Differentially Private Aggregation via Imperfect Shuffling

Badih Ghazi[*] Ravi Kumar[†] Pasin Manurangsi[‡] Jelani Nelson[§] Samson Zhou[▽]

June 5, 2023

Abstract

In this paper, we introduce the imperfect shuffle differential privacy model, where messages sent from users are shuffled in an *almost* uniform manner before being observed by a curator for private aggregation. We then consider the private summation problem. We show that the standard split-and-mix protocol by Ishai et. al. [FOCS 2006] can be adapted to achieve near-optimal utility bounds in the imperfect shuffle model. Specifically, we show that surprisingly, there is no additional error overhead necessary in the imperfect shuffle model.

1 Introduction

Differential privacy (DP) [DMNS06] has emerged as a popular concept that mathematically quantifies the privacy of statistics-releasing mechanisms. Consequently, DP mechanisms have been recently deployed in industry [Gre16, EPK14, Sha14, DKY17], as well as by government agencies such as the US Census Bureau [Abo18]. DP is parameterized by ε and δ, where ε is a privacy loss parameter that is generally a small positive constant such as 1 and δ is an approximation parameter or "failure" probability that is typically (smaller than) inverse-polynomial in n:

Definition 1.1 (Differential privacy)**.** *[DMNS06, DKM$^+$06] Given $\varepsilon > 0$ and $\delta \in (0, 1)$, a randomized algorithm $\mathcal{A} : X \to Y$ is (ε, δ)-differentially private if, for every neighboring datasets x and x', and for all $S \subseteq Y$,*

$$\mathbf{Pr}\left[\mathcal{A}(x) \in S\right] \le e^{\varepsilon} \cdot \mathbf{Pr}\left[\mathcal{A}(x') \in S\right] + \delta.$$

In this paper, we study the real summation problem, where each of n parties holds a number $x_i \in [0, 1]$ for all $i \in [n]$ and the goal is to privately (approximately) compute $\sum_{i=1}^{n} x_i$. Due to its fundamental nature, the private real summation problem has a wide range of applications, such as private distributed mean estimation [SYKM17, BDKU20], e.g., in federated learning [KMY$^+$16, GDD$^+$21, KMA$^+$21], private stochastic gradient descent [SCS13, BST14, ACG$^+$16, ASY$^+$18, CWH20], databases and information systems [KTH$^+$19, WZL$^+$20], and clustering [SK18, Ste21].

[*] Google Research. E-mail: badihghazi@gmail.com.
[†] Google Research. E-mail: ravi.k53@gmail.com.
[‡] Google Research. E-mail: pasin@google.com.
[§] UC Berkeley and Google Research. E-mail: minilek@alum.mit.edu.
[▽] UC Berkeley and Rice University. E-mail: samsonzhou@gmail.com.

In the central model of DP, where a curator is given full access to the raw data in order to release the private statistic or data structure, the Laplace mechanism [DMNS06] can achieve, for real summation, additive error $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$, which is known to be nearly optimal for $\varepsilon \leq 1$ [GRS12].

However, the ability for the curator to observe the full data is undesirable in many commercial settings, where the users do not want their raw data to be sent to a central curator. To address this shortcoming, the local model of DP [KLN+11, War65] (LDP) demands that all messages sent from an individual user to the curator is private. Unfortunately, although the local model achieves near-minimal trust assumptions, numerous basic tasks provably must suffer from significantly larger estimation errors compared to their counterparts in the central model. For the real summation problem, [BNO08] achieves additive error $\mathcal{O}_\varepsilon(\sqrt{n})$ and it is known that smaller error bounds cannot be achieved [CSS12].

Consequently, the shuffle model [BEM+17, EFM+19, CSU+19] of DP was introduced as an intermediary between the generous central model and the strict local model. In the shuffle model, the messages sent from the users are randomly permuted before being observed by the curator, in an encode-shuffle-analyze architecture. Surprisingly, when users are allowed to send multiple messages, there exist protocols in the shuffle model of DP that achieve additive error $\mathcal{O}_\varepsilon(1)$ for the private real summation problem [GMPV20, BBGN20, GKM+21]. Unfortunately, practical applications can lack the ideal settings that provide the full assumptions required by the shuffle model of DP.

1.1 Model and Motivation

We first define a natural generalization of the uniform shuffler that tolerates imperfections. Let Π be the set of permutations on $[n]$. For $\pi, \pi' \in \Pi$, we define $\mathrm{Swap}(\pi, \pi')$ to be the minimum number of coordinate *swaps*[1] that can be applied to π to obtain π'.

Definition 1.2 (γ-Imperfect Shuffler)**.** *For a distortion parameter $\gamma > 0$, we say that $\mathcal{S}$ is a γ-imperfect shuffler if, for all $\pi, \pi' \in \Pi$,*

$$\mathbf{Pr}\left[\mathcal{S} = \pi\right] \leq e^{\gamma \cdot \mathrm{Swap}(\pi, \pi')} \mathbf{Pr}\left[\mathcal{S} = \pi'\right].$$

We call an output from such a shuffler a γ*-imperfect shuffle* or a γ*-I-shuffle*, for short. Here, γ represents an upper bound on the multiplicative distortion of the output probabilities of the distributions of the shuffler, i.e., how the distribution deviates from a perfectly symmetric shuffler. For example, $\gamma = 0$ corresponds to a perfectly symmetric shuffler while $\gamma \to \infty$ represents almost no guarantee from the shuffler.

To understand the motivation behind this definition, consider a setting where a number of user devices collect statistics to be sent to an intermediate buffer, which is ultimately sent to a central curator for processing. The devices may choose to perform this collection over different periods of time, so that immediately sending their statistics over to the curator could reveal information about their identity, through the timestamp.

For example, consider a setting where sensors are monitoring traffic in US cities during peak afternoon hours. Then reports that are received earlier in the day by the curator are more likely to correspond to cities that are in the east, while reports that are received later in the day by the curator are more likely to correspond to cities in the west. To mitigate this, the sensors instead could choose a universally fixed hour during the day to broadcast their reports from the previous day, at some random time during the hour.

[1] We say that π' results from an application of a coordinate swap on π iff $\pi(i) = \pi'(i)$ on all except two $i \in [n]$.

Specifically, each user $i \in [n]$ could choose a time t_i, say normalized without loss of generality to $t_i \in [0,1]$, and send their messages at time t_i. If the t_i are chosen uniformly at random and this protocol was executed perfectly, it would result in a uniform shuffle of the messages for a buffer that strips both the source information and the exact time of arrival, e.g., [TW23].[2] However, issues may arise such as different clock skews, where users may not perfectly synchronize the fixed hour during which the messages should be sent, or communication delays, either because an intermediate link has failed or simply because latency varies across different networks. It is unclear how to model the imperfect shuffle resulting from these issues using the standard shuffle model.

For a better handle on modeling the imperfection, we can assume that each t_i is adversarially chosen in $[0,1]$. Moreover, each message transmission time can now be altered by a random offset from the intended release time, where the offset is drawn, e.g., from a Laplacian distribution. Specifically, each user $i \in [n]$ draws an offset τ_i from the (centered) Laplacian distribution with scale $\frac{2}{\gamma}$, and sends their message at time $t_i + \tau_i$ instead of at time t_i.

In other words, each user $i \in [n]$ sends their message at time $t_i + \tau_i$, which is determined by the two following quantities:

(1) t_i is an arbitrary and possibly adversarially chosen offset due to nature or some other external source, e.g., clock skews, transmission failure, communication delay.

(2) τ_i is an internal source of noise that the protocol can sample from a predetermined distribution to mitigate the negative privacy effects of t_i.

Note that whereas two neighboring permutations π and π' from the set of permutations Π on $[n]$ were equally likely to be output by the shuffler, this may now no longer be the case. On the other hand, for fixed $i, j \in [n]$ and conditioning on the values of $\{t_1, \tau_1, \ldots, t_n, \tau_n\} \setminus \{t_i, \tau_i, t_j, \tau_j\}$, we can see that for $a, b \in [n]$, the probability that $t_a + \tau_a \le t_i + \tau_i \le t_{a+1} + \tau_{a+1}$ and $t_b + \tau_b \le t_j + \tau_j \le t_{b+1} + \tau_{b+1}$ is within an e^γ factor of the probability that $t_a + \tau_a \le t_j + \tau_j \le t_{a+1} + \tau_{a+1}$ and $t_b + \tau_b \le t_i + \tau_i \le t_{b+1} + \tau_{b+1}$.

Specifically, let $\mathcal{E}_1$ be the event that $\tau_i \in [t_a + \tau_a - t_i, t_{a+1} + \tau_{a+1} - t_i]$, where τ_i is a (centered) Laplace random variable and scale $\frac{2}{\gamma}$. Similarly, let $\mathcal{E}_2$ be the event that $\tau_j \in [t_b+\tau_b-t_j, t_{b+1}+\tau_{b+1}-t_j]$ where τ_j is a (centered) Laplace random variable and scale $\frac{2}{\gamma}$. Furthermore, let $\mathcal{E}_3$ be the event that $\tau_j \in [t_a + \tau_a - t_j, t_{a+1} + \tau_{a+1} - t_j]$ and $\mathcal{E}_4$ be the event that $\tau_i \in [t_b + \tau_b - t_i, t_{b+1} + \tau_{b+1} - t_i]$. Then by the properties of the Laplace distribution and the assumption that $t_i, t_j \in [0,1]$, we have

$$\mathbf{Pr}\left[\mathcal{E}_1 \wedge \mathcal{E}_2\right] = \mathbf{Pr}\left[\mathcal{E}_1\right] \mathbf{Pr}\left[\mathcal{E}_2\right] \le (e^{\gamma/2} \cdot \mathbf{Pr}\left[\mathcal{E}_3\right])(e^{\gamma/2} \cdot \mathbf{Pr}\left[\mathcal{E}_4\right]) = e^\gamma \cdot \mathbf{Pr}\left[\mathcal{E}_3 \wedge \mathcal{E}_4\right].$$

Thus, the resulting distribution over permutations is captured by the γ-I-shuffle model.

We can naturally generalize this setting to the model where each user sends m messages, e.g., m buffers collect messages from n users, which results in times $\{t_{i,j}\}_{i\in[n],j\in[m]}$ and offsets $\{\tau_{i,j}\}_{i\in[n],j\in[m]}$. Formally, for m rounds of messages for the n users, $\{m_{i,j}\}_{i\in[n],j\in[m]}$, a separate permutation π_j drawn from a γ-imperfect shuffler is used to shuffle the messages $\{m_{i,j}\}_{i\in[n]}$, for each $j \in [m]$. For example, $\{m_{i,1}\}_{i\in[n]}$ is shuffled according to a permutation π_1 drawn from a γ-imperfect shuffler, $\{m_{i,2}\}_{i\in[n]}$ is shuffled according to an independent permutation π_2 drawn from the same γ-imperfect shuffler, and so on and so forth.

[2] We assume in this example that the buffer can queue the messages, and then forward them to the analyst at some point of time, but that it cannot further shuffle them. The (imperfect) shuffling we consider stems solely from the randomization of the transmission time of the messages by the users.

We remark that the above model is sometimes referred to as the *m-parallel shuffling* model; another model that has been considered in literature is one where all the mn messages are shuffled together using a single shuffler. We only focus on the former in this paper. It remains an interesting open question whether the results can be extended to the latter model.

1.2 Our Contributions

Surprisingly, we present a protocol for the real summation problem that matches the utility bounds of the best protocols in the shuffle model. Thus, we show that there is no additional error overhead necessary in the γ-I-shuffle model, i.e., there is no utility loss due to the imperfect shuffler.

Theorem 1.3. *Let $n \geq 19$ and $\gamma \leq \frac{\log\log n}{80}$ be a distortion parameter. Then there exists an (ε, δ)-DP protocol for summation in the γ-I-shuffle model with expected absolute error $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ and*

$$m = \mathcal{O}\left(e^{4\gamma} + \frac{e^{4\gamma}(\log\frac{1}{\delta} + \log n)}{\log n}\right)$$

messages per party. Each message uses $\mathcal{O}(\log q)$ bits, for $q = \lceil 2n^{3/2} \rceil$.

Observe that when δ is inverse-polynomial in n and the distortion parameter γ is a constant $\mathcal{O}(1)$, then the number of messages m sent by each player in Theorem 1.3 is a constant. Moreover, under these settings, Theorem 1.3 recovers the guarantees in the standard shuffle model from [BBGN20, GMPV20], though we remark that more communication efficient protocols [GKM$^+$21] are known in the standard shuffle model across more general settings. Regardless, we again emphasize that the privacy and utility guarantees of the protocol are independent of the distortion parameter γ.

