Concurrent Composition of Differential Privacy

Salil Vadhan* and Tianhao Wang**

[1] Harvard University
salil_vadhan@harvard.edu
[2] Princeton University
tianhaowang@princeton.edu

Abstract. We initiate a study of the composition properties of *interactive* differentially private mechanisms. An interactive differentially private mechanism is an algorithm that allows an analyst to adaptively ask queries about a sensitive dataset, with the property that an adversarial analyst's view of the interaction is approximately the same regardless of whether or not any individual's data is in the dataset. Previous studies of composition of differential privacy have focused on non-interactive algorithms, but interactive mechanisms are needed to capture many of the intended applications of differential privacy and a number of the important differentially private primitives.

We focus on *concurrent composition*, where an adversary can arbitrarily interleave its queries to several differentially private mechanisms, which may be feasible when differentially private query systems are deployed in practice. We prove that when the interactive mechanisms being composed are *pure* differentially private, their concurrent composition achieves privacy parameters (with respect to pure or approximate differential privacy) that match the (optimal) composition theorem for noninteractive differential privacy. We also prove a composition theorem for interactive mechanisms that satisfy approximate differential privacy. That bound is weaker than even the basic (suboptimal) composition theorem for noninteractive differential privacy, and we leave closing the gap as a direction for future research, along with understanding concurrent composition for other variants of differential privacy.

Keywords: Interactive Differential Privacy · Concurrent Composition Theorem.

1 Introduction

1.1 Differential Privacy

Differential privacy is a framework for protecting privacy when performing statistical releases on a dataset with sensitive information about individuals. (See the

* Supported by NSF grant CNS-1565387, a grant from the Sloan Foundation, and a Simons Investigator Award.

** Work done while at Harvard University.

surveys [10,23].) Specifically, for a differentially private mechanism, the probability distribution of the mechanism's outputs of a dataset should be nearly identical to the distribution of its outputs on the same dataset with any single individual's data replaced. To formalize this, we call two datasets x, x', each multisets over a data universe $\mathcal{X}$, *adjacent* if one can be obtained from the other by adding or removing a single element of $\mathcal{X}$.

Definition 1.1 (Differential Privacy [8]). *For $\varepsilon, \delta \geq 0$, a randomized algorithm $\mathcal{M} : \mathrm{MultiSets}(\mathcal{X}) \to \mathcal{Y}$ is* (ε, δ)-differentially private *if for every pair of adjacent datasets $x, x' \in \mathrm{MultiSets}(\mathcal{X})$, we have:*

$$\forall\ T \subseteq \mathcal{Y}\ \ \Pr[\mathcal{M}(x) \in T] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x') \in T] + \delta \tag{1}$$

where the randomness is over the coin flips of the algorithm $\mathcal{M}$.

In the practice of differential privacy, we generally view ε as "privacy-loss budget" that is small but non-negligible (e.g. $\varepsilon = 0.1$), and we view δ as cryptographically negligible (e.g. $\delta = 2^{-60}$). We refer to the case where $\delta = 0$ as *pure differential privacy*, and the case where $\delta > 0$ as *approximate differential privacy.*

1.2 Composition of Differential Privacy

A crucial property of differential privacy is its behavior under composition. If we run multiple distinct differentially private algorithms on the same dataset, the resulting composed algorithm is also differentially private, with some degradation in the privacy parameters (ε, δ). This property is especially important and useful since in practice we rarely want to release only a single statistic about a dataset. Releasing many statistics may require running multiple differentially private algorithms on the same database. Composition is also a very useful tool in algorithm design. In many cases, new differentially private algorithms are created by combining several simpler algorithms. The composition theorems help us analyze the privacy properties of algorithms designed in this way.

Formally, let $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_{k-1}$ be differentially private mechanisms, we define the composition of these mechanisms by independently executing them. Specifically, we define $\mathcal{M} = \mathrm{Comp}(\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_{k-1})$ as follows:

$$\mathcal{M}(x) = (\mathcal{M}_0(x), \ldots, \mathcal{M}_{k-1}(x))$$

where each $\mathcal{M}_i$ is run with independent coin tosses. For example, this is how we might obtain a mechanism answering a k-tuple of queries.

A handful of composition theorems already exist in the literature. The Basic Composition Theorem says that the privacy degrades at most linearly with the number of mechanisms executed.

Theorem 1.2 (Basic Composition [7]). *For every $\varepsilon \geq 0$, $\delta \in [0, 1]$, if $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each (ε, δ)-differentially private mechanisms, then their composition $\mathrm{Comp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $(k\varepsilon, k\delta)$-differentially private.*

Theorem 1.2 shows the global privacy degradation is linear in the number of mechanisms in the composition. However, if we are willing to tolerate an increase in the δ term, the privacy parameter ε only needs to degrade proportionally to $\sqrt{k}$:

Theorem 1.3 (Advanced Composition [12]). *For all $\varepsilon \geq 0$, $\delta \in [0,1]$, if $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each (ε, δ)-differentially private mechanisms and $k < 1/\varepsilon^2$, then for all $\delta' \in (0, 1/2)$, the composition $(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $\left(O\left(\sqrt{k \log(1/\delta')}\right) \cdot \varepsilon, k\delta + \delta'\right)$-differentially private.*

Theorem 1.3 is an improvement if $\delta' = 2^{-o(k)}$. However, despite giving an asymptotically correct upper bound for the global privacy parameter, Theorem 1.3 is not exact. Kairouz, Oh, and Viswanath [18] shows how to compute the optimal bound for composing k mechanisms where all of them are (ε, δ)-differentially private. Murtagh and Vadhan [21] further extends the optimal composition for the more general case where the privacy parameters may differ for each algorithm in the composition:

Theorem 1.4 (Optimal Composition [18,21]). *If $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each $(\varepsilon_i, \delta_i)$-differentially private, then given any $\delta_g > 0$, $\mathrm{Comp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $(\varepsilon_g, \delta_g)$-differentially private for the the least value of $\varepsilon_g \geq 0$ such that*

$$\frac{1}{\prod_{i=0}^{k-1}\left(1+\mathrm{e}^{\varepsilon_i}\right)} \sum_{S \subseteq \{0,\ldots,k-1\}} \max\left\{\mathrm{e}^{\sum_{i \in S} \varepsilon_i} - \mathrm{e}^{\varepsilon_g} \cdot \mathrm{e}^{\sum_{i \notin S} \varepsilon_i}, 0\right\} \leq 1 - \frac{1-\delta_g}{\prod_{i=0}^{k-1}(1-\delta_i)}$$

A special case when all $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are (ε, δ)-differentially private, then privacy parameter is upper bounded by the least value of $\varepsilon_g \geq 0$ such that

$$\frac{1}{(1+\mathrm{e}^{\varepsilon})^k} \sum_{i=0}^{k} \binom{k}{i} \max\left\{\mathrm{e}^{i\varepsilon} - \mathrm{e}^{\varepsilon_g} \cdot \mathrm{e}^{(k-i)\varepsilon}, 0\right\} \leq 1 - \frac{1-\delta_g}{(1-\delta)^k}$$

1.3 Interactive Differential Privacy

The standard treatment of differential privacy, as captured by Definition 1.1, refers to a *noninteractive* algorithm $\mathcal{M}$ that takes a dataset x as input and produces a statistical release $\mathcal{M}(x)$, or a batch by taking $\mathcal{M} = \mathrm{Comp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$. However, in many of the motivating applications of differential privacy, we don't want to perform all of our releases in one shot, but rather allow analysts to make adaptive queries to a dataset. Thus, we should view the mechanism $\mathcal{M}$ as a party in a two-party protocol, interacting with a (possibly adversarial) analyst.

To formalize the concept of interactive DP, we recall one of the standard formalizations of an interactive protocol between two parties A and B. We do this by viewing each party as a function, taking its private input, all messages it has received, and the party's random coins, to the party's next message to be sent out.

