}^{\text{merge}} = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) K^{\text{or}} K_{\varepsilon'}.$$

The first column of $K^{\text{or}} K_{\varepsilon'}$ is equal to

$$w' = (1 - q', q', \ldots, q')^T.$$

It follows from Fact A.3 that $K_{\varepsilon'}^{\text{merge}} \mathbf{1} = \mathbf{1}$ and therefore $K_{\varepsilon'}^{\text{merge}}$ is stochastic if and only if:

$$u' = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})\, w' \in [0, 1]^{2^k}.$$

We may write u'_i, the i-th entry of u', as the inner product of w' with the i-th row of $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})$. We denote by r_i this row vector. Writing $w' = q'\mathbf{1} + (1 - 2q')e_1$, u'_i may be written:

$$\begin{aligned} u'_i &= \langle r_i,\ w' \rangle \\ &= \langle r_i,\ q'\mathbf{1} + (1 - 2q')e_1 \rangle \\ &= q' \langle r_i,\ \mathbf{1} \rangle + (1 - 2q') \langle r_i, e_1 \rangle \\ &= q' + (1 - 2q')(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1}, \end{aligned}$$

where the final equality comes from the fact that $\langle r_i, \mathbf{1} \rangle = 1$ (Fact A.3) and that $\langle r_i, e_1 \rangle$ is equal to the $(i, 1)$-th entry of $K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}$.

Since each u'_i entry must be in $[0, 1]$, each entry defines a constraint on q'. In particular, since these values are affine functions of q', each constraint corresponds to an interval. We can see that $q' = \frac{1}{2}$ is valid for each of these constraints, as $u'_i = \frac{1}{2}$ when $q' = \frac{1}{2}$ for all i. Thus it suffices to find a lower bound for q' using these constraints.

We ask next which values of u'_i are most extreme. For any fixed choice of $q' \in [0, \frac{1}{2}]$, we obtain the largest entry u'_i where $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1}$ is maximized and the smallest where that same value is minimized. In particular:

$$\max_i (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1} = \left(\prod_i (1 - 2q_i)^{-1} \right) \prod_i (1 - q_i),$$

since $1 - q_i > q_i > 0$ for all i. (The constant $\prod_i (1 - 2q_i)^{-1}$ appears in all entries.) Similarly, we can see that:

$$\min_i (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1} = \left(\prod_i (1 - 2q_i)^{-1} \right) (-q_j) \prod_{i \neq j} (1 - q_i),$$

where $j = \arg\max_j q_j$, since this yields the most extreme negative term.

It suffices to constrain the two most extreme entries of u' to $[0, 1]$. The constraints defined by these two entries are:

$$q' + (1 - 2q') \underbrace{\frac{\prod_i (1 - q_i)}{\prod_i (1 - 2q_i)}}_{=:c_1} \leq 1 \iff q' \overset{(a)}{\geq} \frac{c_1 - 1}{2c_1 - 1},$$

and

$$q' + (1 - 2q') \underbrace{(-q_j) \frac{\prod_{i \neq j} (1 - q_i)}{\prod_i (1 - 2q_i)}}_{=:-c_2} \geq 0 \iff q' \overset{(b)}{\geq} \frac{c_2}{1 + 2c_2}.$$

We note that $c_1 > 1$ and $c_2 > 0$. Moreover, comparing c_1 and c_2, we find that $c_1 > c_2 + 1$, as:

$$\begin{aligned}
c_1 - (c_2 + 1) &= \frac{\prod_i (1 - q_i)}{\prod_i (1 - 2q_i)} - \frac{q_j \prod_{i \neq j} (1 - q_i) + \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&= \frac{(1 - q_j) \prod_{i \neq j} (1 - q_i)}{\prod_i (1 - 2q_i)} - \frac{q_j \prod_{i \neq j} (1 - q_i) + \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&= \frac{(1 - 2q_j) \prod_{i \neq j} (1 - q_i) - \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&> 0.
\end{aligned}$$

So it follows that:

$$\begin{aligned}
\frac{c_1 - 1}{2c_1 - 1} - \frac{c_2}{1 + 2c_2} &= \frac{(c_1 - 1)(1 + 2c_2) - c_2(2c_1 - 1)}{(2c_1 - 1)(1 + 2c_2)} \\
&= \frac{c_2 - (c_1 + 1)}{(2c_1 - 1)(1 + 2c_2)} \\
&> 0,
\end{aligned}$$

which indicates that constraint (a) implies constraint (b).