1.3 Overview of our Techniques

In this section, we describe both our protocol for private real summation in the γ-I-shuffle model and the corresponding analysis for correctness and privacy.

A natural starting point is the recent framework by [ZS22, ZSCM22], which achieves amplification of privacy using *differentially oblivious* (DO) shufflers that nearly match amplification of privacy results using fully anonymous shufflers [EFM$^+$19, BBGN19a, CSU$^+$19, FMT21]. Unfortunately, the framework crucially uses LDP protocols, which are known to not give optimal bounds even with fully anonymous shufflers. For instance, [BBGN20, CSU$^+$19, BBGN19b] showed that any single-message shuffled protocol for summation based on LDP protocols must exhibit mean squared error $\Omega(n^{1/3})$ or absolute error $\Omega(n^{1/6})$.

Another natural approach is to adapt recent works for private real summation in the shuffle model, e.g., [GMPV20, GKM$^+$21]. One challenge in generalizing these proofs is that they often leverage the fully anonymous shuffler by analyzing a random sample from an alternate view of the output of the local randomizers, which often have some algebraic or combinatoric interpretation that facilitates the proof of specific desirable properties. However, these properties often seem substantially more difficult to prove once the symmetry of the fully anonymous shuffler is lost. In fact, we do not even know the mass that the γ-imperfect shuffler places on each permutation.

From private real summation to statistical security of summation on fixed fields. We first use an observation from [BBGN20] that reduces the problem of private real summation to the problem of private summation on a fixed field of size q, so that each user has an input $x_i \in \mathbb{F}_q$ for all $i \in [n]$. We then consider the well-known split-and-mix protocol [IKOS06], where each user i outputs a set of m messages $x_{i,1}, \ldots, x_{i,m} \in \mathbb{F}_q$ uniformly at random conditioned on $x_{i,1} + \ldots + x_{i,m} = x_i$. For the private summation on a fixed field problem, we adapt a well-known reduction [BBGN20] for the split-and-mix protocol from differential privacy in the shuffle model to the notion of statistical security in the γ-I-shuffle model. Statistical security demands small total variation between the output of a protocol on input x and input x', if $\sum_{i=1}^n x_i = \sum_{i=1}^n x'_i$. In other words, it suffices to show that the output distribution looks "similar" for two inputs with the same sum. See Definition 1.5 for a formal definition of statistical security.

To show statistical security, we first upper-bound the total variation distance in terms of the probability that two independent instances of the same protocol with the *same* input give the same output. [BBGN20] uses a similar approach, but then utilizes the symmetry of the fully anonymous shuffler to further upper-bound this quantity in terms of the probability that $\vec{\mathcal{R}}(\vec{\mathsf{X}}) = \mathcal{S} \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$, where $\vec{\mathsf{X}} = (x_1, \ldots, x_n)$ is the input vector, $\vec{\mathcal{R}}$ and $\vec{\mathcal{R}}'$ are independent instances of the local randomizer, and $\mathcal{S}$ is an instance of the uniform shuffler. We do not have access to such symmetries in the γ-I-shuffle model or even explicit probabilities that the γ-imperfect shuffler places on each permutation.

Connected components on a communication graph. Instead, we first upper-bound the total variation distance by $\vec{\mathcal{R}}(\vec{\mathsf{X}}) = \mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$, where $\mathcal{S}^{-1}$ is the inverse of an instance of a γ-imperfect shuffle and $\mathcal{S}'$ is an independent instance of the same γ-imperfect shuffle. Intuitively, $\vec{\mathcal{R}}(\vec{\mathsf{X}})$ and $\mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$ can look very different if there exists a large number of users whose messages are not shuffled with those of other users. Formally, this can be captured by looking at the number of connected components in the communication graph of $\mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$, so that there exists an edge connecting users i and j if the protocol swaps one of their messages. Hence, evaluating the number of connected components within the communication graph is closely related to analyzing the probability that there is no edge between S and $[n] \setminus S$, for a given set $S \subseteq [n]$.

Although this quantity would be somewhat straightforward to evaluate for a uniform shuffler [BBGN20], it seems more challenging to evaluate for γ-imperfect shufflers, since we do not have explicit probabilities for each permutation. Therefore, we develop a novel coupling argument to relate the probability that there is no edge between S and $[n] \setminus S$ in the γ-I-shuffle model to the probability of this event in the shuffle model. In particular, a specific technical challenge that our argument handles is when both S and $[n] \setminus S$ has large cardinality, because then there can be a permutation π that swaps many coordinates while still leaving S and $[n] \setminus S$ disconnected. However, if we simply relate the probability of Π in the shuffle and the γ-I-shuffle model, we incur a gap of $e^{t \cdot \gamma}$, where γ is the distortion parameter and t is the number of swaps by Π, which can have size $\Omega(n)$. Thus without additional care, this gap can overwhelm the probability achieved from the coupling argument. We circumvent this issue by considering a subset of S with size k and coupling the "good" permutations in the shuffle and the γ-I-shuffle model, which results in a smaller gap of $e^{k \cdot \gamma}$. For more details, see Lemma 4.10.

Putting things together. At this point, we are almost done. Unfortunately, our coupling only addresses the case where a single imperfect shuffle is performed on a local randomizer, but we require the bound for the composition $\mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$, which seems significantly more challenging because communication between users i and j under $\mathcal{S}'$ may be "erased" by $\mathcal{S}^{-1}$. Instead, we show a simple observation for γ-imperfect shuffling, which states that if $\mathcal{S}, \mathcal{S}'$ are two shufflers such that $\mathcal{S}$ is a γ-imperfect shuffler, then $\mathcal{S}' \circ \mathcal{S}$ is a γ-imperfect shuffler. This statement, presented in Lemma 4.7, can be considered as a post-processing preservation property of γ-imperfect shuffling. In light of this statement, we can now view $\mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})$ as a single γ-imperfect shuffler applied to the local randomizer, and use our new results upper-bounding the number of connected components in the resulting communication graph to ultimately show σ-security.

1.4 Preliminaries

For an integer $n > 0$, we define $[n] := \{1, \ldots, n\}$.

Definition 1.4 (Total variation distance)**.** *Given probability measures μ, ν on a domain Ω, their total variation distance is defined by*

$$\mathrm{TVD}(\mu, \nu) = \frac{1}{2}\|\mu - \nu\|_1 = \frac{1}{2}\sum_{x \in \Omega} |\mu(x) - \nu(x)|.$$

Definition 1.5 (σ-security)**.** *Given a security parameter $\sigma > 0$, a protocol $\mathcal{P}$ is σ-secure for computing a function $f : \mathcal{X}^n \to Z$ if, for any $x, x' \in \mathcal{X}^n$ such that $f(x) = f(x')$, we have*

$$\mathrm{TVD}(\mathcal{P}(x), \mathcal{P}(x')) \le 2^{-\sigma}.$$

Recall the following two well-known properties of differential privacy:

Theorem 1.6 (Basic Composition of differential privacy, e.g., [DR14])**.** *Let $\varepsilon, \delta \ge 0$. Any mechanism that permits k adaptive interactions with mechanisms that preserve (ε, δ)-differential privacy guarantees $(k\varepsilon, k\delta)$-differential privacy.*

Theorem 1.7 (Post-processing of differential privacy [DR14])**.** *Let $\mathcal{M} : \mathcal{U}^* \to X$ be an (ε, δ)-differential private algorithm. Then, for any arbitrary random mapping $g : X \to X'$, we have that $g(\mathcal{M}(x))$ is (ε, δ)-differentially private.*

We use $\mathsf{DLap}(\alpha)$ to denote the discrete Laplace distribution, so that $Z \sim \mathsf{DLap}(\alpha)$ has the probability mass function $\mathbf{Pr}[Z = k] \propto \alpha^{|k|}$ for $k \in \mathbb{Z}$. We use $\mathsf{Polya}(r, p)$ to denote the Polya distribution with parameter $r > 0, p \in (0, 1)$, which induces the probability density function $k \mapsto \binom{k+r-1}{k} p^k (1-p)^r$ for $k \in \mathbb{Z}_{\ge 0}$. We require the following equivalence between a discrete Laplacian random variable and the sum of a differences of Polya random variables.

Fact 1.8. *Let $x_1, \ldots, x_n, y_1, \ldots, y_n \sim \mathsf{Polya}\left(\frac{1}{n}, \alpha\right)$. Then $z = \sum_{i=1}^n (x_i - y_i) \sim \mathsf{DLap}(\alpha)$.*

We also require the following property about randomized rounding.

Lemma 1.9. *[BBGN19a] Given a precision $p \ge 1$, let $x_1, \ldots, x_n \in \mathbb{R}$ and $y_i = \lfloor x_i p \rfloor + \mathsf{Ber}(x_i p - \lfloor x_i p \rfloor)$ for each $i \in [n]$. Then*

$$\mathbb{E}\left[\left(\sum_{i=1}^n \left(x_i - \frac{y_i}{p}\right)\right)^2\right] \le \frac{n}{4p^2}.$$

1.5 Related Work

To amplify the privacy in the shuffle model, the trusted shuffler is the key component of the shuffle model, which in some sense only shifts the point of vulnerability from the curator to the shuffler, particularly in the case where the shuffler may be colluding with the curator. Hence among the various relaxations for distributed DP protocols, e.g. [BKM$^+$20, CY23], the DO shuffle model has been recently proposed [SW21, GKLX22] to permit some differentially private leakage in the shuffling stage, called a DO shuffle. In fact, [SW21, GKLX22] showed that DO-shuffling can be more efficient to achieve than a fully anonymous shuffle while [ZS22, ZSCM22] showed that locally private protocols can be used in conjunction with a DO shuffler to achieve almost the same privacy amplification bounds as with a fully anonymous shuffler, up to a small additive loss resulting from the DO shuffle. However, the best known results in the shuffle model of DP do not utilize LDP protocols, and thus cannot directly be applied in the framework of [ZS22, ZSCM22].

2 A Simple Reduction

In this section, we briefly describe a simple reduction for showing amplification of privacy for imperfect shuffling. The result can be viewed as in the same spirit as similar privacy amplification statements, e.g., [FMT21, ZS22, FMT23], but for imperfect shuffling. In particular, the following well-known result achieves privacy amplification for local randomizers in the shuffle model:

Theorem 2.1. *[FMT21] For any domain $\mathcal{D}$ and $i \in [n]$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \ldots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$, where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$, such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \ldots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \ldots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniform random permutation π over $[n]$ and sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Then for any $\delta \in [0,1]$ such that $\varepsilon_0 \le \log\left(\frac{n}{16\log(2/\delta)}\right)$, $\mathcal{A}_s$ is (ε, δ)-DP for*

$$\varepsilon \le \log\left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0}\log(4/\delta)}}{\sqrt{n}}\right)\right).$$

We would like to show privacy amplification statements for the imperfect shuffle model that are qualitatively similar to Theorem 2.1. To that end, we first recall the following definition of differentially oblivious shufflers.

Definition 2.2 (Differentially Oblivious Shuffle)**.** *A shuffle protocol is (ε, δ)-differentially oblivious if for all adversaries $\mathcal{V}$, all $\pi, \pi' \in \Pi$, and all subsets S of the view space,*

$$\mathbf{Pr}\left[\mathsf{View}^{\mathcal{V}}(\pi) \in S\right] \le e^{\varepsilon \cdot \mathrm{Swap}(\pi, \pi')}\mathbf{Pr}\left[\mathsf{View}^{\mathcal{V}}(\pi') \in S\right] + \delta.$$

[ZS22] showed that differentially oblivious shufflers also amplify privacy.

Theorem 2.3 (Theorem 1 in [ZS22])**.** *For any domain $\mathcal{D}$ and range space $\mathcal{S}$, $i \in [n]$, let $\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)} : \mathcal{D} \to \mathcal{S}$ be ε_0-DP local randomizers and let $\mathcal{A}_s$ be a $(\varepsilon_1, \delta_1)$-DO shuffler. Then the composed protocol $\mathcal{A}_s(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)})$ is $(\varepsilon + \varepsilon_1, \delta + \delta_1)$-DP for*

$$\varepsilon = \mathcal{O}\left(\frac{(1 - e^{\varepsilon_0})e^{\varepsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right).$$

It turns out that imperfect shufflers can be parametrized by differentially oblivious shufflers. That is, imperfect shufflers are a specific form of differentially oblivious shufflers. Therefore, we can immediately apply the previous statement to obtain the following statement for privacy amplification for imperfect shufflers.