Definition 1.5 (Interactive protocols). *An* interactive protocol (A, B) *is any pair of functions. The interaction between A with input x_A and B with input x_B is the following random process (denoted $(A(x_A), B(x_B))$):*

1. *Uniformly choose random coins r_A and r_B (binary strings) for A and B, respectively.*
2. *Repeat the following for $i = 0, 1, \ldots$:*
   (a) *If i is even, let $m_i = A\left(x_A, m_1, m_3, \ldots, m_{i-1}; r_A\right)$.*
   (b) *If i is odd, let $m_i = B\left(x_B, m_0, m_2, \ldots, m_{i-1}; r_B\right)$.*
   (c) *If $m_{i-1} = \texttt{halt}$, then exit loop.*

We further define the view of a party in an interactive protocol to capture everything the party "sees" during the execution:

Definition 1.6 (View of a party in an interactive protocol). *Let (A, B) be an interactive protocol. Let r_A and r_B be the random coins for A and B, respectively. A's view of $(A(x_A; r_A), B(x_B; r_B))$ is the tuple $\texttt{View}_A\langle A(x_A; r_A), B(x_B; r_B)\rangle = (r_A, x_A, m_1, m_3, \ldots)$ consisting of all the messages received by A in the execution of the protocol together with the private input x_A and random coins r_A. If we drop the random coins r_A and/or r_B, $\texttt{View}_A\langle A(x_A), B(x_B)\rangle$ becomes a random variable. B's view of $(A(x_A), B(x_B))$ is defined symmetrically.*

In our case, A is the adversary and B is the mechanism whose input is usually a database x. Since A does not have an input in our case, we will denote the interactive protocol as $(A, B(x))$ for the ease of notation. Since we will only be interested in A's view and A does not have an input, we will drop the subscript and write A's view as $\texttt{View}\langle A, B(x)\rangle$.

Now we are ready to define the interactive differential privacy as a type of interactive protocol between an adversary (without any computational limitations) and an interactive mechanism of special properties.

Definition 1.7 (Interactive Differential Privacy). *A randomized algorithm $\mathcal{M}$ is an* (ε, δ)-differentially private interactive mechanism *if for every pair of adjacent datasets $x, x' \in \text{MultiSets}(\mathcal{X})$, for every adversary algorithm $\mathcal{A}$ we have:*

$$\begin{aligned} &\forall T \subseteq \text{Range}\left(\texttt{View}\langle \mathcal{A}, \mathcal{M}(\cdot)\rangle\right), \\ &\Pr\left[\texttt{View}\langle \mathcal{A}, \mathcal{M}(x)\rangle \in T\right] \leq e^{\varepsilon} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathcal{M}(x')\rangle \in T\right] + \delta \end{aligned} \tag{2}$$

where the randomness is over the coin flips of both the algorithm $\mathcal{M}$ and the adversary $\mathcal{A}$.

In addition to being the "right" modelling for many applications of differential privacy, interactive differential privacy also captures the full power of fundamental DP mechanisms such as the Sparse Vector Technique [9,22] and Private Multiplicative Weights [17], which are in turn useful in the design of other DP algorithms (which can use these mechanisms as subroutines and issue adaptive queries to them). Interactive DP was also chosen as the basic abstraction in the programming framework for the new open-source software project OpenDP [14], which was our motivation for this research.

Despite being such a natural and useful notion, interactive DP has not been systematically studied in its own right. It has been implicitly studied in the context of distributed forms of DP, starting with [1], where the sensitive dataset is split amongst several parties, who execute a multiparty protocol to estimate a joint function of their data, while each party ensures that their portion of the dataset has the protections of DP against the other parties. Indeed, in an m-party protocol, requiring DP against malicious coalitions of size $m-1$ is equivalent to requiring that each party's strategy is an interactive DP mechanism in the sense of Definition 1.7. An extreme case of this is the *local model* of DP, where each party holds a single data item in $\mathcal{X}$ representing data about themselves [19]. There been extensive research about the power of interactivity in local DP; see [5] and the references therein. In contrast to these distributed models, in Definition 1.7 we are concerned with the *centralized DP* scenario where only one party ($\mathcal{M}$) holds sensitive data, and how an adversarial data analyst ($\mathcal{A}$) may exploit adaptive queries to extract information about the data subjects.

Some of the aforementioned composition theorems for noninteractive DP, such as in [12,21], are framed in terms of an adaptive "composition game" where an adversary can adaptively select the mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$, and thus the resulting composition $\mathrm{Comp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ can be viewed as an interactive mechanism, but the results are not framed in terms of a general definition of Interactive DP. In particular, the mechanisms $\mathcal{M}_i$ being composed are restricted to be noninteractive in the statements and proofs of these theorems.

1.4 Our Contributions

In this paper, we initiate a study of the composition of interactive DP mechanisms. Like in the context of cryptographic protocols, there are several different forms of composition we can consider. The simplest is *sequential composition*, where all of the queries to $\mathcal{M}_{i-1}$ must be completed before any queries are issued to $\mathcal{M}_i$. It is straightforward to extend the proofs of the noninteractive DP composition theorems to handle sequential composition of interactive DP mechanisms; in particular the Optimal Composition Theorem (Theorem 1.4) extends to this case. (Details omitted.)

Thus, we turn to *concurrent composition*, where an adversary can arbitrarily interleave its queries to the k mechanisms. Although the mechanisms use independent randomness, the adversary may create correlations between the executions by coordinating its actions; in particular, its queries in one execution may also depend on messages it received in other executions. Concurrent composability is important for the deployment of interactive DP in practice, as one or more organizations may set up multiple DP query systems on datasets that refer to some of the same individuals, and we would not want the privacy of those individuals to be violated by an adversary that can concurrently access those systems. Concurrent composability may also be useful in the design of DP algorithms; for example, one might design a DP machine learning algorithm that uses adaptive and interleaved queries to two instantiations of an interactive DP mechanism like the Sparse Vector Technique [9,22].

Although the concurrent composition for the case of differential privacy has not been explored before, it has been studied extensively for many primitives in cryptography, and it is often much more subtle than the sequential composition. (See the surveys [4,15].) For example, standard zero-knowledge protocols are no longer zero-knowledge when a single prover is involved in multiple, simultaneous zero-knowledge proofs with one or multiple verifiers [13,16].

We use $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ to denote the concurrent composition of interactive mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$. (See Section 2 for a formal definition.)

Our first result is roughly an analogue of the Basic Composition Theorem.

Theorem 1.8. *If interactive mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each (ε, δ)-differentially private, then their concurrent composition $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $\left(k \cdot \varepsilon, \frac{e^{k\varepsilon}-1}{e^{\varepsilon}-1} \cdot \delta\right)$-differentially private.*

More generally, if interactive mechanism $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private for $i = 0, \ldots, k-1$, then the concurrent composition $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $(\varepsilon_g, \delta_g)$-differentially private, where

$$\varepsilon_g = \sum_{i=0}^{k-1} \varepsilon_i, \text{ and}$$

$$\delta_g = \sum_{i=0}^{k-1} e^{\sum_{j=0}^{i-1} \varepsilon_j} \cdot \delta_i \leq e^{\varepsilon_g} \cdot \sum_{i=0}^{k-1} \delta_i.$$

Just like in the Basic Composition Theorem for noninteractive DP (Theorem 1.2), the privacy-loss parameters ε_i just sum up. However, the bound on δ_g is worse by a factor of at most e^{ε_g}. In the typical setting where we want to enforce a global privacy loss of $\varepsilon_g = O(1)$, this is only a constant-factor loss compared to the Basic Composition Theorem, but that constant can be important in practice. Note that expression for δ_g depends on the ordering of the k mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$, so one can optimize it further by taking a permutation of the mechanisms that minimizes δ_g.

The proof of Theorem 1.8 is by a standard hybrid argument. We compare the distributions of $H_0 = \mathtt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x), \mathcal{M}_1(x), \ldots, \mathcal{M}_{k-1}(x))\rangle$ and $H_k = \mathtt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \mathcal{M}_1(x'), \ldots, \mathcal{M}_{k-1}(x'))\rangle$ on adjacent datasets x, x' by changing x to x' for one mechanism at a time, so that H_{i-1} and H_i differ only on the input to $\mathcal{M}_{i-1}$. To relate H_{i-1} and H_i we consider an adversary strategy $\mathcal{A}_i$ that emulates $\mathcal{A}$'s interaction with $\mathcal{M}_{i-1}$, while internally simulating all of the other $\mathcal{M}_j$'s. Applying a "triangle inequality" to the distance notion given in Requirement (2) yields the result. This proof is very similar to the proof of the "group privacy" property of (noninteractive) differential privacy, where (ε, δ)-DP for datasets that differ on one record implies $\left(k \cdot \varepsilon, \frac{e^{k\varepsilon}-1}{e^{\varepsilon}-1} \cdot \delta\right)$ for datasets that differ on k records.