We denote by q^* the minimal q' allowed under constraint (a):

$$q^* = \frac{c_1 - 1}{2c_1 - 1} = \frac{\prod_i \frac{1 - q_i}{1 - 2q_i} - 1}{2 \prod_i \frac{1 - q_i}{1 - 2q_i} - 1}.$$

Putting this in terms of ε, note that $1 - q_i = \frac{e^{\varepsilon_i}}{e^{\varepsilon_i} + 1}$, $1 - 2q_i = \frac{e^{\varepsilon_i} - 1}{e^{\varepsilon_i} + 1}$, so:

$$\frac{1 - q_i}{1 - 2q_1} = \frac{e^{\varepsilon_i}}{e^{\varepsilon_i} - 1} = (1 - e^{-\varepsilon_i})^{-1},$$

and hence:

$$q^* = \frac{\prod_i (1 - e^{-\varepsilon_i})^{-1} - 1}{2 \prod_i (1 - e^{-\varepsilon_i})^{-1} - 1} = \frac{1 - \prod_i (1 - e^{-\varepsilon_i})}{2 - \prod_i (1 - e^{-\varepsilon_i})},$$

which gives a final ε^* of:

$$\begin{aligned}
\varepsilon^* &= q^{-1}(q^*) \\
&= \log\left((q^*)^{-1} - 1\right) \\
&= \log\left(\frac{2 - \prod_i (1 - e^{-\varepsilon_i})}{1 - \prod_i (1 - e^{-\varepsilon_i})} - 1\right) \\
&= \log\left(\frac{1}{1 - \prod_i (1 - e^{-\varepsilon_i})}\right) \\
&= -\log\left(1 - \prod_{i=1}^{k} (1 - e^{-\varepsilon_i})\right).
\end{aligned}$$

Finally, to translate from transition probability matrices back to the theorem statement, note that $K_{\varepsilon^*}^{\text{merge}} = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) K^{\text{or}} K_{\varepsilon^*}$ maps the $\{0,1\}^k$ state space to $\{0,1\}$—i.e., the 2^k possible inputs map to Bernoulli random variables with probabilities taken from the second column of $K_{\varepsilon^*}^{\text{merge}}$. It follows from the preceding discussion that this vector is precisely $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) v^*$. □

A.5. Proofs of Results for Section 5

Proof of Theorem 5.1. Let $f(\hat{n}) = \mathbb{E}[\ell(\hat{n}; T)]$. We will use the notation $\mathbb{E}_n[\cdot]$ to denote expectation under a cardinality of n, while abusing notation with $\mathbb{E}_{\hat{n}}[\cdot]$ to denote the equivalent quantity with $\hat{n}$ replacing n, as below:

$$\begin{aligned}
\mathbb{E}_n[T_{ij}] &= \Pr(T_{ij} = 1) \\
&= p(1 - \gamma_j^n) + q\gamma_j^n \\
&= p - (p-q)\gamma_j^n \\
\mathbb{E}_{\hat{n}}[T_{ij}] &= p - (p-q)\gamma_j^{\hat{n}}.
\end{aligned}$$

(Note that $\mathbb{E}_{\hat{n}}[\cdot]$ is not truly an expectation, since when $\hat{n}$ is non-integer, the distribution of T is not defined.) Observe that:

$$\begin{aligned}
\mathbb{E}_n[f'(\hat{n})] &= B \sum_{j=1}^{P} (1 - \mathbb{E}_n[T_{ij}])(p-q)\gamma_j^{\hat{n}} \log(\gamma_j)(1 - \mathbb{E}_{\hat{n}}[T_{ij}])^{-1} \\
&\quad - B \sum_{j=1}^{P} \mathbb{E}_n[T_{ij}](p-q)\gamma_j^{\hat{n}} \log(\gamma_j)(\mathbb{E}_{\hat{n}}[T_{ij}])^{-1} \\
&= B(p-q) \sum_{j=1}^{P} \phi_j'(\hat{n}),
\end{aligned}$$

where

$$\phi_j'(\hat{n}) = \gamma_j^{\hat{n}} \log(\gamma_j) \left(\frac{1 - \mathbb{E}_n[T_{ij}]}{1 - \mathbb{E}_{\hat{n}}[T_{ij}]} - \frac{\mathbb{E}_n[T_{ij}]}{\mathbb{E}_{\hat{n}}[T_{ij}]} \right).$$