Theorem 2.4. *For any domain $\mathcal{D}$ and range space $\mathcal{S}$, $i \in [n]$, let $\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)} : \mathcal{D} \to \mathcal{S}$ be ε_0-DP local randomizers and let $\mathcal{A}_s$ be a γ-imperfect shuffler. Then the composed protocol $\mathcal{A}_s(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)})$ is $(\varepsilon + \gamma, \delta)$-DP for*

$$\varepsilon = \mathcal{O}\left(\frac{(1 - e^{\varepsilon_0})e^{\varepsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right).$$

Proof. By the definition of γ-imperfect shuffle, we have that for all $\pi, \pi' \in \Pi$,

$$\mathbf{Pr}\left[\mathcal{S} = \pi\right] \le e^{\gamma \cdot \mathsf{Swap}(\pi,\pi')}\mathbf{Pr}\left[\mathcal{S} = \pi'\right].$$

Since no additional information is leaked by the shuffler, then for all adversaries $\mathcal{V}$ and all subsets S of the view space,

$$\mathbf{Pr}\left[\mathsf{View}^{\mathcal{V}}(\pi) \in S\right] \le e^{\gamma \cdot \mathsf{Swap}(\pi,\pi')}\mathbf{Pr}\left[\mathsf{View}^{\mathcal{V}}(\pi') \in S\right].$$

In other words, the γ-imperfect shuffler is a $(\gamma, 0)$-DO shuffler. Thus by Theorem 2.3, the composed protocol $\mathcal{A}_s(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)})$ is $(\varepsilon + \gamma, \delta)$-DP for

$$\varepsilon = \mathcal{O}\left(\frac{(1 - e^{\varepsilon_0})e^{\varepsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right).$$

□

3 Differentially Private Summation

In this section, we first introduce the structural statements necessary to argue privacy for the standard split-and-mix protocol [IKOS06]. We then assume correctness of these statements, deferring their proofs to subsequent sections, and we prove the guarantees of Theorem 1.3. We also give an application to private vector aggregation as a simple corollary of Theorem 1.3.

We first relate differentially private protocols for summation under a γ-imperfect shuffler to σ-secure protocols. Lemma 4.1 in [BBGN19a] showed this relationship for uniform shufflers. It turns out their proof extends to γ-imperfect shufflers as well. For the sake of completeness, we include the proof in Appendix A.

Lemma 3.1. *[Lemma 4.1 in [BBGN19a]] Given a σ-secure protocol Ξ in the γ-I-shuffle model for n-party private summation on $\mathbb{Z}_q$ such that each player sends $f(n, q, \sigma)$ bits of messages, there exists a $(\varepsilon, (1+e^{\varepsilon})2^{-\sigma-1})$-differentially private protocol in the γ-I-shuffle model for n-party private summation on real numbers with expected absolute error $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ such that each player sends $f(n, O(n^{3/2}), \sigma)$ bits of messages.*

In Section 4, we prove the following guarantees about the split-and-mix protocol from [IKOS06].

Theorem 3.2. *Let $n \geq 19$ and $\gamma \leq \frac{\log\log n}{80}$ be a distortion parameter. For worst-case statistical security with parameter σ, it suffices to use*

$$m = \mathcal{O}\left(e^{4\gamma} + \frac{e^{4\gamma}(\sigma + \log n)}{\log n}\right)$$

messages. Each message uses $\mathcal{O}(\log q)$ bits, for $q = \lceil 2n^{3/2} \rceil$.

By Lemma 3.1 and Theorem 3.2, we have our main statement:

Pasin: I don't understand this min$\left(\frac{\log\log n}{80}, \frac{\log n}{4m}\right)$ term, isn't the first term always smaller than the second term?

Samson: Oops, yes

Theorem 1.3. *Let $n \geq 19$ and $\gamma \leq \frac{\log\log n}{80}$ be a distortion parameter. Then there exists an (ε, δ)-DP protocol for summation in the γ-I-shuffle model with expected absolute error $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ and*

$$m = \mathcal{O}\left(e^{4\gamma} + \frac{e^{4\gamma}(\log\frac{1}{\delta} + \log n)}{\log n}\right)$$

messages per party. Each message uses $\mathcal{O}(\log q)$ bits, for $q = \lceil 2n^{3/2} \rceil$.

Applications to private vector summation. An immediate application of our results is to the problem of private vector aggregation, where n parties have vectors $\vec{x_1}, \ldots, \vec{x_n} \in [0,1]^d$ and the goal is to privately compute $\vec{X} = \sum_{i=1}^n \vec{x_i} \in \mathbb{R}^d$. Given a protocol $\mathcal{P}$ for private summation where n players each send m messages, the n players can perform a protocol $\mathcal{P}'$ for vector aggregation by performing $\mathcal{P}$ on each of their d coordinates. In particular, the n players can first perform $\mathcal{P}$ on the first coordinate of their vectors, then perform $\mathcal{P}$ on the second coordinate of their vectors, and so on and so forth, by sending md messages in total. Equivalently, the n players can perform $\mathcal{P}$ on a field of size q^d rather than size q and just send m messages in total. However, the total communication size is still the same, because each message increases by a factor of d due to the larger field size. Thus we consider the approach where the n players perform $\mathcal{P}$ on each of the d coordinates.

To argue privacy, we observe that the n players run d iterations of the protocol $\mathcal{P}$, once for each of the coordinates. By composition of DP, i.e., Theorem 1.6, to guarantee ε-privacy for the overall protocol, it suffices to run each of the d iterations with privacy $\varepsilon' = \frac{\varepsilon}{d}$ and failure probability $\delta' = \frac{\delta}{d}$. By post-processing of DP, i.e., Theorem 1.7, the resulting vector where each coordinate is computed using the corresponding protocol is (ε, δ)-DP.

Then as a corollary to Theorem 1.3 with privacy parameter $\varepsilon' = \frac{\varepsilon}{d}$ and failure probability $\delta' = \frac{\delta}{d}$:

Theorem 3.3. *Let $n \geq 19$, $d \geq 1$, $\varepsilon > 0$ be a (constant) privacy parameter, and $\gamma \leq \frac{\log\log n}{80}$ be a distortion parameter. Then there exists an (ε, δ)-DP protocol for vector summation in the γ-I-shuffle model with expected absolute error $\mathcal{O}\left(\frac{d}{\varepsilon}\right)$ per coordinate and*

$$m = \mathcal{O}\left(d\left(e^{4\gamma} + \frac{e^{4\gamma}(\log\frac{d}{\delta} + \log n)}{\log n}\right)\right)$$

messages per party. Each message uses $\mathcal{O}(\log q)$ bits, for $q = \lceil 2n^{3/2} \rceil$.

4 Security of Split-and-Mix Protocol

In this section, we prove the σ-security of the split-and-mix protocol. The proof largely attempts to follow the outline of the split-and-mix protocol analysis for private aggregation by [BBGN19a], which first reduces from worst-case input to average-case input and then analyzes the connectivity of the resulting communication graph induced by a uniform shuffle.

We similarly first reduce from worst-case input to average-case input and then analyze the connectivity of the resulting communication graph induced by a uniform shuffle. The former appears in Section 4.1 and the latter appears in Section 4.2.

However, the main challenge is that the symmetric properties of the uniform shuffler is often crucially utilized in various steps of the approach. Unfortunately, these properties do not often seem to translate to γ-imperfect shufflers, where we might not even know the mass that is placed on each permutation. Thus we need to handle a number of technical challenges to recover qualitatively similar structural properties to the uniform shuffling model. Along the way, we show that the composition of two shufflers, where the inner shuffler is a γ-imperfect shuffler, is also a γ-imperfect shuffler with the same parameter, which can be interpreted as a post-processing statement for γ-imperfect shuffling.

We first formally define the split-and-mix protocol:

Definition 4.1 (Split-and-Mix Protocol, e.g., [IKOS06])**.** *Given an integer parameter $m \geq 1$, the m-message n-player split-and-mix protocol $\mathcal{P}_{m,n}$ is defined as follows. Each player i outputs a set of m messages $x_{i,1}, \ldots, x_{i,m}$ uniformly at random conditioned on $x_{i,1} + \ldots + x_{i,m} = x_i$. For each $j \in [m]$, the set of messages $x_{1,j}, \ldots, x_{n,j}$ are then swapped according to a γ-imperfect shuffler $\mathcal{S}^{(j)}$.*

4.1 Worst-case to Average-case Reduction

In this section, we show a reduction from worst-case input to average-case input. In other words, rather than analyze the split-and-mix protocol over the worst-case input, we show it suffices to analyze the expected performance of the split-and-mix protocol across all possible inputs. The approach is nearly identical to that of [BBGN20], but they can further simplify their final expression due to the symmetric properties of the uniform shuffler, which do not hold for the γ-imperfect shuffler.

Let $\mathcal{P}_{m,n}$ denote the m-message n-player split-and-mix protocol and let $\tilde{\mathcal{P}}_{m,n}$ be defined as follows. Each player i outputs a set of $m+1$ messages $x_{i,1}, \ldots, x_{i,m+1}$ uniformly at random conditioned on $x_{i,1} + \ldots + x_{i,m+1} = x_i$. For each $i \in [n]$, we use the notation $\mathcal{R}_m(x_i) = (x_{i,1}, \ldots, x_{i,m})$ to denote the choice of the m messages for player i. Let $\mathbb{G} = \mathbb{F}_q$ and for $j \in [m]$, let $\mathcal{S}^{(j)} : \mathbb{G}^n \to \mathbb{G}^n$ be independent shufflers. Then the output of $\tilde{\mathcal{P}}_{m,n}$ is $\mathcal{S}^{(j)}$ applied to the first m messages of each player, concatenated with the unshuffled final message of each player, i.e.,

$$\tilde{\mathcal{P}}_{m,n}(x_1, \ldots, x_n) = \mathcal{S}^{(1)}(x_{1,1}, \ldots, x_{n,1}) \circ \ldots \circ \mathcal{S}^{(m)}(x_{1,m}, \ldots, x_{n,m}) \circ x_{1,m+1}, \ldots, x_{n,m+1}.$$

We first reduce the problem to average-case statistical security using the approach of Lemma 6.1 in [BBGN20]. Formally, we say that a protocol $\mathcal{P}_{m,n}$ provides average-case statistical security with parameter σ if

$$\mathbb{E}_{\vec{X},\vec{X}'}[\underset{|\vec{X},\vec{X}'}{\mathrm{TVD}}(\mathcal{P}_{m,n}(\vec{X}), \mathcal{P}_{m,n}(\vec{X}'))] \leq 2^{-\sigma},$$

where $\vec{\mathsf{X}}$ and $\vec{\mathsf{X}}'$ are each drawn uniformly at random from all pairs of vectors in $\mathbb{G}^n$ with the same sum. Here we use the notation $\mathrm{TVD}_{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}$ to denote the total variation distance between two distributions conditioned on fixings of $\vec{\mathsf{X}}$ and $\vec{\mathsf{X}}'$.

Lemma 4.2. *Suppose $\mathcal{P}_{m,n}$ provides average-case statistical security with parameter σ, then $\mathcal{P}_{m+1,n}$ and $\tilde{\mathcal{P}}_{m,n}$ provide worst-case statistical security with parameter σ.*

Proof. Let $\vec{x}$ and $\vec{x'}$ be a pair of vectors in $\mathbb{G}^n$ with the same sum. Given an output of $\tilde{\mathcal{P}}_{m,n}(\vec{x})$, the protocol $\mathcal{P}_{m+1,n}(\vec{x})$ can be simulated by using an additional application of $\mathcal{R}_{m+1}$ to randomly permute the last message of each of the players according to the distribution of the γ-imperfect shuffle. Hence,

$$\mathrm{TVD}(\mathcal{P}_{m+1,n}(\vec{x}), \mathcal{P}_{m+1,n}(\vec{x'})) \leq \mathrm{TVD}(\tilde{\mathcal{P}}_{m,n}(\vec{x}), \tilde{\mathcal{P}}_{m,n}(\vec{x'})).$$

It thus suffices to upper bound the worst-case statistical security of $\tilde{\mathcal{P}}_{m,n}$ by σ.