Next we show that the Advanced and Optimal Composition Theorems (Theorems 1.3 and 1.4) for noninteractive DP extend to interactive DP, provided that the mechanisms $\mathcal{M}_i$ being composed satisfy pure DP (i.e. $\delta_i = 0$). Note that the final composed mechanism $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ can be approximate DP, by taking $\delta_g = \delta' > 0$, and thereby allowing for a privacy loss ε_g that grows linearly in $\sqrt{k}$ rather than k.

We do this by extending the main proof technique of [18,21] to interactive DP mechanisms. Specifically, we reduce the analysis of interactive $(\varepsilon, 0)$-DP mechanisms to that of analyzing the following simple "randomized response" mechanism:

Definition 1.9 ([25,8]). *For $\varepsilon > 0$, define a randomized noninteractive algorithm $\mathrm{RR}_\varepsilon : \{0,1\} \to \{0,1\}$ as follows:*

$$\mathrm{RR}_\varepsilon(b) = \begin{cases} b & w.p.\ \frac{e^\varepsilon}{1+e^\varepsilon} \\ \neg b & w.p.\ \frac{1}{1+e^\varepsilon}. \end{cases}$$

Note that RR_ε is a noninteractive $(\varepsilon, 0)$-DP mechanism. We show that every interactive $(\varepsilon, 0)$-DP mechanism can be, in some sense, simulated from RR_ε:

Theorem 1.10. *Suppose that $\mathcal{M}$ is an interactive $(\varepsilon, 0)$-differentially private mechanism. Then for every pair of adjacent datasets x_0, x_1 there exists an interactive mechanism T s.t. for every adversary $\mathcal{A}$ and every $b \in \{0,1\}$ we have*

$$\mathtt{View}(\mathcal{A}, \mathcal{M}(x_b)) \equiv \mathtt{View}(\mathcal{A}, T(\mathrm{RR}_\varepsilon(b)))$$

Here T is an interactive mechanism that depends on $\mathcal{M}$ as well as a fixed pair of adjacent datasets x_0 and x_1. It receives a single bit as an output of $\mathrm{RR}_\varepsilon(b)$, and then interacts with the adversary $\mathcal{A}$ just like $\mathcal{M}$ would. Kairouz, Oh, and Viswanath [18] proved Theorem 1.10 result for the case that $\mathcal{M}$ and T are noninteractive. The interactive case is more involved because we need a single T that works for all adversary strategies $\mathcal{A}$. (If we allow T to depend on the adversary strategy $\mathcal{A}$, then the result would readily follow from that of [18], but this would not suffice for our application to concurrent composition.)

Given the Theorem 1.10, to analyze $\mathrm{ConComp}(\mathcal{M}_0(x_b), \ldots, \mathcal{M}_{k-1}(x_b))$ on $b = 0$ vs. $b = 1$, it suffices to analyze $\mathrm{ConComp}(T_0(\mathrm{RR}_{\varepsilon_0}(b)), \ldots, T_{k-1}(\mathrm{RR}_{\varepsilon_{k-1}}(b)))$. An adversary's view interacting with the latter concurrent composition can be simulated entirely from the output of $\mathrm{Comp}(\mathrm{RR}_{\varepsilon_0}(b), \ldots, \mathrm{RR}_{\varepsilon_{k-1}}(b))$, which is the composition of entirely noninteractive mechanisms. Thus, we conclude:

Corollary 1.11. *The Advanced and Optimal Composition Theorems (Theorems 1.3 and 1.4) extend to the concurrent composition of $(\varepsilon_i, \delta_i)$-interactive DP mechanisms $\mathcal{M}_i$ provided that $\delta_0 = \delta_1 = \cdots = \delta_{k-1} = 0$.*

We leave the question of whether or not the Advanced and/or Optimal Composition Theorems extend to the concurrent composition of approximate DP mechanisms (with $\delta_i > 0$) for future work. The Optimal Composition Theorem for noninteractive approximate DP (Theorem 1.4) is also proven by showing that any noninteractive (ε, δ)-DP mechanism can be simulated by an approximate-DP generalization of randomized response, $\mathrm{RR}_{(\varepsilon,\delta)}$, analogously to Theorem 1.10.

Based on computer experiments described in Section 6, we conjecture that such a simulation also exists for every approximate DP interactive mechanism, and the Optimal Composition Theorem should extend at least to 2-round interactive mechanisms in which all messages are 1 bit long.

Another interesting question for future work is analyzing concurrent composition for variants of differential privacy, such as Concentrated DP [11,3,2], Rényi DP [20], and Gaussian DP [6]. Some of these notions require bounds on Rényi divergences, e.g. that

$$D_\alpha(\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x)\rangle || \mathtt{View}\langle \mathcal{A}, \mathcal{M}(x')\rangle) \leq \rho,$$

for adjacent datasets x, x' and certain pairs (α, ρ). Here sequential composition can be argued using a chain rule for Rényi divergence:

$$D_\alpha((Y,Z)||(Y',Z')) \leq D_\alpha(Y||Y') + \sup_y D_\alpha(Z|_{Y=y}||Z'|_{Y'=y}). \tag{3}$$

Taking Y to be the view of the analyst interacting with $\mathcal{M}_0(x)$, Z to be the view of the analyst in a subsequent interaction with $\mathcal{M}_1(x)$, and Y' and Z' to be analogously defined with respect to an adjacent dataset x', we obtain the usual composition bound of $\rho_0 + \rho_1$ on the overall Rényi divergence of order α, where ρ_0 and ρ_1 are the privacy-loss parameters of the individual mechanisms. However, this argument fails for concurrent DP, since we can no longer assert the privacy properties of $\mathcal{M}_1$ conditioned on any possible value y of the adversary's view of the interaction with $\mathcal{M}_0$. Unfortunately, the Chain Rule (3) does not hold if we replace the supremum with an expectation, so a new proof strategy is needed (if the composition theorem remains true).

2 Definitions and Basic Properties

The formal definition of the concurrent composition of interactive protocols is provided here.

Definition 2.1 (Concurrent Composition of Interactive Protocols). *Let $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ be interactive mechanisms. We say $\mathcal{M} = \mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is the* concurrent composition *of mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ if $\mathcal{M}$ runs as follows:*

1. *Random coin tosses for $\mathcal{M}$ consist of $r = (r_0, \ldots, r_{k-1})$ where r_j are random coin tosses for $\mathcal{M}_j$.*
2. *Inputs for $\mathcal{M}$ consists of $x = (x_0, \ldots, x_{k-1})$ where x_j is private input for $\mathcal{M}_j$.*
3. *$\mathcal{M}(x, m_0, \ldots, m_{i-1}; r)$ is defined as follows:*
   (a) *Parse m_{i-1} as (q, j) where q is a query and $j \in [k]$. If m_{i-1} cannot be parsed correctly, output $\mathtt{halt}$.*
   (b) *Extract history $(m_0^j, \ldots, m_{t-1}^j)$ from $(m_0, \ldots, m_{i-1})$ where m_i^j are all of the queries to mechanism $\mathcal{M}_j$.*

(c) Output $\mathcal{M}_j(x_j, m_0^j, \ldots, m_{t-1}^j; r_j)$.

We are mainly interested in the case where all mechanisms operate on the same dataset, i.e., the private input for each $\mathcal{M}_i$ are all the same.

We show that to prove an interactive DP mechanism is (ε, δ)-differentially private, it suffices to consider all deterministic adversaries.