As expected, this equals zero when $\hat{n} = n$. Moreover, it is strictly positive for $\hat{n} < n$ and strictly negative for $\hat{n} > n$. Thus the same properties hold for $f'(\hat{n})$, and so n is the global maximizer of f.

By similar logic, we may write

$$\mathbb{E}_n[f''(\hat{n})] = B(p-q) \sum_{j=1}^{P} \phi_j''(\hat{n}),$$

where

$$\phi_j''(\hat{n}) = (\log \gamma_j)^2 \gamma_j^{\hat{n}} \left((1-p) \frac{1 - \mathbb{E}_n[T_{ij}]}{(1 - \mathbb{E}_{\hat{n}}[T_{ij}])^2} - p \frac{\mathbb{E}_n[T_{ij}]}{(\mathbb{E}_{\hat{n}}[T_{ij}])^2} \right).$$

Although $\phi_j''(\hat{n}) > 0$ for sufficiently large $\hat{n}$, note that the parenthetical quantity is monotonically increasing in $\hat{n}$. Moreover, we know $\phi_j''(n) < 0$ since $\phi_j'(n)$ corresponds to a maximum. Thus the parenthetical (and indeed all of $\phi_j''(\hat{n})$) must be negative for $\hat{n} \le n$. Since this is true for all j, it follows that $f''(\hat{n}) < 0$ for $\hat{n} \le n$. □

Lemma A.4. *Consider a bucket in a PCSA summary with v items where the bucket has $P = \infty$ bits. The probability that a new item allocated to the bucket modifies the bucket is bounded by c/v for all $v > v_0$ for some constants c, v_0.*

Proof. The probability a bucket containing v items is modified by a new item allocated to the bucket is $\sum_{\ell=1}^{\infty} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v$. Split this sum into the ranges $\ell \in \mathcal{I}_1 := (0, \log_2 v - \log_2 \log_2 v)$, $\ell \in \mathcal{I}_2 := [\log_2 v - \log_2 \log_2 v, \log_2 v)$, $\ell \in \mathcal{I}_3 := [\log_2 v, \infty)$. Since $2^{-\ell} \le 1$ and $(1 - 2^{-\ell})^v < \exp(-2^{-\ell} v) \le 1$,

$$\sum_{\ell \in \mathcal{I}_1} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v \le \log_2 v \exp(-\log_2 v) = o(1),$$

$$\sum_{\ell \in \mathcal{I}_2} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v \le \int_{\log_2 v - \log_2 \log_2 v}^{\log_2 v} 2^{-x} \exp(-2^{-x} v) dx$$

$$= \frac{1}{v \log 2} \exp(-v2^{-x})\Big|_{\log_2 v - \log_2 \log_2 v}^{\log_2 v}$$
$$= \frac{e^{-1}}{v \log(2)} - O(\exp(-\log_2 v)/v),$$
$$\sum_{\ell \in \mathcal{I}_3} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v \leq \sum_{\ell \in \mathcal{I}_3} 2^{-\ell} \leq \frac{1}{v}.$$

Summing these components gives the desired result. □

Proof of Theorem 5.2. The modified PCSA summary $\overline{S}_n$ can be generated in the following way. Draw a new cardinality $\overline{N} \sim \text{Poisson}(n)$. The first $\min\{n, \overline{N}\}$ items are shared for the regular SFM summary S_n and modified summary $\overline{S}_n$. Each of these items are allocated to the same bucket for both summaries. Denote the remaining items by $R = |\overline{N} - n|$. The variance of a $\text{Poisson}(n)$ gives $R = O_p(\sqrt{n})$.