The worst-case security of $\tilde{\mathcal{P}}_{m,n}$ is reduced to the average-case security of $\tilde{\mathcal{P}}_{m,n}$ by noting that the addition of the $(m+1)$-th message to each player can effectively be viewed as adding a random value to each player's input and thus transforming each input value x_i into a uniformly random value in $\mathbb{G}$. More formally, consider the definition

$$\mathcal{R}_{m+1}(x) = (\mathcal{R}_m(x - \mathsf{U}), \mathsf{U}),$$

for $x \in \mathbb{G}$, where U is a uniformly random element of $\mathbb{G}$.

Since $\vec{x} - \vec{\mathsf{U}}$ is a uniformly random vector in $\mathbb{G}^n$, then we can couple the randomness observed from two instances $\vec{\mathsf{U}}, \vec{\mathsf{U}}'$ resulting from two independent executions of $\mathcal{P}_{m,n}$ with two inputs having the same sum. Therefore,

$$\begin{aligned}\mathrm{TVD}(\tilde{\mathcal{P}}_{m+1,n}(\vec{x}), \tilde{\mathcal{P}}_{m+1,n}(\vec{x'})) &= \mathrm{TVD}((\mathcal{P}_{m,n}(\vec{x} - \vec{\mathsf{U}}), \vec{\mathsf{U}}), (\mathcal{P}_{m,n}(\vec{x'} - \vec{\mathsf{U}}'), \vec{\mathsf{U}}')) \\ &= \mathbb{E}_{\vec{\mathsf{U}},\vec{\mathsf{U}}'}[\mathrm{TVD}(\mathcal{P}_{m,n}(\vec{x} - \vec{\mathsf{U}}), \mathcal{P}_{m,n}(\vec{x'} - \vec{\mathsf{U}}'))]] \\ &= \mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\mathrm{TVD}(\mathcal{P}_{m,n}(\vec{\mathsf{X}}), \mathcal{P}_{m,n}(\vec{\mathsf{X}}'))],\end{aligned}$$

where $\vec{\mathsf{X}}, \vec{\mathsf{X}}'$ are chosen uniformly at random conditioned on $\vec{\mathsf{X}} = \vec{\mathsf{X}}' + \vec{x} - \vec{x'}$. □

We now upper bound the expected total variation distance between the two independent executions of the γ-imperfect shuffle, using an approach similar to Lemma C.1 in [BBGN20].

Lemma 4.3. *Let $\vec{\mathsf{X}}$ and $\vec{\mathsf{X}}'$ be drawn uniformly at random from all pairs of vectors in $\mathbb{G}^n$ with the same sum, noting that $\vec{\mathsf{X}}$ and $\vec{\mathsf{X}}'$ are not independent. For two independent executions $\mathcal{P}_{m,n}$ and $\mathcal{P}'_{m,n}$ of the γ-imperfect shuffle,*

$$\mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\underset{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}{\mathrm{TVD}}(\mathcal{P}_{m,n}(\vec{\mathsf{X}}), \mathcal{P}_{m,n}(\vec{\mathsf{X}}'))] \leq \sqrt{q^{mn-1}\mathbf{Pr}\left[\mathcal{P}_{m,n}(\vec{\mathsf{X}}) = \mathcal{P}'_{m,n}(\vec{\mathsf{X}})\right] - 1}.$$

Proof. We write $\mathcal{P}$ and $\mathcal{P}'$ as shorthand for $\mathcal{P}_{m,n}$ and $\mathcal{P}'_{m,n}$, respectively. Let $\vec{\mathsf{V}}$ be a uniformly random vector drawn from $\mathbb{G}^{mn}$, conditioned on $\vec{\mathsf{V}}$ having the same sum as $\vec{\mathsf{X}}$ and $\vec{\mathsf{X}}'$. Then by the triangle inequality,

$$\mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\underset{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \mathcal{P}(\vec{\mathsf{X}}'))] \leq \mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\underset{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}}) + \underset{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}{\mathrm{TVD}}(\vec{\mathsf{V}}, \mathcal{P}(\vec{\mathsf{X}}'))]$$

$$
\begin{aligned}
&= \mathbb{E}_{\vec{\mathsf{X}}}[\underset{|\vec{\mathsf{X}}}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}})] + \mathbb{E}_{\vec{\mathsf{X}}'}[\underset{|\vec{\mathsf{X}}'}{\mathrm{TVD}}(\vec{\mathsf{V}}, \mathcal{P}(\vec{\mathsf{X}}'))]\\
&= 2\mathbb{E}_{\vec{\mathsf{X}}}[\underset{|\vec{\mathsf{X}}}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}})].
\end{aligned}
$$

Moreover, considering the distribution over $\vec{\mathsf{V}}$,

$$
\begin{aligned}
2\,\underset{|\vec{\mathsf{X}}}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}}) &= \sum_{\vec{v}\in\mathbb{G}^{mn}} \left|\mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{v}\right] - \mathbf{Pr}\left[\vec{\mathsf{V}} = \vec{v}\right]\right|\\
&= \sum_{\vec{v}\in\mathbb{G}^{mn}, \sum\vec{v}=\sum\vec{\mathsf{X}}} |\mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{v}\right] - q^{1-mn}|\\
&= q^{mn-1}\mathbb{E}_{\vec{\mathsf{V}}}\left[\left|\mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{\mathsf{V}}\right] - q^{1-mn}\right|\right].
\end{aligned}
$$

Since $\vec{\mathsf{V}}$ is a uniformly random vector from $\mathbb{G}^{mn}$ with its sum being equal to that of $\vec{\mathsf{X}}$, then for the random variable $\mathcal{Z} := \mathcal{Z}(\mathsf{X}, \mathsf{V}) := \mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{\mathsf{V}}\right]$, we have $\mathbb{E}\left[\mathcal{Z}\right] = q^{1-mn}$. Therefore,

$$
2\,\underset{|\vec{\mathsf{X}}}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}}) \le q^{mn-1}\mathbb{E}[|\mathcal{Z} - \mathbb{E}[\mathcal{Z}]|].
$$

By convexity,

$$
\mathbb{E}[|\mathcal{Z} - \mathbb{E}[\mathcal{Z}]|] \le \sqrt{\mathbb{E}[\mathcal{Z}^2]}.
$$

Since we have

$$
\begin{aligned}
\mathbb{E}_{\vec{\mathcal{V}}}[\mathcal{Z}^2] &= q^{1-mn}\sum_{\vec{v}\in\mathbb{G}^{mn}, \sum\vec{v}=\sum\vec{\mathsf{X}}} \mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{v}\right]^2\\
&= q^{1-mn}\mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \mathcal{P}'(\vec{\mathsf{X}})\right],
\end{aligned}
$$

we thus have

$$
\begin{aligned}
\mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\underset{|\vec{\mathsf{X}},\vec{\mathsf{X}}'}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \mathcal{P}(\vec{\mathsf{X}}'))] &\le 2\,\underset{|\vec{\mathsf{X}}}{\mathrm{TVD}}(\mathcal{P}(\vec{\mathsf{X}}), \vec{\mathsf{V}})\\
&\le q^{mn-1}\mathbb{E}_{\mathcal{V}(\vec{\mathsf{X}}')}[|\mathbf{Pr}\left[\mathcal{P}(\vec{\mathsf{X}}) = \vec{\mathsf{V}}\right] - q^{1-mn}|]\\
&\le \sqrt{q^{mn-1}\mathbf{Pr}\left[\mathcal{P}_{m,n}(\vec{\mathsf{X}}) = \mathcal{P}'_{m,n}(\vec{\mathsf{X}})\right] - 1}. \qquad \square
\end{aligned}
$$

We note that the probability that two independent executions of the protocol can be decomposed into the split protocol and the mix protocol as follows. By comparison, Lemma C.2 in [BBGN20] was able to prove a simpler relationship by leveraging properties of their symmetric shuffler, which we do not have for an imperfect shuffler.

Lemma 4.4. *Let $\mathcal{R}_{m,n}$ and $\mathcal{R}'_{m,n}$ denote two independent executions of the split protocol in $\mathcal{P}_{m,n}$ so that $\mathcal{P}_{m,n} = \mathcal{S}_{m,n} \circ \mathcal{R}_{m,n}$. Then*

$$
\mathbf{Pr}\left[\mathcal{P}_{m,n}(\vec{\mathsf{X}}) = \mathcal{P}'_{m,n}(\vec{\mathsf{X}})\right] = \mathbf{Pr}\left[\mathcal{R}_{m,n}(\vec{\mathsf{X}}) = \mathcal{S}^{-1}_{m,n} \circ \mathcal{S}'_{m,n} \circ \mathcal{R}'_{m,n}(\vec{\mathsf{X}})\right].
$$

Proof. Note that

$$\begin{aligned}\mathbf{Pr}\left[\mathcal{P}_{m,n}(\vec{\mathsf{X}})=\mathcal{P}'_{m,n}(\vec{\mathsf{X}})\right]&=\mathbf{Pr}\left[\mathcal{S}_{m,n}\circ\mathcal{R}_{m,n}(\vec{\mathsf{X}})=\mathcal{S}'_{m,n}\circ\mathcal{R}'_{m,n}(\vec{\mathsf{X}})\right]\\&=\mathbf{Pr}\left[\mathcal{R}_{m,n}(\vec{\mathsf{X}})=\mathcal{S}^{-1}_{m,n}\circ\mathcal{S}'_{m,n}\circ\mathcal{R}'_{m,n}(\vec{\mathsf{X}})\right].\end{aligned}$$

□

From Lemma 4.3 and Lemma 4.4, we have

Lemma 4.5. *For two independent executions $\mathcal{P}_{m,n}$ and $\mathcal{P}'_{m,n}$ of the split-and-mix protocol with a γ-imperfect shuffler,*

$$\mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\mathrm{TVD}(\mathcal{P}_{m,n}(\vec{\mathsf{X}}),\mathcal{P}_{m,n}(\vec{\mathsf{X}}'))]\le\sqrt{q^{mn-1}\mathbf{Pr}\left[\mathcal{R}_{m,n}(\vec{\mathsf{X}})=\mathcal{S}^{-1}_{m,n}\circ\mathcal{S}'_{m,n}\circ\mathcal{R}'_{m,n}(\vec{\mathsf{X}})\right]-1}.$$

4.2 Reduction to Connected Components

In this section, we prove the following general statement upper bounding the probability that the shuffler $\mathcal{S}^{-1}_{m,n}\circ\mathcal{S}'_{m,n}(\cdot)$ on the output of a randomizer achieves the same output as an independent instance of the randomizer by the expectation of a quantity relating to the number of connected components in the communication graph of the shuffler $\mathcal{S}^{-1}_{m,n}\circ\mathcal{S}'_{m,n}(\cdot)$. Specifically, we can view a protocol $\mathcal{P}_{m,n}$ that is an ordered pair $\pi_1,\ldots,\pi_m$, where π_j is a permutation on $[n]$ for each $j\in[m]$, so that in each round $j\in[m]$, user $i\in[n]$ sends a message to user $\pi_j(i)$.

Then we can define the *communication graph* for the multi-message shuffle protocol $\mathcal{P}_{m,n}$ as follows. The graph G consists of n vertices, which we associate with $[n]$, corresponding to the players $[n]$ participating in the protocol $\mathcal{P}_{m,n}$. We add an edge between vertices i and j if player i passes one of their m messages to player j.

The following proof is the same as Lemma C.4 in [BBGN20].