Lemma 2.2. *An interactive mechanism* $\mathcal{M}$ *is* (ε, δ)*-differentially private if and only if for every pair of adjacent datasets* x, x'*, for every* deterministic *adversary algorithm* $\mathcal{A}$*, for every possible output set* $T \subseteq \text{Range}\left(\mathtt{View}\langle \mathcal{A}, \mathcal{M}(\cdot)\rangle\right)$ *we have*

$$\Pr\left[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x)\rangle \in T\right] \leq e^{\varepsilon} \Pr\left[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x')\rangle \in T\right] + \delta \tag{4}$$

Proof. The necessity is immediately implied by the definition of interactive differential privacy. We prove the direction of sufficiency here. Assume that mechanism $\mathcal{M}$ satisfies (4) for every deterministic adversary. Suppose, for contradiction, that there exists a randomized adversary $\mathcal{A}$ and some output set T s.t.

$$\Pr\left[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x)\rangle \in T\right] > e^{\varepsilon} \Pr\left[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x')\rangle \in T\right] + \delta \tag{5}$$

Since the random coins of $\mathcal{A}$ and $\mathcal{M}$ are independently chosen, we have

$$\Pr\left[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x)\rangle \in T\right] = \mathbb{E}_{r_A}\left[\Pr_{r_{\mathcal{M}}}\left[\mathtt{View}\langle \mathcal{A}(r_A), \mathcal{M}(x; r_{\mathcal{M}})\rangle \in T\right]\right].$$

Therefore, there must exists at least one fixed r_A s.t.

$$\Pr\left[\mathtt{View}\langle \mathcal{A}(r_A), \mathcal{M}(x)\rangle \in T\right] > e^{\varepsilon} \Pr\left[\mathtt{View}\langle \mathcal{A}(r_A), \mathcal{M}(x')\rangle \in T\right] + \delta$$

otherwise 5 is impossible. Therefore, we can define a deterministic adversary $\mathcal{A}_{r_A} = \mathcal{A}(r_A)$. For set $T_{r_A} = \{(m_1, m_3, \ldots) : (r_A, m_1, m_3, \ldots) \in T\}$, since we have

$$\Pr\left[\mathtt{View}\langle \mathcal{A}(r_A), \mathcal{M}(x)\rangle \in T\right] = \Pr\left[\mathtt{View}\langle \mathcal{A}_{r_A}, \mathcal{M}(x)\rangle \in T_{r_A}\right]$$

we know that $\mathcal{A}_{r_A}$ is a counter example for our assumption, which leads to the conclusion.

For the convenience of the proof, we introduce a variant of concurrent composition of interactive protocols, which only accept queries in the exact order of $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$.

Definition 2.3 (Ordered Concurrent Composition of Interactive Protocols). *Let* $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ *be interactive mechanisms. We say* $\mathcal{M} = \text{ConComp}_{\mathtt{order}}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ *is the* ordered concurrent composition *of mechanisms* $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ *if* $\mathcal{M}(x)$ *runs as follows:*

1. *Random coin tosses and inputs for* $\mathcal{M}$ *are the same as* $\text{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$.
2. $\mathcal{M}(x, m_0, \ldots, m_{i-1}; r)$ *is defined as follows:*
   (a) Let $j = i \bmod k$*,* $t = \lfloor i/k \rfloor$.
   (b) Output $\mathcal{M}_j(x, m_j, m_{j+k}, \ldots, m_{j+t\cdot k}; r_j)$.

We also introduce a special kind of interactive mechanism, which ignores all query strings begin with 0.

Definition 2.4 (Null-query Extension). *Given an interactive mechanism $\mathcal{M}$, define its* null-query extension *$\mathcal{M}^{\emptyset}$ defined as follows: For any input message sequence m, $\mathcal{M}^{\emptyset}(x, m; r) = \mathcal{M}(x, m'; r)$ where $m' = (m'_1, \ldots, m'_k)$ such that $(1m'_1, \ldots, 1m'_k)$ is the subsequence of m consisting of all strings that begin with bit 1. That is, all messages that begin with 0 are "null queries" that are ignored. By convention, $\mathcal{M}(x, \lambda; r) = \perp$ where λ is an empty tuple.*

Now we show that in order to prove $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is (ε, δ)-differentially private, it suffices to prove a corresponding ordered concurrent composition is also (ε, δ)-differentially private. We use $X \equiv Y$ to denote that two random variables X and Y have the same distribution.

Lemma 2.5. $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ *is an (ε, δ)-differentially private interactive mechanism if the ordered concurrent composition of the null-query extensions of $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$, i.e.,*
$\mathrm{ConComp}_{\mathit{order}}(\mathcal{M}^{\emptyset}_1, \ldots, \mathcal{M}^{\emptyset}_k)$*, is an (ε, δ)-differentially private interactive mechanism.*

Proof. Suppose $\mathrm{ConComp}_{\mathtt{order}}\left(\mathcal{M}^{\emptyset}_0, \ldots, \mathcal{M}^{\emptyset}_{k-1}\right)$ is (ε, δ)-differentially private. For every adversary $\mathcal{A}$ interacting with $\mathrm{ConComp}\left(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}\right)$, we construct another adversary $\mathcal{A}'$ interacting with $\mathrm{ConComp}_{\mathtt{order}}(\mathcal{M}^{\emptyset}_0, \ldots, \mathcal{M}^{\emptyset}_{k-1})$ as follows: given any settings of coin tosses r, and any history $(q_0, a_0, \ldots, q_{i-1}, a_{i-1})$ between $\mathcal{A}$ and $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$,

1. Let $q_i = \mathcal{A}(a_0, \ldots, a_{i-1}; r)$.
2. Parse q_{i-1} as (q^*_{i-1}, s) where q^*_{i-1} is a query and $s \in \{0, \ldots, k-1\}$ the index of target mechanism. Parse q_i as (q^*_i, t) in a similar way.
3. Send the null query 0 to $\mathcal{M}^{\emptyset}_{(s+1) \bmod k}, \ldots, \mathcal{M}^{\emptyset}_{(t-1) \bmod k}$ in order.
4. Send $1q^*_i$ to $\mathcal{M}^{\emptyset}_t$.

Write $\mathcal{M} = \mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$, and $\mathcal{M}' = \mathrm{ConComp}_{\mathtt{order}}(\mathcal{M}^{\emptyset}_0, \ldots, \mathcal{M}^{\emptyset}_{k-1})$. For every query sequence q from $\mathcal{A}$, we have $\mathcal{M}(x, q; r) = \mathcal{M}'(x, q'; r)$ where q' is the sequence of queries that $\mathcal{A}'$ asks based on q (with '1' in front of every query in q and additional 0s). Therefore, for every $\mathcal{A}$ interact with $\mathcal{M}$, and for every dataset x we have

$$\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x) \rangle \equiv \mathtt{Post}(\mathtt{View}\langle \mathcal{A}', \mathcal{M}'(x) \rangle)$$

where $\mathtt{Post}$ refers to remove all repeated answers due to the null queries. This immediately leads to

$$\begin{aligned} & \Pr[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x) \rangle \in T] \\ & = \Pr[\mathtt{Post}(\mathtt{View}\langle \mathcal{A}', \mathcal{M}'(x) \rangle) \in T] \\ & \leq e^{\varepsilon} \Pr[\mathtt{Post}(\mathtt{View}\langle \mathcal{A}', \mathcal{M}'(x') \rangle) \in T] + \delta \\ & = e^{\varepsilon} \Pr[\mathtt{View}\langle \mathcal{A}, \mathcal{M}(x') \rangle \in T] + \delta \end{aligned}$$

Therefore, $\mathcal{M}$ is also (ε, δ)-DP.

Given Lemma 2.5, for all of the concurrent compositions we considered in this paper, we assume that the concurrent compositions are ordered. For example, if an adversary $\mathcal{A}$ is concurrently interacting with two mechanisms $\mathrm{ConComp}(\mathcal{M}_0, \mathcal{M}_1)$, we assumes that the queries are alternates between $\mathcal{M}_0$ and $\mathcal{M}_1$.

3 Concurrent Composition for Pure Interactive Differential Privacy

In this section, we show that for pure differential privacy, the privacy bound for concurrent composition is the same as for sequential or noninteractive composition. The proof idea is that in an interactive protocol where the adversary is concurrently interacting with multiple mechanisms, its interaction with one particular mechanism could be viewed as the combination of the adversary and the remaining mechanisms interacting with that mechanism, and the differential privacy guarantee still holds for the "combined adversary".

A useful notation for thinking about differential privacy and simplify presentations is defined below.

Definition 3.1. *Two random variables Y and Z taking values in the same output space $\mathcal{Y}$ is (ε, δ)-indistinguishable if for every event $T \subseteq \mathcal{Y}$, we have:*

$$\Pr[Y \in T] \le e^{\varepsilon} \Pr[Z \in T] + \delta$$

$$\Pr[Z \in T] \le e^{\varepsilon} \Pr[Y \in T] + \delta$$

which is denoted as $Y \overset{(\varepsilon,\delta)}{\approx} Z$.

Notice that an algorithm $\mathcal{M}$ is (ε, δ) differentially private if and only if for all pairs of adjacent datasets x, x', we have $\mathcal{M}(x) \overset{(\varepsilon,\delta)}{\approx} \mathcal{M}(x')$.