Using Theorem 3 in Kolchin et al. (1978) for the asymptotic distribution of the maximum value in a multinomial vector, the bucket allocated the largest number of remaining items has $O_p(R/B \log B) = O_p(\sqrt{n}/B \log B)$ items. Likewise the bucket with the minimum number of items has $n/B + O_p(\sqrt{n}/B \log B)$ items. By Lemma A.4, the probability a new item in a bucket will update the bucket's value is $O(1/V_i)$ given V_i, the number of items already in the bucket. Thus, the probability that no bucket is updated by one of the remaining items is

$$\left(1 - \frac{O(\sqrt{n}/B \log B)}{n/B + O(\sqrt{n}/B \log B)}\right)^B + o(1) = \left(1 - \frac{O(\log B)}{\sqrt{n}}\right)^B + o(1)$$
$$= (1 - o(1/B))^B + o(1) \to 1$$

as $n \to \infty$. This gives that $\Pr(\overline{S}_n = S_n) \to 1$ as $n \to \infty$ and the true PCSA sketch and the Poissonized one are asymptotically equal.

We can also relate the Poissonized PCSA sketch to the one whose true likelihood is the composite marginal likelihood. By Poisson splitting, the entries of $\overline{S}_n$ are independent. Thus, we can couple the entries of $\overline{S}_n$ with those of $\tilde{S}_n$ via the inverse CDF method by using the same underlying $\text{Uniform}(0, 1)$ random variables. The probability that an entry in level j is different across the coupled sketches is

$$\Pr(\tilde{S}_n(i,j) \neq \overline{S}_n(i,j)) = \exp\left(-\frac{n}{B\,2^j}\right) - \left(1 - \frac{1}{B\,2^j}\right)^n$$
$$= \exp\left(-\frac{n}{B\,2^j}\right)\left(1 - \exp\left[n \log\left(1 - \frac{1}{B2^j}\right) + \frac{n}{B2^j}\right]\right)$$
$$< \exp\left(-\frac{n}{B\,2^j}\right)\left(1 - \exp\left[-\frac{n}{B^2 2^{2j+1}}\right]\right)$$
$$< \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B^2 2^{2j+1}}.$$

Applying a union bound gives and splitting the sum at some positive integer k gives

$$\Pr(\tilde{S}_n \neq \overline{S}_n) \leq B \sum_{j=1}^{\infty} \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B^2 2^{2j+1}}$$
$$\leq \sum_{j=1}^{k} \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B} + \sum_{j=k+1}^{\infty} \frac{n}{B2^{2j+1}}$$
$$\leq \exp\left(-\frac{n}{B\,2^k}\right)\frac{nk}{B} + \frac{n}{B2^{2k}}.$$

Take $k = \log_2\left(\frac{n/B}{2\log(n/B)}\right) + \delta$ for some $\delta \in [-1/2, 1/2)$. Then the first part

$$\exp\left(-\frac{n}{B\,2^k}\right)\frac{nk}{B} = \frac{nk}{B}\exp\left(-2^{1-\delta}\log(n/B)\right)$$

$$\leq \frac{nk}{B}(n/B)^{-3/2} \to 0.$$

as $n/B \to \infty$. Likewise, the second part

$$\frac{n}{B2^{2k}} = \frac{n}{B}\left(\frac{n/B}{2\log(n/B)}\right)^{-2} 2^{-2\delta} = \frac{4\log^2(n/B)}{n/B}2^{-2\delta} \to 0$$

as $n/B \to \infty$. Thus $\Pr(\tilde{S}_n = \overline{S}_n) \to 1$ as $n \to \infty$ as well.

□

Proof of Corollary 5.3. Since both a PCSA sketch S_n and modified sketch with independent bins $\tilde{S}_n$ are equal with probability going to 1 as $n \to \infty$, the private SFM sketches T_n, $\tilde{T}_n$ obtained by applying the same randomized response noise to them are also equal with probability going to 1. Let $\hat{\theta}(T_n)$ be some cardinality estimator and $V(\hat{\theta}(T_n))$ denote its asymptotic variance. Then $\min_{\hat{\theta}\in\Theta} V(\hat{\theta}(T_n)) = \min_{\hat{\theta}\in\Theta} V(\hat{\theta}(\tilde{T}_n))$, and a cardinality estimator for T_n is asymptotically efficient if and only if it is asymptotically efficient for $\tilde{T}_n$. Since the composite likelihood estimator for T_n is the true maximum likelihood estimator for $\tilde{T}_n$, it is asymptotically efficient. □

B. Proofs and Results for General Boolean Operations

In the main text, we concern ourselves primarily with merge operations under randomized response that emulate the logical operator *or* ($\vee$). Here we discuss generalizations of these merge operations to other Boolean operations.