Lemma 4.6. *Let G be the graph on n vertices formed the communication graph of the shuffle $\mathcal{S}^{-1}\circ\mathcal{S}'$. Let $C(G)$ be the number of connected components of G. Then*

$$\mathbf{Pr}\left[\vec{\mathcal{R}}(\vec{\mathsf{X}})=\mathcal{S}^{-1}\circ\mathcal{S}'\circ\vec{\mathcal{R}}'(\vec{\mathsf{X}})\right]\le\mathbb{E}\left[q^{C(G)-mn}\right].$$

Proof. By the law of total expectation,

$$\mathbf{Pr}\left[\vec{\mathcal{R}}(\vec{\mathsf{X}})=\mathcal{S}^{-1}\circ\mathcal{S}'\circ\vec{\mathcal{R}}'(\vec{\mathsf{X}})\right]=\mathbb{E}\left[\mathbf{Pr}\left[\vec{\mathcal{R}}(\vec{\mathsf{X}})=\mathcal{S}^{-1}\circ\mathcal{S}'\circ\vec{\mathcal{R}}'(\vec{\mathsf{X}})\mid\mathcal{S},\mathcal{S}'\right]\right].$$

Thus for the graph G conditioned on $\mathcal{S}$ and $\mathcal{S}'$, it suffices to show that

$$\mathbf{Pr}\left[\vec{\mathcal{R}}(\vec{\mathsf{X}})=\mathcal{S}^{-1}\circ\mathcal{S}'\circ\vec{\mathcal{R}}'(\vec{\mathsf{X}})\mid\mathcal{S},\mathcal{S}'\right]=q^{C(G)-mn}.$$

Note that $C(G)$ depends on the choices of $\mathcal{S}$ and $\mathcal{S}'$ but we omit these dependencies in the notation for the sake of presentation. Recall that $\mathcal{P}_{m,n}(\vec{\mathsf{X}})=\mathcal{S}_{m,n}\circ\vec{\mathcal{R}}_{m,n}(\vec{\mathsf{X}})$ is currently indexed so that the first message of each player after the shuffle protocol completes are the first n indices, followed by the second message of each of the n players and so forth. We thus define a re-indexing permutation

$\psi : [mn] \to [mn]$ to that the m messages of the first player will be the first m indices, followed by the m messages of the second player and so forth. That is,

$$\psi(j) = \left\lfloor \frac{j-1}{m} \right\rfloor + n(j - 1 \bmod n) + 1.$$

Let $\mathsf{W}, \mathsf{W}' \in \mathbb{G}^{mn}$ be defined so that $\mathsf{W}_j = \psi(\vec{\mathcal{R}}(\vec{\mathsf{X}}))_j$ and $\mathsf{W}'_j = \psi(\mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}}))_j$. The task then becomes to show that

$$\mathbf{Pr}\left[\mathsf{W} = \mathsf{W}' \mid \mathcal{S}, \mathcal{S}'\right] = q^{C(G)-mn}.$$

Toward that end, for each $j \in [mn]$, we define $\mathcal{E}_j$ to be the event that $\mathsf{W}_j = \mathsf{W}'_j$ and $p_j = \mathbf{Pr}\left[\mathcal{E}_j \mid \mathcal{E}_1, \ldots, \mathcal{E}_{j-1}\right]$, so that

$$\mathbf{Pr}\left[\mathsf{W} = \mathsf{W}' \mid \mathcal{S}, \mathcal{S}'\right] = \prod_{j=1}^{m} np_j.$$

Firstly, consider the messages that are not the last message by a particular player, i.e., consider the values of $j \in [mn]$ that are not divisible by m. Observe that conditioning on fixed values of $\vec{\mathsf{X}}$ and $\vec{\mathcal{R}}'$, as well as the events $\mathcal{E}_1, \ldots, \mathcal{E}_{j-1}$, the value of W_j remains uniformly distributed and has probability q^{-1} of being equal to to W'_j. Hence, we have $p_j = q^{-1}$.

For the cases where j is divisible by m, we further consider two subcases. In particular, we consider the case where j is the largest index in C_j and the case where j is not the largest index in C_j, where C_j is the set of vertices in the same connected component as j in G.

In the first subcase, the multisets of W' and $\vec{\mathcal{R}}'(\mathsf{X}')$ restricted to C_i are the same and thus the multisets of the summands are the same, so that

$$\sum_{i \mid C_i = C_j} \mathsf{W}'_i = \sum_{i \mid C_i = C_j} \psi(\vec{\mathcal{R}}'(\mathsf{X}'))_i.$$

Moreover, since the indices corresponding to all messages of a fixed player are in the same connected component, then

$$\sum_{i \mid C_i = C_j} \psi(\vec{\mathcal{R}}'(\mathsf{X}'))_i = \sum_{i \mid C_i = C_j} \mathsf{W}_i.$$

Finally, we have that conditioning on $\mathcal{E}_1, \ldots, \mathcal{E}_{j-1}$ and the fact that j is the largest index in C_j,

$$\sum_{i \mid C_i = C_j, i \neq j} \mathsf{W}'_i = \sum_{i \mid C_i = C_j, i \neq j} \mathsf{W}_i.$$

Therefore, we have $p_j = 1$.

For the second subcase, we shall show that $p_j = q^{-1}$. Let $\mathcal{T}$ be the subset of $(\mathsf{W}, \mathsf{W}') \in \mathbb{G}^{2mn}$ that are consistent with $\mathcal{E}_1, \ldots, \mathcal{E}_{j-1}$ and a fixed value of $\vec{\mathsf{X}}$. We show there exists a homomorphism $\phi : \mathbb{G} \to \mathbb{G}^{2mn}$ that maps from $g \in \mathbb{G}$ to a $u_g \in \mathbb{G}^{2mn}$ with a specific property to be defined. We then consider the action of $\mathbb{G}^{2mn}$ on itself by addition of u_g. Then the property of ϕ that we show is that u_g fixes $\mathcal{T}$ and W_j but adds g to W'_j. Consider the partitioning of $\mathcal{T}$ into equivalence classes where two elements of $\mathcal{T}$ are equivalent if they are equal under addition by u_g for some g. Then the homomorphism induces a partitioning of $\mathcal{T}$ into subsets of size q such that each subset contains exactly one element for which $\mathcal{E}_j$ holds. Since each value of $\mathcal{T}$ is equally probable, it then follows that $p_j = q^{-1}$ as desired.

We now define the homomorphism ϕ as follows. Since there exists a path in G from the vertex with the j-th message to a higher index vertex, then there exists some path parameter ℓ and a corresponding path $(a_1, b_1, \ldots, a_\ell, b_\ell, a_{\ell+1})$ such that the following hold. Firstly, each of the terms a_i, b_i are elements of $[mn]$ that will ultimately map to indices of elements in $\mathbb{G}^{mn}$. Secondly, for all $i \in [\ell]$, we have $\pi(b_i) = a_i$ for the permutation π induced by the m message n player protocol and moreover, b_i and a_{i+1} correspond to the same vertex. Finally, it holds that $a_1 = j$, $b_\ell > j$, $a_i \neq a_{i'}$ for any $i \neq i'$, and $b_i < j$ for all $i < \ell$. Then we implicitly define the homomorphism ϕ by defining u_g to be the element of $\mathbb{G}^{2mn}$ with the value g in the entries $a_2, \ldots, a_{\ell+1}, b_1 + mn, \ldots, b_\ell + mn$ and the identity 0 in all other coordinates, where we recall that the elements a_i and b_i correspond to indices of elements in $\mathbb{G}^{mn}$.

We observe that the group action of addition by u_g does not affect the realization of $\mathcal{E}_1, \ldots, \mathcal{E}_{j-1}$ since W_{a_i} and $\mathsf{W}'_{a_i} = \vec{\mathcal{R}}'(\vec{\mathsf{X}})_{b_i}$ are increased by exactly the same amount by u_g, except for the case when $i = 1$ or $i = \ell + 1$. However, note that $a_i \geq j$ for both of the cases where $i = 1$ and $i = \ell + 1$, which does not affect the realization of $\mathcal{E}_1, \ldots, \mathcal{E}_{j-1}$. Hence, u_g has the desired properties and so it follows that $p_j = q^{-1}$.

Therefore, conditioned on any fixed realization of $\mathcal{S}$, we have that

$$\prod_{j=1}^{mn} p_j = q^{C(G)-mn},$$

so that in summary

$$\mathbf{Pr}\left[\vec{\mathcal{R}} = \mathcal{S}^{-1} \circ \mathcal{S}' \circ \vec{\mathcal{R}}'(\vec{\mathsf{X}})\right] \leq \mathbb{E}[q^{C(G)-mn}].$$

□

We remark that the statement of Lemma 4.6 holds even for a general shuffler $\mathcal{S}$ with the corresponding communication graph, rather than the specific shuffler $\mathcal{S}_{m,n}^{-1} \circ \mathcal{S}'_{m,n}(\cdot)$.

We now show that the composition of two shufflers, where the inner shuffler is a γ-imperfect shuffler, is also a γ-imperfect shuffler with the same parameter.

Lemma 4.7. *Let $\mathcal{S}, \mathcal{S}'$ be two shufflers such that $\mathcal{S}$ is a γ-imperfect shuffler. Then, $\mathcal{S}' \circ \mathcal{S}$ is a γ-imperfect shuffler.*

Proof. Let $\mathcal{S}'$ be an arbitrary shuffler and $\mathcal{S}$ be a γ-imperfect shuffler. Then, for any $\pi, \pi' \in \Pi$,

$$\begin{aligned}
\mathbf{Pr}\left[\mathcal{S}' \circ \mathcal{S} = \pi\right] &= \mathbf{Pr}\left[\mathcal{S} = (\mathcal{S}')^{-1} \circ \pi\right] \\
&\leq e^{\gamma \cdot \mathsf{Swap}((\mathcal{S}')^{-1} \circ \pi, (\mathcal{S}')^{-1} \circ \pi')} \mathbf{Pr}\left[\mathcal{S} = (\mathcal{S}')^{-1} \circ \pi'\right] \\
&= e^{\gamma \cdot \mathsf{Swap}(\pi, \pi')} \mathbf{Pr}\left[\mathcal{S}' \circ \mathcal{S} = \pi'\right].
\end{aligned}$$

Thus, $\mathcal{S}' \circ \mathcal{S}$ is a γ-imperfect shuffler. □

We now show a few structural statements that upper bound the probability that there exists no edge from a set $S \subset [n]$ to $[n] \setminus S$ for a communication graph induced by a γ-imperfect shuffler.

Lemma 4.8. *Let G be the communication graph of a γ-imperfect shuffler (on an n-player m-message protocol). For a fixed set S with size s, the probability that there exists no edge from S to $[n] \setminus S$ in G is at most $e^{2sm\gamma}\binom{n}{s}^{-m}$ for $s \leq \frac{n}{2}$ and at most $e^{2(n-s)m\gamma}\binom{n}{s}^{-m}$ for $s \geq \frac{n}{2}$.*

Proof. Without loss of generality, let $S = [s]$, i.e., S is the first s integers of $[n]$. Then for a permutation to not induce an edge between S and $[n] \setminus S$, the permutation can be decomposed into a permutation of the first s integers and a permutation of the remaining $n - s$ integers. Hence, there are $s!(n - s)!$ permutations of $[n]$ such that S is preserved. Let Π_S be the set of permutations that preserves S so that $|\Pi_S| = s!(n - s)!$.

For each permutation $\pi \in \Pi_S$, we define a subset C_π of permutations so that (1) $C_{\pi'} \cap C_\pi = \emptyset$ for all $\pi, \pi' \in \Pi_S$ with $\pi \neq \pi'$, (2) π is the only permutation of C_π that preserves S, (3) $|C_\pi| = \binom{n}{s}$, and (4) π and π' have swap distance at most $2s$ for any $\pi' \in C_\pi$, hence implying that $\mathbf{Pr}\left[\mathcal{S} = \pi\right] \leq e^{2s\gamma} \cdot \mathbf{Pr}\left[\mathcal{S} = \pi'\right]$. Recall that since $\pi \in \Pi_S$, then π can be decomposed into permutations π_1 of the first s integers and permutations π_2 of the remaining $n - s$ integers.

Let A be any set of s indices of $[n]$, sorted in increasing order. Consider the following transformation T_A on a permutation π to produce a permutation ψ. Place the elements of π in positions $[s]$ in order into the s indices of A, so that $\pi'(A_i) = \pi(i)$. For the supplanted indices that have not been assigned to indices in A, place them in order into the remaining positions of $[s]$. Formally, let $X = [s] \setminus A$ and $Y = A \setminus [s]$. Then we set $\pi'(X_i) = \pi(Y_i)$ for all $i \in [|X|]$, noting that $|X| = |Y|$. We then define C_π to be the set of permutations that can be obtained from this procedure, i.e., $C_\pi = \{\pi' : \exists A \text{ with } \pi = T_A(\pi)\}$. See Figure 1 for an example of the application of such an example T_A.