Lemma 3.2 ([23]). *For random variables X, Y, Z, if $X \overset{(\varepsilon_1,0)}{\approx} Y$, $Y \overset{(\varepsilon_2,0)}{\approx} Z$, then $X \overset{(\varepsilon_1+\varepsilon_2,0)}{\approx} Z$.*

Theorem 3.3 (Basic Composition of Pure Interactive Differential Privacy). *If interactive mechanisms $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each $(\varepsilon_i, 0)$-differentially private, then their concurrent composition $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $\left(\sum_{i=0}^{k-1} \varepsilon_i, 0\right)$-interactive differentially private.*

Proof. We first consider the simplest case that $\mathcal{A}$ concurrently interact with 2 mechanisms $\mathcal{M}, \tilde{\mathcal{M}}$, and then extend the result to general amount of mechanisms. Suppose $\mathcal{M}$ and $\tilde{\mathcal{M}}$ are each $(\varepsilon, 0)$ and $(\tilde{\varepsilon}, 0)$-differentially private interactive mechanisms. Denote the messages received by $\mathcal{A}$ from $\mathcal{M}$ as $(a_0, a_1, \ldots,)$, and the messages received by $\mathcal{A}$ from $\tilde{\mathcal{M}}$ as $(\tilde{a}_0, \tilde{a}_1, \ldots,)$. Due to Lemma 2.5, we can WLOG assume $\mathcal{A}$ alternates messages between $\mathcal{M}$ and $\tilde{\mathcal{M}}$, i.e., the sequence of messages $\mathcal{A}$ received is $(a_0, \tilde{a}_0, a_1, \tilde{a}_1, \ldots,)$. We use $r_{\mathcal{A}}$, $r_{\mathcal{M}}, r_{\tilde{\mathcal{M}}}$ to denote the random coin tosses for $\mathcal{A}$, $\mathcal{M}$, and $\tilde{\mathcal{M}}$, respectively. We can view $\mathcal{A}$ and $\tilde{\mathcal{M}}(x)$ as a single adversary $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)$ interacting with $\mathcal{M}(x)$ defined as follows:

1. Random coin tosses for $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)$ consist of $r = (r_{\mathcal{A}}, r_{\tilde{\mathcal{M}}})$.
2. $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)(a_0, a_1, \ldots, a_{i-1}; r)$ is computed as follows:
   (a) $\tilde{q}_{i-1} = \mathcal{A}(a_0, \tilde{a}_0, a_1, \tilde{a}_1, \ldots, a_{i-1}; r_{\mathcal{A}})$.
   (b) $\tilde{a}_{i-1} = \tilde{\mathcal{M}}(x, \tilde{q}_0, \tilde{q}_1, \ldots, \tilde{q}_{i-1}; r_{\tilde{\mathcal{M}}})$.
   (c) $q_i = \mathcal{A}(a_0, \tilde{a}_0, \ldots, a_{i-1}, \tilde{a}_{i-1}; r_{\mathcal{A}})$.
   (d) Output q_i.

We can see that $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)$ is a well-defined strategy throughout the entire interactive protocol with $\mathcal{M}$, where the randomness of $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)$ is fixed as $(r_{\mathcal{A}}, r_{\tilde{\mathcal{M}}})$. Given a transcript of $\mathcal{A}^*_{\tilde{\mathcal{M}}}(x)$'s view $(r_{\mathcal{A}}, r_{\tilde{\mathcal{M}}}, x, a_0, a_1, \ldots,)$, we can recover the corresponding transcript of $\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x), \tilde{\mathcal{M}}(x))\rangle$ through the following post-processing algorithm $\texttt{Post}$, which is defined as follows:

$\texttt{Post}\,(r_{\mathcal{A}}, r_{\tilde{\mathcal{M}}}, a_0, a_1, \ldots, a_{T-1})$:

1. For $i = 1 \ldots T-1$, compute
   (a) $\tilde{q}_{i-1} = \mathcal{A}(a_0, \tilde{a}_0, \ldots, a_{i-1}; r_{\mathcal{A}})$
   (b) $\tilde{a}_{i-1} = \tilde{\mathcal{M}}(x, \tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_{i-1}; r_{\tilde{\mathcal{M}}})$
2. Output $(r_{\mathcal{A}}, a_0, \tilde{a}_0, \ldots, a_{T-1}, \tilde{a}_{T-1})$.

Observe that for every $(x, r_{\mathcal{A}}, r_{\mathcal{M}}, r_{\tilde{\mathcal{M}}})$,

$$\begin{aligned}&\texttt{Post}\left(\texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x; r_{\mathcal{A}}, r_{\tilde{\mathcal{M}}}), \mathcal{M}(x; r_{\mathcal{M}})\rangle\right)\\ &= \texttt{View}\langle \mathcal{A}(r_{\mathcal{A}}), \mathrm{ConComp}(\mathcal{M}(x; r_{\mathcal{M}}), \tilde{\mathcal{M}}(x; r_{\tilde{\mathcal{M}}}))\rangle\end{aligned}$$

Therefore we have

$$\begin{aligned}&\Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x), \tilde{\mathcal{M}}(x))\rangle \in T\right]\\ &\equiv \Pr\left[\texttt{Post}\left(\texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x), \mathcal{M}(x)\rangle\right) \in T\right]\end{aligned}$$

for every $T \subseteq \mathrm{Range}(\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x), \tilde{\mathcal{M}}(x))\rangle)$.

Since $\mathcal{M}$ is ε-differentially private, we know that

$$\texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x), \mathcal{M}(x)\rangle \overset{(\varepsilon,0)}{\approx} \texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x), \mathcal{M}(x')\rangle$$

which leads to

$$\begin{aligned}&\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x), \tilde{\mathcal{M}}(x))\rangle\\ &\equiv \texttt{Post}\left(\texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x), \mathcal{M}(x)\rangle\right)\\ &\overset{(\varepsilon,0)}{\approx} \texttt{Post}\left(\texttt{View}\langle \mathcal{A}^*_{\tilde{\mathcal{M}}}(x), \mathcal{M}(x')\rangle\right)\\ &\equiv \texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x'), \tilde{\mathcal{M}}(x))\rangle\end{aligned}$$

Symmetrically, we can obtain

$$\begin{aligned}&\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x))\rangle\\&\overset{(\tilde{\varepsilon},0)}{\approx}\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x'))\rangle\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x),\tilde{\mathcal{M}}(x))\rangle\\&\overset{(\varepsilon+\tilde{\varepsilon},0)}{\approx}\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x'))\rangle\end{aligned}$$

The result can be easily extended to the case when more than 2 mechanisms are concurrently composed by induction. Therefore for every $\varepsilon_i \geq 0$, if interactive mechanism $\mathcal{M}_i$ is $(\varepsilon_i, 0)$-differentially private for $i = 0, \ldots, k-1$, then the concurrent composition $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ is $\left(\sum_{i=0}^{k-1} \varepsilon_i, 0\right)$-differentially private.

This result tells us that even under concurrent composition, the privacy parameters of the resulting composed mechanisms are the "sum up" of the individual algorithms for the case pure differential privacy.

4 Concurrent Composition for Approximate Interactive Differential Privacy

In this section, we explore the privacy guarantee for the concurrent composition of interactive differential privacy when $\delta > 0$. We show a privacy guarantee of concurrent composition in a similar logic flow as in Theorem 3.3, but in approximate differential privacy. As argued in the proof of Theorem 3.3, when the adversary is interacting with two mechanisms, we can view $\mathcal{A}$ and one of the mechanisms as a single adversary interacting with another mechanism, and the view of the combined adversary still enjoy the differential privacy guarantee. Therefore, if both interactive mechanisms $\mathcal{M}$ and $\tilde{\mathcal{M}}$ are (ε, δ)-differentially private, then for all $S \subseteq \mathrm{Range}(\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x),\tilde{\mathcal{M}}(x))\rangle)$, we know that

$$\begin{aligned}&\Pr\left[\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x),\tilde{\mathcal{M}}(x))\rangle\in S\right]\\&\leq e^{\varepsilon}\Pr\left[\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x))\rangle\in S\right]+\delta\end{aligned}$$

and

$$\begin{aligned}&\Pr\left[\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x))\rangle\in S\right]\\&\leq e^{\varepsilon}\Pr\left[\texttt{View}\langle\mathcal{A},\mathrm{ConComp}(\mathcal{M}(x'),\tilde{\mathcal{M}}(x'))\rangle\in S\right]+\delta\end{aligned}$$

and therefore we know that

$$\begin{aligned}
&\Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x), \tilde{\mathcal{M}}(x))\rangle \in S\right] \\
&\le e^{\varepsilon} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x'), \tilde{\mathcal{M}}(x))\rangle \in S\right] + \delta \\
&\le e^{\varepsilon}(e^{\varepsilon} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x'), \tilde{\mathcal{M}}(x'))\rangle \in S\right] + \delta) + \delta \\
&\le e^{2\varepsilon} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}(x'), \tilde{\mathcal{M}}(x'))\rangle \in S\right] + (1 + e^{\varepsilon})\delta
\end{aligned}$$