B.1. Boolean Operations under Symmetric Randomized Response

Recall that a merge for *and* ($\wedge$) under $\mathcal{M}^{\text{sym}}$ was presented in Corollary 4.10.

Proof of Corollary 4.10. We prove this in the more general setting of merging k bits, as in Theorem A.2. Indeed, proving this is essentially equivalent to Theorem A.2, except that we must replace K^{or} with a transition matrix K^{and} that maps $(x_1, \ldots, x_k)$ to 1 only when $x_1 = \cdots = x_k = 1$. Then for fixed ε' we have a potential solution:

$$K^{\text{merge}}_{\varepsilon'} = (K^{-1}_{\varepsilon_1} \otimes \cdots \otimes K^{-1}_{\varepsilon_k})K^{\text{and}}K_{\varepsilon'}.$$

Once again, we seek the largest ε' for which $K^{\text{merge}}_{\varepsilon'}$ is a valid transition probability matrix. This time, we will use the second column of $K^{\text{and}}K_{\varepsilon'}$ to determine constraints on ε'. (This is allowable since $K^{\text{and}}K_{\varepsilon'}\mathbf{1} = \mathbf{1}$.) We write the second column as:

$$w' = (q', \ldots, q', 1-q')^T = q'\mathbf{1} + (1-2q')e_{2k}.$$

Using w' as in the proof of Theorem A.2, we obtain the same constraints on ε'. Applying the remainder of the proof of Theorem A.2 yields the final result. □

Next, we demonstrate *xor* ($\veebar$) merging under $\mathcal{M}^{\text{sym}}$.

Lemma B.1. $\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x) \veebar \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x \veebar y)$ *for* $\varepsilon^* = \log(1 + e^{\varepsilon_1+\varepsilon_2}) - \log(e^{\varepsilon_1} + e^{\varepsilon_2})$.

Proof. Let $p_1 = e^{\varepsilon_1}/(e^{\varepsilon_1}+1), p_2 = e^{\varepsilon_2}/(e^{\varepsilon_2}+1)$. Using Fact A.1, we have that:

$$\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\theta_{x_i,y_i}),$$

where

$$\theta_{0,0} = \theta_{1,1} = p_1(1-p_2) + p_2(1-p_1)$$
$$\theta_{0,1} = \theta_{1,0} = p_1p_2 + (1-p_1)(1-p_2).$$

Through a bit of algebra, we obtain

$$\begin{aligned}\theta_{0,0} = \theta_{1,1} &= \frac{e^{\varepsilon_1}}{(e^{\varepsilon_1}+1)(e^{\varepsilon_2}+1)} + \frac{e^{\varepsilon_2}}{(e^{\varepsilon_1}+1)(e^{\varepsilon_2}+1)} \\ &= \frac{e^{\varepsilon_1}+e^{\varepsilon_2}}{e^{\varepsilon_1+\varepsilon_2}+e^{\varepsilon_1}+e^{\varepsilon_2}+1} \\ &= \frac{1}{\frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}+1},\end{aligned}$$

while

$$\theta_{1,0} = \theta_{0,1} = 1 - \theta_{0,0}.$$

So we have

$$\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y_i) \overset{D}{=} \mathcal{M}_{\theta_{1,0},1-\theta_{1,0}}(x_i \veebar y_i).$$

Finally, to obtain ε^*, note that $\mathcal{M}_{\theta_{1,0},1-\theta_{1,0}} = \mathcal{M}^{\text{sym}}_{\varepsilon^*}$ for

$$e^{\varepsilon^*} = \frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}.$$

□

As our final step in supporting general Boolean operations under $\mathcal{M}^{\text{sym}}$, we show that the unary operation *not* ($\neg$) commutes with $\mathcal{M}^{\text{sym}}$.