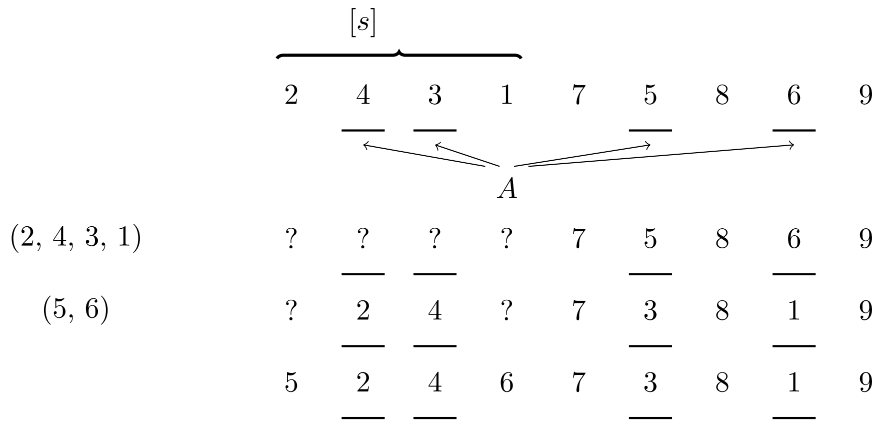

Fig. 1: An example of the transformation T_A for the permutation $\pi = (8, 4, 6, 2, 1, 3, 7, 5, 9)$, with $n = 9$, $s = 4$, and $A = (2, 3, 6, 8)$. Note that the order $(8, 4, 6, 2)$ is preserved within the indices of A in the resulting permutation $\pi' = T_A(\pi)$ and the order $(3, 5)$ is preserved within the indices $[s] - A$.

We first claim that $C_{\pi'} \cap C_\pi = \emptyset$ for all $\pi, \pi' \in \Pi_S$ with $\pi \neq \pi'$. Suppose by way of contradiction, there exists $\psi \in C_\pi \cap C_{\pi'}$, so that there exist sets A and A' with $\psi = T_A(\pi) = T_{A'}(\pi')$. Recall that since $\pi, \pi' \in \Pi_S$, then π, π' can be decomposed into permutations π_1, π'_1 of the first s integers and permutations π_2, π'_2 of the remaining $n - s$ integers. After applying T_A to π, then the first s integers are in the indices of A', in some order. Similarly, after applying $T_{A'}$ to π', then the first s integers are in the indices of A, in some order. Hence for $\psi = T_A(\pi) = T_{A'}(\pi')$, it follows that $A = A'$, so it suffices to show that T_A is injective for a fixed A.

To that end, note that T_A preserves the order of $[s]$ within A and thus for $\pi = \pi_1 \circ \pi_2$, then π_1 is the restriction of $T_A(\pi)$ to A. Similarly, note that T_A does not touch the indices outside of $A \cup [s]$ and so π_2 is preserved by $T_A(\pi)$ in the restriction of $[n] \setminus (A \cup [s])$. Finally, T_A preserves the relative order of π_2 inside the indices of $[s] \setminus A$. Therefore, given A and $T_A(\pi)$, we can completely recover π_1 and π_2 and thus π. In other words, T_A is injective, so that $T_A(\pi) = T_A(\pi')$ implies $\pi = \pi'$, which is a contradiction. Hence $C_{\pi'} \cap C_\pi = \emptyset$.

Pasin: I don't understand the previous sentence... T_A can be well-defined but many-to-one, right?

Samson: Yes sorry, added sentences that T_A is injective

To see that π is the only permutation of C_π that preserves S, note that if any of the s positions are picked outside $[s]$, then there exists a value of $[s]$ outside of the first s positions and so the resulting permutation does not preserve S. However, there is only a single way to pick s indices from $[n]$ that are all inside $[s]$, which corresponds to π. Hence, π is the only permutation of C_π that preserves S.

To see the third property, note that A is formed by choosing s indices of $[n]$. Hence, $|A| = \binom{n}{s}$. Since A is exactly the set of positions for which π_1 is mapped to, then each element of A corresponds to a unique element in C_π. Thus, $|C_\pi| = \binom{n}{s}$.

To see the fourth property, note that the only swaps are indices in A with indices in $[s]$, meaning that at most $2s$ indices are changed. Thus we have π and π' have swap distance at most $2s$ for any $\pi' \in C_\pi$. Then by the γ-imperfect shuffle property, $\mathbf{Pr}\left[\mathcal{S} = \pi\right] \le e^{2s\gamma} \cdot \mathbf{Pr}\left[\mathcal{S} = \pi'\right]$.

Since we have associated each $\pi \in \Pi_S$ with a set C_π of size $\binom{n}{s}$ such that $\pi' \notin C_\pi$ for $\pi' \in \Pi_S$ with $\pi' \neq \pi$ and $\mathbf{Pr}\left[\mathcal{S} = \pi\right] \le e^{s\gamma} \cdot \mathbf{Pr}\left[\mathcal{S} = \pi'\right]$, then it follows from a coupling argument that the probability that there exists no edge from S to $[n] \setminus S$ after one iteration of the γ-imperfect shuffle is at most $e^{2s\gamma}\binom{n}{s}^{-1}$. By independence, the probability that there exists no edge from S to $[n] \setminus S$ in G after the m iterations is at most $e^{2sm\gamma}\binom{n}{s}^{-m}$.

By symmetry for sets S with size s and $n - s$, we have the probability is at most

$$\min\left(e^{2sm\gamma}\binom{n}{s}^{-m}, e^{2(n-s)m\gamma}\binom{n}{s}^{-m}\right)$$

across all ranges of s. □

Lemma 4.9. *Let G be the communication graph of a γ-imperfect shuffler (on an n-player m-message protocol). For a fixed set S with size s, the probability that there exists no edge from S to $[n] \setminus S$ in G is at most $e^{km\gamma}\binom{n/2}{k}^{-m}$ for any integer k with $0 \le k \le \min(s, n-s)$.*

Proof. We can similarly show that the probability that there exists no edge from S to $[n] \setminus S$ in G after the m iterations is at most $e^{km\gamma}\binom{n/2}{k}^{-m}$ for any integer k with $0 \le k \le \min(s, n-s)$ by the following modifications to the coupling argument. We again let $S = [s]$ without loss of generality and let $k \le \min(s, n-s)$ be a fixed non-negative integer.

Recall that there are $s!(n-s)!$ permutations of $[n]$ such that S is preserved. We define Π_S to be the set of permutations that preserves S so that $|\Pi_S| = s!(n-s)!$ and we define a transformation $T_A(\pi)$ for a permutation $\pi \in \Pi_S$ as follows.

If $s \le \frac{n}{2}$, we let A be a set of k positions in $\{s+1, \ldots, n\}$, sorted in increasing order. We then initialize $\psi = \pi$ and iteratively perform the following procedure k times. For each $i \in [k]$, we swap the value in the i-th index of ψ with the value in the A_i-th index of A. We then output set $T_A(\pi)$ to be the result of ψ after applying these k swaps. Note that since $[s]$ and A are disjoint, we can also explicitly define the resulting $\psi = T_A(\pi)$ by

$$\psi(i) = \begin{cases} \pi(i), & i \notin (A \cup [k]) \\ \pi(A_i), & i \in [k] \\ \pi(j), & j = A_i, i \in [k]. \end{cases}.$$

Similarly, if $s \geq \frac{n}{2}$, we let A be a set of k positions in $[n-s]$, sorted in increasing order, and initialize $\psi = \pi$. Then for each $i \in [k]$, we swap the value in the $(n-i+1)$-th index of ψ with the value in the i-th index of A. Alternatively, we can also explicitly define the resulting $\psi = T_A(\pi)$ by

$$\psi(i) = \begin{cases} \pi(i), & i \notin (A \cup \{n-k+1,\ldots,n\}) \\ \pi(A_i), & i \in \{n-k+1,\ldots,n\} \\ \pi(j), & j = A_i, i \in [k]. \end{cases} .$$

We again define C_π to be the set of permutations that can be obtained from this procedure, i.e., $C_\pi = \{\pi' : \exists A \text{ with } \pi = T_A(\pi)\}$. By the same argument as in Lemma 4.9, we have (1) $C_{\pi'} \cap C_\pi = \emptyset$ for all $\pi, \pi' \in \Pi_S$ with $\pi \neq \pi'$, (2) π is the only permutation of C_π that preserves S, (3) $|C_\pi| = \binom{n}{k}$. By the construction of T_A performing k swaps on π, we also have that π and π' have swap distance at most k for any $\pi' \in C_\pi$, so that $\mathbf{Pr}\left[\mathcal{S} = \pi\right] \leq e^{k\gamma} \cdot \mathbf{Pr}\left[\mathcal{S} = \pi'\right]$.

Also by construction, we have $|C_\pi| \geq \binom{n/2}{k}$ and so by adapting the above coupling argument, we have that the probability that there exists no edge from S to $[n] \setminus S$ in G after the m iterations is at most $e^{km\gamma}\binom{n/2}{k}^{-m}$. □

By Lemma 4.8 and Lemma 4.9, we have:

Lemma 4.10. *Let G be the communication graph of a γ-imperfect shuffler (on an n-player m-message protocol). For a fixed set S with size s, the probability that there exists no edge from S to $[n] \setminus S$ in G is at most $e^{2sm\gamma}\binom{n}{s}^{-m}$ for $s \leq \frac{n}{2}$, at most $e^{2(n-s)m\gamma}\binom{n}{s}^{-m}$ for $s \geq \frac{n}{2}$, and at most $e^{km\gamma}\binom{n/2}{k}^{-m}$ for any integer k with $0 \leq k \leq \min(s, n-s)$.*

Lemma 4.7 and Lemma 4.10 are the two main structural properties of imperfect shufflers that we use to overcome the challenge of adapting the analysis of [BBGN20] to shufflers without symmetry.

We now upper bound the probability that the number of connected components of G is c, where G is the underlying communication graph for the split-and-mix-protocol under a γ-imperfect shuffle.

Lemma 4.11. *Let $n \geq 19$ and $m \geq 8e^{4\gamma}$. Let G be the communication graph of a γ-imperfect shuffler (on an n-player m-message protocol). Let $p(n,c)$ denote the probability that the number of connected components of G is c. Then*

$$p(n,c) \leq \frac{2^{c-1}}{c!}\left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)(c-1)}.$$

Pasin: One thing I am a little unsure is why we can decompose to the term $p(|S|,1)p(n-|S|,c-1)$, i.e. are the event that S has a single component and $n-|S|$ has $c-1$ components independent? This is not obvious to me since our shuffler is not uniform, so conditioning on the fact that S is separated from $n-S$ does not seem to imply that these two events become independent. In any case, we can probably get way with this by just ignoring the term $p(|S|,1)$ completely anyway. Wdyt?

Samson: Yes I agree. In fact, is it unclear whether n refers to graph size or the subgraph size?

Proof. For a fixed set S, let $\mathbb{P}_S$ denote the probability that there is no edge from S to $[n] \setminus S$. Let $p(n,c)$ denote the probability that the number of connected components of G is c. Then

$$\begin{aligned} p(n,c) &= \frac{1}{c}\sum_{S \subseteq [n]} \mathbb{P}_S \cdot p(n-|S|, c-1) \\ &\leq \frac{1}{c}\sum_{s=1}^{n-c+1}\binom{n}{s}\mathbb{P}_S \cdot p(n-|S|, c-1). \end{aligned}$$

We decompose this sum and apply Lemma 4.10.