A more general concurrent composition bound is stated and derived as follows:

Theorem 4.1 (Theorem 1.8 restated). *Let* $\sigma : \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, n-1\}$ *be any permutation of* $0, \ldots, n-1$. *If interactive mechanisms* $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ *are each* $(\varepsilon_i, \delta_i)$*-differentially private, then their concurrent composition* $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ *is* $\left(\sum_{i=0}^{k-1} \varepsilon_i, \delta_g\right)$*-differentially private, where*

$$\delta_g = \min_{\sigma} \left(\delta_{\sigma(0)} + \sum_{i=1}^{k-1} e^{\sum_{j=0}^{i-1} \varepsilon_{\sigma(j)}} \delta_{\sigma(i)} \right)$$

For mathematical convenience, we use an upper bound for δ_g *in practice and* $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ *is* $\left(\sum_{i=0}^{k-1} \varepsilon_i, k e^{\sum_{i=0}^{k-1} \varepsilon_i} \max_i(\delta_i)\right)$*-differentially private.*

Proof. We use a hybrid argument. For each $0 \le i \le k-1$, since $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$ differentially private, we know that

$$\begin{aligned}
&\Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \ldots, \mathcal{M}_{i-1}(x'), \mathcal{M}_i(x), \ldots, \mathcal{M}_{k-1}(x))\rangle \in S\right] \\
&\le e^{\varepsilon_i} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \ldots, \mathcal{M}_{i-1}(x'), \mathcal{M}_i(x'), \ldots, \mathcal{M}_{k-1}(x))\rangle \in S\right] + \delta_i
\end{aligned}$$

by viewing $\mathcal{A}$ and $\mathcal{M}_0, \ldots, \mathcal{M}_{i-1}, \mathcal{M}_{i+1}, \mathcal{M}_{k-1}$ as a combined adversary and follow a similar argument as in the proof of Theorem 1.8.

Therefore,

$$\begin{aligned}
&\Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x), \mathcal{M}_1(x), \ldots, \mathcal{M}_{k-1}(x))\rangle \in S\right] \\
&\le e^{\varepsilon_0} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \mathcal{M}_1(x), \ldots, \mathcal{M}_{k-1}(x))\rangle \in S\right] + \delta_0 \\
&\le e^{\varepsilon_0}(e^{\varepsilon_1} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \mathcal{M}_1(x'), \ldots, \mathcal{M}_{k-1}(x))\rangle \in S\right] + \delta_1) + \delta_0 \\
&\le \ldots \\
&\le e^{\sum_{i=0}^{k-1} \varepsilon_i} \Pr\left[\texttt{View}\langle \mathcal{A}, \mathrm{ConComp}(\mathcal{M}_0(x'), \mathcal{M}_1(x'), \ldots, \mathcal{M}_{k-1}(x'))\rangle \in S\right] \\
&\quad + (\delta_0 + e^{\varepsilon_0}\delta_1 + e^{\varepsilon_0 + \varepsilon_1}\delta_2 + \ldots + e^{\sum_{i=0}^{k-2} \varepsilon_i}\delta_{k-1})
\end{aligned}$$

We can see that the δ term of $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ depends on different permutations of $(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$, and the tightest possible bound for the δ term is

$$\min_{\sigma} \left(\delta_{\sigma(0)} + \sum_{i=1}^{k-1} e^{\sum_{j=0}^{i-1} \varepsilon_{\sigma(j)}} \delta_{\sigma(i)} \right)$$

We also note that $\delta_0 + e^{\varepsilon_0}\delta_1 + e^{\varepsilon_0+\varepsilon_1}\delta_2 + \ldots + e^{\sum_{i=0}^{k-2}\varepsilon_i}\delta_{k-1} \leq ke^{\sum_{i=0}^{k-1}\varepsilon_i}\max_i(\delta_i)$, which is more easier to work with in practice.

Notice that if the privacy parameters are homogeneous, i.e. every interactive mechanism is (ε, δ) differentially private, then this bound reduce to the bound of group privacy for (ε, δ)-differential privacy.

5 Characterization of ConComp for Pure Interactive Differential Privacy

[18] shows that to analyze the composition of arbitrary noninteractive $(\varepsilon_i, \delta_i)$-DP algorithms, it suffices to analyze the composition of the following simple variant of randomized response.

Definition 5.1 ([18]). *Define a randomized noninteractive algorithm* $\mathrm{RR}_{(\varepsilon,\delta)} : \{0,1\} \to \{0, 1, \text{'Iam0'}, \text{'Iam1'}\}$ *as follows:*

$$\begin{array}{ll}
\Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(0) = \text{'Iam0'}\right] = \delta & \Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(1) = \text{'Iam0'}\right] = 0 \\
\Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(0) = 0\right] = (1-\delta)\cdot\frac{e^{\varepsilon}}{1+e^{\varepsilon}} & \Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(1) = 0\right] = (1-\delta)\cdot\frac{1}{1+e^{\varepsilon}} \\
\Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(0) = 1\right] = (1-\delta)\cdot\frac{1}{1+e^{\varepsilon}} & \Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(1) = 1\right] = (1-\delta)\cdot\frac{e^{\varepsilon}}{1+e^{\varepsilon}} \\
\Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(0) = \text{'Iam1'}\right] = 0 & \Pr\left[\mathrm{RR}_{(\varepsilon,\delta)}(1) = \text{'Iam1'}\right] = \delta
\end{array}$$

Note that $\mathrm{RR}_{(\varepsilon,\delta)}$ is a noninteractive (ε, δ)-differentially private mechanism. [18] and [21] showed that $\mathrm{RR}_{(\varepsilon,\delta)}$ can be used to simulate the output of every (noninteractive) (ε, δ)-DP algorithm on adjacent databases. RR refers to "randomized response", as this mechanism is a generalization of the classic randomized response to $\delta > 0$ and $\varepsilon \neq \ln 2$ [25].

Theorem 5.2 ([18]). *Suppose that $\mathcal{M}$ is (ε, δ)-differentially private. Then for every pair of adjacent datasets x_0, x_1 there exists a randomized algorithm T s.t. $T(\mathrm{RR}(b))$ is identically distributed to $\mathcal{M}(x_b)$ for both $b = 0$ and $b = 1$.*

This theorem is useful due to one of the central properties of differential privacy is that it is preserved under "post-processing" [8,10], which is formulated as follows:

Lemma 5.3 (Post-processing). *If a randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is (ε, δ)-differentially private, and $\mathcal{F} : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then $\mathcal{F} \circ \mathcal{M} : \mathcal{X} \to \mathcal{Z}$ is also (ε, δ)-differentially private.*

In noninteractive setting, Theorem 5.2 can be used to prove the optimal composition theorem [18,21] since to analyze the composition of arbitrary $(\varepsilon_i, \delta_i)$-DP algorithms, it suffices to analyze the composition of $\mathrm{RR}_{(\varepsilon_i,\delta_i)}$ algorithms.

If we are able to prove a similar result that arbitrary interactive differential private mechanisms can also be simulated by the post-processing of randomized response where the interactive post-processing algorithm does not depend on the adversary, then we will be able to extend all results of composition theorem for noninteractive mechanisms to interactive mechanisms. In this paper, we consider the case of pure differential privacy.