Lemma B.2. *For any bit vector x, we have $\neg(\mathcal{M}_{p,q}(x)) \overset{D}{=} \mathcal{M}_{p,q}(\neg x)$ if and only if $q = 1-p$. In particular, $\neg(\mathcal{M}^{\text{sym}}_{\varepsilon}(x)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon}(\neg x)$.*

Proof. Note that

$$\neg(\mathcal{M}_{p,q}(x_i)) \overset{D}{=} \mathcal{M}_{1-p,1-p}(x_i) \overset{D}{=} \mathcal{M}_{1-q,1-p}(\neg x_i).$$

Consequently,

$$\neg(\mathcal{M}_{p,q}(x)) \overset{D}{=} \mathcal{M}_{p,q}(\neg x) \iff p = 1-q \iff q = 1-p.$$

□

B.2. Boolean Operations with Deterministic Merging

In contrast with the results that leveraged randomized merging, we demonstrate that a deterministic merge for a given Boolean operation requires specific choices of randomized response mechanism, precluding the use of general Boolean operations under a single RR mechanism. In particular, we demonstrate that a deterministic *and* ($\wedge$) merge requires a different privacy mechanism than the *or* ($\vee$) merge described in Theorem 4.1.

Corollary B.3. *Let $f_1 = \mathcal{F}_{p_1,q_1}, f_2 = \mathcal{F}_{p_2,q_2}, f_3 = \mathcal{F}_{p_3,q_3}$, and let $\bullet : \{0,1\}^2 \to \{0,1\}$ denote a deterministic and symmetric operation. The following conditions may only be satisfied simultaneously if $\circ = \veebar$ and $q_1 = q_2 = 1/2$:*

1. *f_1, f_2 are (respectively) ε_1-DP and ε_2-DP for $\varepsilon_1, \varepsilon_2 < \infty$.*
2. *$f_1(x) \bullet f_2(y) \overset{D}{=} f_3(x \wedge y)$.*
3. *$f_i(0) \overset{D}{\neq} f_i(1)$ for $i = 1, 2, 3$.*

Proof. Note that this theorem statement is identical to that of Theorem 4.1 except that condition (2) has changed from using the *or* operation $\vee$ to the *and* operation $\wedge$ and that the necessary condition is no longer on p_1, p_2 but instead q_1, q_2.

Consider the original condition (2) of Theorem 4.1. Since this must hold for all $x, y \in \{0, 1\}$, we may alternatively write this condition in terms of $\neg x$ and $\neg y$ instead:

$$\begin{aligned}
\mathcal{F}_{q_1,p_1}(x) \circ \mathcal{F}_{q_2,p_2}(y) &\overset{D}{=} \mathcal{F}_{p_1,q_1}(\neg x) \circ \mathcal{F}_{p_2,q_2}(\neg y) \\
&\overset{D}{=} f_1(\neg x) \circ f_2(\neg y) \\
&\overset{D}{=} f_3((\neg x) \vee (\neg y)) \\
&\overset{D}{=} f_3(\neg(x \wedge y)) \\
&\overset{D}{=} \mathcal{F}_{q_3,p_3}(x \wedge y).
\end{aligned}$$

From Theorem 4.1, we know that we can only satisfy this condition simultaneously with (1) and (3) if $p_1 = p_2 = 1/2$. Recognizing that this statement is equivalent to condition (2) of the corollary but with the roles of p_i and q_i swapped, it is apparent that to satisfy (1)–(3) of our corollary, we must have $\bullet = \veebar$ and $q_1 = q_2 = 1/2$. □

From here, it follows that no privacy mechanism $\mathcal{M}_{p,q}$ can satisfy the conditions of Theorem 4.1 and Corollary B.3 simultaneously.

Proof of Corollary 4.11. Assume conditions (1)–(4) are satisfied. By Theorem 4.1, we must have $p_1 = p_2 = 1/2$, while Corollary B.3 states that $q_1 = q_2 = 1/2$. This results in a contradiction: $f_1(0) \overset{D}{=} \text{Bernoulli}(1/2) \overset{D}{=} f_1(1)$, violating (3). □