By Lemma 4.10, we have $\mathbb{P}_S \leq \min(e^{2(n-s)m\gamma}\binom{n}{s}^{-m}, e^{2sm\gamma}\binom{n}{s}^{-m})$. By Lemma 4.10, we also have $\mathbb{P}_S \leq e^{km\gamma}\binom{n/2}{k}^{-m}$ for any $k \leq \min(s, n-s)$. Observe that for $k \geq n-s \geq \frac{n}{2}$, we have $e^{2(n-s)m\gamma}\binom{n}{s}^{-m} \leq e^{2km\gamma}\binom{n}{k}^{-m} \leq e^{2km\gamma}\binom{n/2}{k}^{-m}$. Thus for $k = \frac{n}{4e^{4\gamma}}$,

$$\begin{aligned} p(n,c) &\leq \frac{1}{c}\sum_{s=1}^{k}\binom{n}{s}\binom{n}{s}^{-m} e^{2sm\gamma} \cdot p(n-|S|, c-1) \\ &+ \frac{1}{c}\sum_{s=k+1}^{n-c+1}\binom{n}{s}\binom{n/2}{k}^{-m} e^{2km\gamma} \cdot p(n-|S|, c-1). \end{aligned}$$

Observe that $k = \frac{n}{4e^{4\gamma}}$ implies that

$$\begin{aligned} e^{2\gamma} &\leq \left(\frac{n}{2k}\right)^{1/2} \\ e^{2km\gamma} &\leq \left(\frac{n}{2k}\right)^{km/2} \leq \binom{n/2}{k}^{m/2} \\ \binom{n/2}{k}^{-m} e^{2km\gamma} &\leq \binom{n/2}{k}^{-m/2} \leq \binom{n}{k}^{-m/2}. \end{aligned}$$

Thus we have

$$\begin{aligned} p(n,c) &\leq \frac{1}{c}\sum_{s=1}^{k}\binom{n}{s}\binom{n}{s}^{-m} e^{2sm\gamma} \cdot p(n-|S|, c-1) \\ &+ \frac{1}{c}\sum_{s=k+1}^{n-c+1}\binom{n}{s}\binom{n}{k}^{-m/2} \cdot p(n-|S|, c-1). \end{aligned}$$

Since $k = \frac{n}{4e^{4\gamma}}$, then

$$\binom{n}{k}^{-m/2} \leq (4e^{4\gamma})^{-\frac{nm}{8e^{4\gamma}}} \leq (2e)^{-\frac{nm}{8e^{4\gamma}}} \leq \binom{n}{n/2}^{-\frac{m}{4e^{4\gamma}}} \leq \binom{n}{s}^{-\frac{m}{4e^{4\gamma}}}.$$

Hence,

$$\begin{aligned} p(n,c) &\leq \frac{1}{c}\sum_{s=1}^{k}\binom{n}{s}\binom{n}{s}^{-m} e^{2sm\gamma} \cdot p(n-|S|, c-1) \\ &+ \frac{1}{c}\sum_{s=k+1}^{n-c+1}\binom{n}{s}^{1-\frac{m}{4e^{4\gamma}}} \cdot p(n-|S|, c-1). \end{aligned}$$

For $m \geq 8e^{4\gamma}$, we have $1 \leq \frac{m}{8e^{4\gamma}}$ and thus

$$\begin{aligned} p(n,c) &\leq \frac{1}{c}\sum_{s=1}^{k}\binom{n}{s}\binom{n}{s}^{-m} e^{2sm\gamma} \cdot p(n-|S|, c-1) \\ &+ \frac{1}{c}\sum_{s=k+1}^{n-c+1}\binom{n}{s}^{-\frac{m}{8e^{4\gamma}}} \cdot p(n-|S|, c-1). \end{aligned}$$

We first apply the induction hypothesis that $p(n,c) \leq \frac{2^{c-1}}{c!} \left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{\gamma(m-1)(c-1)}$:

$$\begin{aligned}p(n,c) \leq \frac{2^{c-1}}{c!} \left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)(c-1)} \cdot \frac{1}{2} \cdot e^{\frac{(1-m)}{32e^{4\gamma}}} \cdot e^{2\gamma(1-m)} \\ \cdot \left(\sum_{s=1}^{k} \binom{n}{s}^{1-m} e^{2sm\gamma} \left(\frac{n^{c-1}}{(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}} + \sum_{s=k+1}^{n-c+1} \left(\frac{(n-s)!s!n^{c-1}}{n!(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}} \right).\end{aligned}$$

We upper bound $p(n,c)$ by upper bounding the summation across the first k terms, i.e., the head of the summation, then upper bounding the tail terms of the summation, i.e., the terms with $s \geq \frac{3n}{4}$, and finally upper bounding the remaining terms of the summation, i.e., $s \in \left[k, \frac{3n}{4}\right]$.

Upper bounding the head terms in the summation. We now upper bound the summation across all $s \leq k$. Let $a_s = \binom{n}{s}^{1-m} e^{2sm\gamma} \left(\frac{n^{c-1}}{(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}}$. For $s \leq k = \frac{n}{4e^{4\gamma}}$ and $m \geq 8e^{4\gamma}$,

$$\begin{aligned}\frac{a_s}{a_{s-1}} &= \left(\frac{s}{n-s+1}\right)^{m-1} e^{2m\gamma} \left(\frac{n-s+1}{n-s}\right)^{\frac{(m-1)(c-2)}{32e^{4\gamma}}} \\ &\leq \left(\frac{1}{8e^{4\gamma}}\right)^{m-1} e^{2m\gamma} e^{\frac{(m-1)(c-2)}{n-s}} \\ &\leq \left(\frac{1}{8e^{4\gamma}}\right)^{m-1} (e^{4\gamma})^{m-1} e^{\frac{4(m-1)}{3}} \\ &\leq \left(\frac{e^{4/3}}{8}\right)^{m-1} \leq \left(\frac{1}{2}\right)^{m-1} \leq \frac{1}{25}.\end{aligned}$$

Then through a geometric series, we bound the summation

$$\begin{aligned}\sum_{s=1}^{k} a_s \leq \sum_{s=1}^{\infty} \frac{a_1}{25^{s-1}} &\leq \frac{26a_1}{25} \\ &\leq \frac{26}{25} n^{1-m} e^{m\gamma} \left(\frac{n^{c-1}}{(n-1)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}} \\ &\leq \frac{26}{25} e^{m\gamma} e^{\frac{m-1}{32e^{4\gamma}}}\end{aligned}$$

Upper bounding the tail terms in the summation. We now upper bound the summation across all $s \geq \lceil \frac{3n}{4} \rceil$. Let $b_s = \left(\frac{(n-s)!s!n^{c-1}}{n!(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}}$. Then for $s \geq \frac{3n}{4}$,

$$\begin{aligned}\frac{b_s}{b_{s-1}} &= \left(\frac{s}{n-s+1}\left(\frac{n-s+1}{n-s}\right)^{c-2}\right)^{\frac{m-1}{32e^{4\gamma}}} \\ &\geq \left(\frac{s}{n-s}\right)^{\frac{m-1}{32e^{4\gamma}}} \geq 9.\end{aligned}$$

We again bound another subset of the sum through a geometric series:

$$\begin{aligned}\sum_{s=\lceil 3n/4\rceil}^{n-c+1} b_s \le \sum_{s=\lceil 3n/4\rceil}^{n-c+1} \frac{b_{n-c+1}}{9^{n-c+1-s}} &\\ &\sum_{s=-\infty}^{n-c+1} \frac{b_{n-c+1}}{9^{n-c+1-s}}\\ &= \frac{9b_{n-c+1}}{8}\\ &= \frac{9}{8}\left(\frac{(c-1)!(n-c+1)!n^{c-1}}{n!(c-1)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}}.\end{aligned}$$

Similar to [BBGN20], we bound the last expression using Sterling's bound, $\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n} \le n! \le en^{n+\frac{1}{2}}e^{-n}$, so that

$$\frac{9}{8}\left(\frac{(c-1)!(n-c+1)!n^{c-1}}{n!(c-1)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}} \le \frac{9}{8}\left(\frac{e}{\sqrt{2\pi}}(c-1)^{1.5}\left(1-\frac{(c-1)}{n}\right)^{n-c+1.5}\right)^{\frac{m-1}{32e^{4\gamma}}},$$

which is maximized at $c=3$ for $n \ge 19$, $m \ge 8e^{4\gamma}$, and $c \le \frac{n}{4}$. Thus,

$$\begin{aligned}\frac{9}{8}\left(\frac{e}{\sqrt{2\pi}}(c-1)^{1.5}\left(1-\frac{(c-1)}{n}\right)^{n-c+1.5}\right)^{\frac{m-1}{32e^{4\gamma}}} &\le \frac{9}{8}\left(\frac{2e}{\sqrt{\pi}}\left(1-\frac{2}{n}\right)^{n-1.5}\right)^{\frac{m-1}{32e^{4\gamma}}}\\ &\le \frac{9}{8}(1.27)^{\frac{m-1}{32e^{4\gamma}}}.\end{aligned}$$

Upper bounding the middle terms in the summation. It remains to upper bound the summation across $s \in \left[\frac{n}{4e^{4\gamma}}, \frac{3n}{4}\right]$. We have for $\alpha = \frac{s}{n}$,

$$b_s = \left(\frac{((1-\alpha)n)!(\alpha n)!}{(n-1)!(1-\alpha)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}}.$$

By Sterling's bound, we have

$$b_s \le \left(\frac{e^2}{\sqrt{2\pi}}\sqrt{n}(1-\alpha)^{2.5-c+(1-\alpha)n}\alpha^{\alpha n+\frac{1}{2}}\right)^{\frac{m-1}{32e^{4\gamma}}} \le \left(\frac{e^2\sqrt{n}}{\sqrt{2\pi}}\alpha^{\alpha n}\right)^{\frac{m-1}{32e^{4\gamma}}}.$$

Since there are at most n such terms b_s, then

$$\sum_{s=k+1}^{\lceil 3n/4\rceil-1} b_s \le n\left(\frac{e^2\sqrt{n}}{\sqrt{2\pi}}\left(\frac{3}{4}\right)^{\frac{3n}{4}}\right)^{\frac{m-1}{32e^{4\gamma}}} \le 2\left(en\left(\frac{3}{4}\right)^{\frac{3n}{4}}\right)^{\frac{m-1}{32e^{4\gamma}}} \le 2.$$

Putting things together. Combining the upper bounds across the three summations, we have

$$\sum_{s=1}^{k}\binom{n}{s}^{1-m} e^{2sm\gamma}\left(\frac{n^{c-1}}{(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}} + \sum_{s=k+1}^{n-c+1}\left(\frac{(n-s)!s!n^{c-1}}{n!(n-s)^{c-2}}\right)^{\frac{m-1}{32e^{4\gamma}}}$$

$$\leq \frac{26}{25} e^{m\gamma} e^{\frac{m-1}{32e^{4\gamma}}} + 2 + \frac{9}{8}(1.27)^{\frac{m-1}{32e^{4\gamma}}}$$
$$\leq 2e^{\frac{m-1}{32e^{4\gamma}}} \cdot e^{m\gamma} \leq 2e^{\frac{m-1}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)}.$$

Therefore, we have

$$p(n,c) \leq \frac{2^{c-1}}{c!} \left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)(c-1)},$$

as desired. □

We now upper bound the expected value of $\mathbb{E}\left[q^{C(G)}\right]$ for the purposes of upper bounding the right hand side of Lemma 4.6.

Lemma 4.12. *Let $n \geq 19$, $m \geq 8e^{4\gamma}$, and $q \leq \left(\frac{n}{e}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(1-m)}$. Let G be the graph on n vertices formed a random instantiation of the split-and-mix protocol $\mathcal{P}_{m,n}$ with m messages for each of n players, using a γ-imperfect shuffler $\mathcal{S}$. That is, let G have an edge between i and j if and only if player i passes one of their m messages to player j. Then*

$$\mathbb{E}\left[q^{C(G)}\right] \leq q + 3q^2 e^{2\gamma(m-1)} \left(\frac{e}{n}\right)^{\frac{m-1}{32e^{4\gamma}}}.$$

Proof. By Lemma 4.11, we have

$$p(n,c) \leq \frac{2^{c-1}}{c!} \left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)(c-1)}.$$

Taking the expectation, we have

$$\mathbb{E}\left[q^{C(G)}\right] \leq \sum_{c=1}^{n} q^c \frac{2^{c-1}}{c!} \left(\frac{e}{n}\right)^{\frac{(m-1)(c-1)}{32e^{4\gamma}}} \cdot e^{2\gamma(m-1)(c-1)}.$$

Since term in the summand after the second term is at most $\frac{2q}{3}\left(\frac{e}{n}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(m-1)}$ times the previous term in the summand, then

$$\mathbb{E}\left[q^{C(G)}\right] \leq q + q^2 e^{2\gamma(m-1)} \left(\frac{e}{n}\right)^{\frac{m-1}{32e^{4\gamma}}} \sum_{i=0}^{\infty} \left(\frac{2q}{3}\left(\frac{e}{n}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(m-1)}\right)^i.$$

Since $q \leq \left(\frac{n}{e}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(1-m)}$ by assumption, then

$$\mathbb{E}\left[q^{C(G)}\right] \leq q + 3q^2 e^{2\gamma(m-1)} \left(\frac{e}{n}\right)^{\frac{m-1}{32e^{4\gamma}}}.$$

□

We now analyze the statistical security of the split-and-mix protocol.