Theorem 5.4 (Theorem 1.10 restated). *Suppose that $\mathcal{M}$ is an interactive $(\varepsilon, 0)$-differentially private mechanism. Then for every pair of adjacent datasets x_0, x_1 there exists an interactive mechanism T s.t. for every adversary $\mathcal{A}$ and every $b \in \{0, 1\}$ we have*

$$\texttt{View}(\mathcal{A}, \mathcal{M}(x_b)) \equiv \texttt{View}(\mathcal{A}, T(\mathrm{RR}_{(\varepsilon,0)}(b)))$$

Proof. For arbitrary sequence of queries $\boldsymbol{q}^{(t)} = (q_0, \ldots, q_{t-1})$ from $\mathcal{A}$, we denote by $\vec{\mathcal{M}}(x, \boldsymbol{q}^{(t)}) = (\mathcal{M}(x, \boldsymbol{q}^{(1)}), \mathcal{M}(x, \boldsymbol{q}^{(2)}), \ldots, \mathcal{M}(x, \boldsymbol{q}^{(t)}))$ the random variable consisting the first t responses from mechanism $\mathcal{M}$. We construct the interactive mechanism T receiving queries $\boldsymbol{q}^{(t)}$ as follows:

1. If $t = 0$, we have

$$\Pr\left[T(0, q_0) = a_0\right] = \frac{e^{\varepsilon} \Pr[\mathcal{M}(x_0, q_0) = a_0] - \Pr[\mathcal{M}(x_1, q_0) = a_0]}{e^{\varepsilon} - 1} \tag{6}$$

$$\Pr\left[T(1, q_0) = a_0\right] = \frac{e^{\varepsilon} \Pr[\mathcal{M}(x_1, q_0) = a_0] - \Pr[\mathcal{M}(x_0, q_0) = a_0]}{e^{\varepsilon} - 1} \tag{7}$$

2. If $t > 0$, given earlier responses $(a_0, \ldots, a_{t-2})$, we define

$$\begin{aligned}&\Pr\left[T(0, \boldsymbol{q}^{(t)}) = a_{t-1} | a_0, \ldots, a_{t-2}\right] \\ &= \frac{e^{\varepsilon} \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] - \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]}{(e^{\varepsilon} - 1) \Pr\left[\vec{T}(0, \boldsymbol{q}^{(t-1)}) = (a_0, \ldots, a_{t-2})\right]}\end{aligned} \tag{8}$$

$$\begin{aligned}&\Pr\left[T(1, \boldsymbol{q}^{(t)}) = a_{t-1} | a_0, \ldots, a_{t-2}\right] \\ &= \frac{e^{\varepsilon} \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] - \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]}{(e^{\varepsilon} - 1) \Pr\left[\vec{T}(1, \boldsymbol{q}^{(t-1)}) = (a_0, \ldots, a_{t-2})\right]}\end{aligned} \tag{9}$$

Therefore, the distribution of $\vec{T}$ is

$$\begin{aligned}&\Pr\left[\vec{T}(0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] \\ &= \frac{e^{\varepsilon} \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] - \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]}{e^{\varepsilon} - 1}\end{aligned}$$

$$\begin{aligned}&\Pr\left[\vec{T}(1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] \\ &= \frac{e^{\varepsilon} \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] - \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]}{e^{\varepsilon} - 1}\end{aligned}$$

We can easily verify that all of the above are valid probability distributions. For example,

$$\begin{aligned}
&\sum_{a_{t-1}} \Pr\left[T(0, \boldsymbol{q}^{(t)}) = a_{t-1} | a_0, \ldots, a_{t-2}\right] \\
&= \frac{e^\varepsilon \sum_{a_{t-1}} \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] - \sum_{a_{t-1}} \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]}{(e^\varepsilon - 1) \Pr\left[\vec{T}(0, \boldsymbol{q}^{(t-1)}) = (a_0, \ldots, a_{t-2})\right]} \\
&= \frac{e^\varepsilon \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-2})\right] - \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-2})\right]}{(e^\varepsilon - 1) \Pr\left[\vec{T}(0, \boldsymbol{q}^{(t-1)}) = (a_0, \ldots, a_{t-2})\right]} \\
&= 1 \qquad (10)
\end{aligned}$$

and for every possible a_{t-1}, the probability density is never negative since

$$\Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right] \le e^\varepsilon \Pr\left[\vec{\mathcal{M}}(x_1, \boldsymbol{q}^{(t)}) = (a_0, \ldots, a_{t-1})\right]$$

as $\mathcal{M}$ is $(\varepsilon, 0)$-DP.

We now show

$$\texttt{View}(\mathcal{A}, \mathcal{M}(x_b)) \equiv \texttt{View}(\mathcal{A}, T(\mathrm{RR}_{(\varepsilon,0)}(b)))$$

for the case of $b = 0$.

Fix any possible view $(r, a_0, \ldots, a_{t-1})$, we can derive the queries $\boldsymbol{q}^{(t)} = (q_0, \ldots, q_{t-1})$ from $\mathcal{A}$, where $q_i = \mathcal{A}(a_0, \ldots, a_{i-1}; r)$. Denote R as the random variable of the randomness of $\mathcal{A}$.

$$\begin{aligned}
&\Pr\left[\texttt{View}(\mathcal{A}, T(\mathrm{RR}_{(\varepsilon,0)}(0))) = (r, a_0, \ldots, a_{t-1})\right] \\
&= \Pr\left[\mathrm{RR}_{(\varepsilon,0)}(0) = 0\right] \Pr\left[\texttt{View}(\mathcal{A}, T(0)) = (r, a_0, \ldots, a_{t-1})\right] \\
&\quad + \Pr\left[\mathrm{RR}_{(\varepsilon,0)}(1) = 0\right] \Pr\left[\texttt{View}(\mathcal{A}, T(1)) = (r, a_0, \ldots, a_{t-1})\right] \\
&= \frac{e^\varepsilon}{1 + e^\varepsilon} \Pr\left[\texttt{View}(\mathcal{A}, T(0)) = (r, a_0, \ldots, a_{t-1})\right] \\
&\quad + \frac{1}{1 + e^\varepsilon} \Pr\left[\texttt{View}(\mathcal{A}, T(1)) = (r, a_0, \ldots, a_{t-1})\right] \\
&= \frac{e^\varepsilon}{1 + e^\varepsilon} \Pr[R = r] \Pr\left[\vec{T}(0, \boldsymbol{q}_r^{(t)}) = (a_0, \ldots, a_{t-1}) | R = r\right] \\
&\quad + \frac{1}{1 + e^\varepsilon} \Pr[R = r] \Pr\left[\vec{T}(1, \boldsymbol{q}_r^{(t)}) = (a_0, \ldots, a_{t-1}) | R = r\right] \\
&= \Pr[R = r] \Pr\left[\vec{\mathcal{M}}(x_0, \boldsymbol{q}_r^{(t)}) = (a_0, \ldots, a_{t-1}) | R = r\right] \\
&= \Pr\left[\texttt{View}(\mathcal{A}, \mathcal{M}(x_0)) = (r, a_0, \ldots, a_{t-1})\right]
\end{aligned}$$

The case of $b = 1$ could be similarly proved. Therefore, we proved the existence of such an interactive mechanism T for any $(\varepsilon, 0)$ interactive DP mechanisms.

The above theorem suggests that the noninteractive $\mathrm{RR}_{(\varepsilon,0)}$ can simulate any $(\varepsilon,0)$ interactive DP algorithm. Since it is known that post-processing preserves differential privacy (Lemma 5.3), it follows that to analyze the concurrent composition of arbitrary $(\varepsilon_i,0)$ interactive differentially private algorithms, it suffices to analyze the composition of randomized response $\mathrm{RR}_{(\varepsilon_i,0)}$. For an interactive mechanism $\mathcal{M}$, we define $\mathrm{PrivLoss}(\mathcal{M},\delta) = \inf\{\varepsilon \geq 0 : \mathcal{M} \text{ is } (\varepsilon,\delta)\text{-DP}\}$, thus given a target security parameter δ_g, the privacy loss of the concurrent composition of mechanisms $\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}$ is denoted as $\mathrm{PrivLoss}(\mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}),\delta_g)$. When the mechanisms $\mathcal{M}_i$ are noninteractive (like $\mathrm{RR}_{(\varepsilon,\delta)}$) we write Comp rather than ConComp.