Lemma 4.13. *Let $n \geq 19$, $m \geq 8e^{4\gamma}$, and $q \leq \left(\frac{n}{e}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(1-m)}$. Then we have worst-case statistical security with parameter*

$$\sigma \leq (m-1)\left(\frac{\log n - \log e}{64e^{4\gamma}} - 2\gamma \log e\right) - 3\log(3q),$$

Proof. By Lemma 4.5 and Lemma 4.6, we have

$$\mathbb{E}_{\vec{\mathsf{X}},\vec{\mathsf{X}}'}[\mathrm{TVD}(\mathcal{P}_{m,n}(\vec{\mathsf{X}}),\mathcal{P}_{m,n}(\vec{\mathsf{X}}'))] \le \sqrt{q^{mn-1}\mathbb{E}\left[q^{C(G)-mn}\right]-1},$$

where $C(G)$ is the communication graph for the shuffle $\mathcal{S}^{-1}\circ\mathcal{S}'$. By Lemma 4.7 and the fact that $\mathcal{S}'$ is a γ-imperfect shuffler, we have that $\mathcal{S}^{-1}\circ\mathcal{S}'$ is also a γ-imperfect shuffler and thus it suffices to upper bound $\mathbb{E}\left[q^{C(G)-mn}\right]$ where $C(G)$ is the communication graph for an arbitrary γ-imperfect shuffler $\mathcal{S}$. Therefore by Lemma 4.12, we have average case statistical security less than or equal to

$$2^{-\sigma} \ge \sqrt{3q^3 e^{2\gamma(m-1)}\left(\frac{e}{n}\right)^{\frac{m-1}{32e^{4\gamma}}}},$$

which holds for

$$\sigma \le (m-1)\left(\frac{\log n - \log e}{64e^{4\gamma}} - 2\gamma\log e\right) - 3\log(3q).$$

The claim then follows by the reduction of worst-case input to average-case input by Lemma 4.2. □

Now it can be verified that by restricting $\gamma \le \frac{\log\log n}{80}$, then we have both $728e^{4\gamma} \le \log n$ and $\lceil 2n^{3/2}\rceil \le \left(\frac{n}{e}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(1-m)}$. These conditions imply that 1) $\left(\frac{\log n - \log e}{64e^{4\gamma}} - 2\gamma\log e\right) = \mathcal{O}\left(\frac{\log n}{e^{4y}}\right)$, so that the parameter σ has a non-empty range in the statement of Lemma 4.13, and 2) $q = \lceil 2n^{3/2}\rceil$ satisfies $q \le \left(\frac{n}{e}\right)^{\frac{(m-1)}{32e^{4\gamma}}} e^{2\gamma(1-m)}$ in the statement of Lemma 4.13. As a corollary, we obtain the following guarantees for worst-case statistical security:

Theorem 3.2. *Let $n \ge 19$ and $\gamma \le \frac{\log\log n}{80}$ be a distortion parameter. For worst-case statistical security with parameter σ, it suffices to use*

$$m = \mathcal{O}\left(e^{4\gamma} + \frac{e^{4\gamma}(\sigma + \log n)}{\log n}\right)$$

messages. Each message uses $\mathcal{O}(\log q)$ bits, for $q = \lceil 2n^{3/2}\rceil$.

5 Conclusion and Discussion

In this work, we introduce the imperfect shuffle DP model, as a means of abstracting out real-world scenarios that prevent perfect shuffling. We also give a real summation protocol with nearly optimal error and small communication complexity. The protocol, which is based on the split-and-mix protocol [IKOS06], is similar to that of the (perfect) shuffle model [BBGN20, GMPV20], while the main challenge comes in the analysis. Although we overcome this hurdle for this particular protocol, our techniques are quite specific. Therefore, an interesting open question is whether there is a general theorem that transfer the privacy guarantee in the perfect shuffle model to that in the imperfect shuffle model, possibly with some loss in the privacy parameters.

References

[Abo18] John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018. 1

[ACG$^+$16] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016. 1

[ASY$^+$18] Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 7575–7586, 2018. 1

[BBGN19a] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially private summation with multi-message shuffling. *CoRR*, abs/1906.09116, 2019. 4, 6, 8, 10, 27

[BBGN19b] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Proceedings, Part II*, pages 638–667, 2019. 4

[BBGN20] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 657–676, 2020. 2, 4, 5, 10, 11, 12, 13, 18, 21, 23

[BDKU20] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan R. Ullman. Coinpress: Practical private mean and covariance estimation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020. 1

[BEM$^+$17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459, 2017. 2

[BKM$^+$20] Borja Balle, Peter Kairouz, Brendan McMahan, Om Dipakbhai Thakkar, and Abhradeep Thakurta. Privacy amplification via random check-ins. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 7

[BNO08] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Proceedings*, pages 451–468, 2008. 2

[BST14] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 464–473, 2014. 1

[CSS12] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *Algorithms - ESA - 20th Annual European Symposium, Proceedings*, pages 277–288, 2012. 2

[CSU$^+$19] Albert Cheu, Adam D. Smith, Jonathan R. Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques,, Proceedings, Part I*, pages 375–403, 2019. 2, 4

[CWH20] Xiangyi Chen, Zhiwei Steven Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 1

[CY23] Albert Cheu and Chao Yan. Necessary conditions in multi-server differential privacy. In *14th Innovations in Theoretical Computer Science Conference, ITCS*, pages 36:1–36:21, 2023. 7

[DKM$^+$06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006. 1

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 3571–3580, 2017. 1

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings*, pages 265–284, 2006. 1, 2

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. 6

[EFM$^+$19] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2468–2479, 2019. 2, 4

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1054–1067, 2014. 1

[FMT21] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 954–964, 2021. 4, 7

[FMT23] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Stronger privacy amplification by shuffling for renyi and approximate differential privacy. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 4966–4981, 2023. 7

[GDD$^+$21] Antonious M. Girgis, Deepesh Data, Suhas N. Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE J. Sel. Areas Inf. Theory*, 2(1):464–478, 2021. 1

[GKLX22] S. Dov Gordon, Jonathan Katz, Mingyu Liang, and Jiayu Xu. Spreading the privacy blanket: - differentially oblivious shuffling for differential privacy. In *Applied Cryptography and Network Security - 20th International Conference, ACNS, Proceedings*, pages 501–520, 2022. 7

[GKM$^+$21] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, pages 3692–3701, 2021. 2, 4

[GMPV20] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part II*, pages 798–827, 2020. 2, 4, 23

[Gre16] Andy Greenberg. Apple's "differential privacy" is about collecting your data – but not your data, June 2016. 1

[GRS12] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM J. Comput.*, 41(6):1673–1693, 2012. 2

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS) USA, Proceedings*, pages 239–248, 2006. 5, 8, 10, 23

[KLN$^+$11] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011. 2

[KMA$^+$21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. 1

[KMY$^+$16] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016. 1

[KTH$^+$19] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. Privatesql: A differentially private SQL query engine. *Proc. VLDB Endow.*, 12(11):1371–1384, 2019. 1

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP*, pages 245–248, 2013. 1

[Sha14] Stephen Shankland. How google tricks itself to protect chrome user privacy. *CNET, October*, 2014. 1

[SK18] Uri Stemmer and Haim Kaplan. Differentially private k-means with constant multiplicative error. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 5436–5446, 2018. 1

[Ste21] Uri Stemmer. Locally private k-means clustering. *J. Mach. Learn. Res.*, 22:176:1–176:30, 2021. 1

[SW21] Elaine Shi and Ke Wu. Non-interactive anonymous router. In *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III*, pages 489–520, 2021. 7

[SYKM17] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 3329–3337, 2017. 1

[TW23] Martin Thomson and Christopher A. Wood. Oblivious http, 2023. 3

[War65] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. 2

[WZL$^+$20] Royce J. Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. Differentially private SQL with bounded user contribution. *Proc. Priv. Enhancing Technol.*, 2020(2):230–250, 2020. 1

[ZS22] Mingxun Zhou and Elaine Shi. The power of the differentially oblivious shuffle in distributed privacy mechanisms. *IACR Cryptol. ePrint Arch.*, page 177, 2022. 4, 7

[ZSCM22] Mingxun Zhou, Elaine Shi, T.-H. Hubert Chan, and Shir Maimon. A theory of composition for differential obliviousness. *IACR Cryptol. ePrint Arch.*, page 1357, 2022. 4, 7

A Additional Proofs

Lemma 3.1. *[Lemma 4.1 in [BBGN19a]] Given a σ-secure protocol Ξ in the γ-I-shuffle model for n-party private summation on $\mathbb{Z}_q$ such that each player sends $f(n, q, \sigma)$ bits of messages, there exists a $(\varepsilon, (1+e^{\varepsilon})2^{-\sigma-1})$-differentially private protocol in the γ-I-shuffle model for n-party private summation* *on real numbers with expected absolute error $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ such that each player sends $f(n, O(n^{3/2}), \sigma)$ bits of messages.*

Proof. Given Ξ, we construct a protocol $\mathcal{P}$ on a field of size $q := \lceil 2n^{3/2} \rceil$ as follows. Each input $x_i \in [0,1]$ is rounded to a value y_i precision $p = \sqrt{n}$, so that $y_i = \lfloor x_i p \rfloor + \mathsf{Ber}(x_i p - \lfloor x_i p \rfloor)$. We then add Polya noise to each term, so that $z_i = y_i + \mathsf{Polya}\left(\frac{1}{n}, e^{-\varepsilon/p}\right) - \mathsf{Polya}\left(\frac{1}{n}, e^{-\varepsilon/p}\right)$. The protocol $\mathcal{P}$ then runs Ξ on the inputs $z_1, \ldots, z_n$ to achieve a sum Z. The protocol $\mathcal{P}$ then decodes Y by outputting $\widetilde{X} = \frac{Z}{p}$ if $Z \le \frac{3np}{2}$ and by outputting $\widetilde{X} = \frac{Z-q}{p}$ otherwise if $Z > \frac{3np}{2}$.

Upper bounding the expected error. There are multiple sources of error. The first source of error comes from the randomized rounding that produces the values $y_1, \ldots, y_n$ from $x_1, \ldots, x_n$. By Lemma 1.9, we have that

$$\mathbb{E}\left[\left(\sum_{i=1}^n \left(x_i - \frac{y_i}{p}\right)\right)^2\right] \le \frac{n}{4p^2}.$$

Since $p = \sqrt{n}$, then by Markov's inequality,

$$\mathbf{Pr}\left[\left(\sum_{i=1}^n \left(x_i - \frac{y_i}{p}\right)\right)^2 \ge \frac{n^2}{16}\right] \le \frac{4}{n^2}.$$

The second source of error comes from the noise added to the variables $y_1, \ldots, y_n$ to obtain the values $z_1, \ldots, z_n$. Beyond the error incurred from the randomized rounding, $\widetilde{X}$ has additional error distributed according to the discrete Laplacian distribution unless the total noise added has magnitude larger than $\frac{n}{2}$, which could potentially cause an additive $\mathcal{O}\left(n^2\right)$ squared error due to incorrect decoding of $\widetilde{X}$ from Z. By Fact 1.8, the total noise added to the variables y_i is a random variable drawn from the discrete Laplace distribution, i.e.,

$$\sum_{i=1}^n (z_i - y_i) \sim \mathsf{DLap}\left(e^{-\varepsilon/p}\right).$$

By the distribution of the discrete Laplacian, the noise added by the discrete Laplacian distribution is more than $\frac{n}{4}$ with probability $\mathcal{O}\left(e^{-\varepsilon n/4}\right)$. Therefore, the expected mean squared error in this protocol is at most $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right) + \frac{4}{n^2} \cdot \mathcal{O}\left(n^2\right) + \mathcal{O}\left(e^{-\varepsilon n/4}\right) \cdot \mathcal{O}\left(n^2\right) = \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$. By Jensen's inequality, the expected absolute error in this protocol is at most $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$.

Privacy considerations. Consider the protocol Ξ' with input $w_1 = Z$ and $w_2 = \ldots = w_n = 0$, where $Z = z_1 + \ldots + z_n$, as defined above. Note that by Fact 1.8, we have $w_1 = Z \sim \sum_{y=1}^n y_i + \mathsf{DLap}\left(e^{-\varepsilon/p}\right)$. Since the sensitivity of $\sum_{y=1}^n y_i$ is p, then Z is $(\varepsilon, 0)$-differentially private. Therefore, by post-processing, Ξ' is also $(\varepsilon, 0)$-differentially private.

Moreover, we can upper bound the statistical distance between Ξ and Ξ' by $2^{-\sigma}$ by a coupling argument. Specifically, by coupling the noise added to x_i by both Ξ and Ξ', the protocols Ξ and Ξ' will output the same sum given this coupled randomness outside of the $2^{-\sigma}$ difference induced by the σ-secure property. □