Lemma 5.5. *Suppose there are interactive mechanisms $\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}$ where for each $0 \leq i \leq k-1$, $\mathcal{M}_i$ is $(\varepsilon_i,0)$-differentially private. For any values of $\varepsilon_0,\ldots,\varepsilon_{k-1} \geq 0$, $\delta_g \in [0,1)$, we have*

$$\begin{aligned}&\mathrm{PrivLoss}(\mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}),\delta_g)\\ &= \mathrm{PrivLoss}\left(\mathrm{Comp}(\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}),\delta_g\right)\end{aligned}$$

Proof. We want to show that

$$\begin{aligned}&\inf\{\varepsilon_g \geq 0 : \mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}) \text{ is } (\varepsilon_g,\delta_g)-\mathrm{DP}\}\\ &= \inf\left\{\varepsilon_g \geq 0 : \mathrm{Comp}\left(\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}\right) \text{ is } (\varepsilon_g,\delta_g)-\mathrm{DP}\right\}\end{aligned}$$

Since the noninteractive $\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}$ can be viewed as a special case of interactive DP mechanisms, we have

$$\begin{aligned}&\inf\{\varepsilon_g \geq 0 : \mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}) \text{ is } (\varepsilon_g,\delta_g)-\mathrm{DP}\}\\ &\geq \inf\left\{\varepsilon_g \geq 0 : \mathrm{Comp}\left(\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}\right) \text{ is } (\varepsilon_g,\delta_g)-\mathrm{DP}\right\}\end{aligned}$$

For the other direction, suppose $\mathrm{Comp}\left(\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}\right)$ is $(\varepsilon_g^*,\delta_g)$-DP. By post-processing inequality, we know any for any tuple of post-processing interactive mechanisms $T_0,\ldots,T_{k-1}$, $\mathrm{ConComp}\left(T_0\left(\mathrm{RR}_{(\varepsilon_0,0)}\right),\ldots,T_{k-1}\left(\mathrm{RR}_{(\varepsilon_{k-1},0)}\right)\right)$ is also $(\varepsilon_g^*,\delta_g)$-DP. We know from Theorem 1.10 that for every pair of adjacent datasets x_0,x_1, there must exist interactive mechanisms $T_0,\ldots,T_{k-1}$ such that for every adversary $\mathcal{A}$, $\mathtt{View}\langle\mathcal{A},\mathcal{M}_i(x_b)\rangle$ is identically distributed as $\mathtt{View}\langle\mathcal{A},T_i(\mathrm{RR}_{(\varepsilon,0)}(b))\rangle$ for all $i = 0,\ldots,k-1$. Therefore, we know that $\mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1})$ is also $(\varepsilon_g^*,\delta_g)$-DP. Taking the infimum over ε_g^* will then complete the proof.

We note that $\mathrm{RR}_{(\varepsilon_0,0)},\ldots,\mathrm{RR}_{(\varepsilon_{k-1},0)}$ are noninteractive mechanisms, therefore we can use any composition theorems for noninteractive DP mechanisms to bound the privacy parameter of their composition. The tightest composition theorem for noninteractive DP is derived in [21].

Theorem 5.6 (Optimal Composition Theorem for noninteractive DP). *If $\mathcal{M}_0,\ldots,\mathcal{M}_{k-1}$ are each (ε_i,δ_i)-differentially private, then given the target security parameter δ_g, the privacy parameter of concurrent composition* $\mathrm{ConComp}(\mathcal{M}_0,\ldots,\mathcal{M}_{k-1})$

is upper bounded by the least value of $\varepsilon_g \geq 0$ such that

$$\frac{1}{\prod_{i=0}^{k-1}\left(1+\mathrm{e}^{\varepsilon_i}\right)} \sum_{S \subseteq\{0, \ldots, k-1\}} \max \left\{\mathrm{e}^{\sum_{i \in S} \varepsilon_i}-\mathrm{e}^{\varepsilon_g} \cdot \mathrm{e}^{\sum_{i \notin S} \varepsilon_i}, 0\right\} \leq 1-\frac{1-\delta_g}{\prod_{i=0}^{k-1}\left(1-\delta_i\right)}$$

Therefore, we are ready to bound the concurrent composition for an arbitrary set of interactive differentially private algorithms by simply plugging parameters to the optimal composition bound for noninteractive DP mechanisms in [21].

Theorem 5.7 (Corollary 1.11 Restated). *If $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are each $(\varepsilon_i, 0)$-differentially private, then given the target security parameter δ_g, the privacy parameter of concurrent composition* $\mathrm{ConComp}(\mathcal{M}_0, \ldots, \mathcal{M}_{k-1})$ *is upper bounded by the least value of $\varepsilon_g \geq 0$ such that*

$$\frac{1}{\prod_{i=0}^{k-1}\left(1+\mathrm{e}^{\varepsilon_i}\right)} \sum_{S \subseteq\{0, \ldots, k-1\}} \max \left\{\mathrm{e}^{\sum_{i \in S} \varepsilon_i}-\mathrm{e}^{\varepsilon_g} \cdot \mathrm{e}^{\sum_{i \notin S} \varepsilon_i}, 0\right\} \leq \delta_g$$

A special case when all $\mathcal{M}_0, \ldots, \mathcal{M}_{k-1}$ are $(\varepsilon, 0)$-differentially private, then privacy parameter is upper bounded by the least value of $\varepsilon_g \geq 0$ such that

$$\frac{1}{\left(1+\mathrm{e}^{\varepsilon}\right)^k} \sum_{i=0}^{k}\binom{k}{i} \max \left\{\mathrm{e}^{i \varepsilon}-\mathrm{e}^{\varepsilon_g} \cdot \mathrm{e}^{(k-i) \varepsilon}, 0\right\} \leq \delta_g$$

6 Experimental Results

In this section, we present empirical evidence for our conjecture that the Optimal Composition Theorems can be extended to the concurrent composition of approximate DP mechanisms. Specifically, we experimentally evaluate the conjecture for 3-message interactive mechanisms with 1-bit messages, as illustrated in Figure 1. The input for the mechanism is a bit $x \in \{0,1\}$ (corresponding to fixing two adjacent datasets). In the first round, the mechanism outputs a bit a_0 regardless of the query, so we omit q_0 and directly writing the probability of outputting a_0 as $\Pr[\mathcal{M}(x) = a_0]$. In the second round, the mechanism receives a query bit $\mathcal{A}(a_0)$ from the adversary, and output another bit a_1. Each such mechanism $\mathcal{M}_{\boldsymbol{p}}$ is defined by 10 parameters $\boldsymbol{p} = (p_0, p_{00}, p_{01}, p_{10}, p_{11}, p'_0, p'_{00}, p'_{01}, p'_{10}, p'_{11})$, where $p_0 = \Pr[\mathcal{M}_{\boldsymbol{p}}(0) = 0]$, $p'_0 = \Pr[\mathcal{M}_{\boldsymbol{p}}(1) = 0]$, $p_{ij} = \Pr[\mathcal{M}_{\boldsymbol{p}}(0, j) = (i, 0)]$, $p'_{ij} = \Pr[\mathcal{M}_{\boldsymbol{p}}(1, j) = (i, 0)]$. We note that the concurrent composition of two copies of such a mechanism already has a nontrivial interleaving, as shown in Figure 2.

We experimentally test whether instantiations of this 2-round interactive mechanism that are (ε, δ)-DP can be simulated as the interactive post-processing of randomized response $\mathrm{RR}_{(\varepsilon,\delta)}$. Specifically, we sample over 10,000 choices of the parameter vector $\boldsymbol{p}$ defining the mechanism $\mathcal{M}_{\boldsymbol{p}}$. For each one, we pre-define a value for δ and compute $\varepsilon = \mathrm{PrivLoss}(\mathcal{M}_{\boldsymbol{p}}, \delta)$ through enumerating over all possible adversaries.

Next, we used linear programming to see if there exists an interactive post-processing mechanism $\vec{T}$ which takes an output from $\mathrm{RR}_{(\varepsilon,\delta)}$, and sets it to have the exact same output distribution as the original 2-round for every possible query $q = (q_1)$ and output sequence (a_0, a_1):

$$\Pr\left[\vec{\mathcal{M}}(0,q) = (a_0,a_1)\right] = \delta \cdot \Pr\left[\vec{T}(\texttt{`Iam0'},q) = (a_0,a_1)\right]$$
$$+(1-\delta)\cdot\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\Pr\left[\vec{T}(0,q) = (a_0,a_1)\right] + (1-\delta)\cdot\frac{1}{e^{\varepsilon}+1}\Pr\left[\vec{T}(1,q) = (a_0,a_1)\right]$$

$$\Pr\left[\vec{\mathcal{M}}(1,q) = (a_0,a_1)\right] = (1-\delta)\cdot\frac{1}{e^{\varepsilon}+1}\Pr\left[\vec{T}(0,q) = (a_0,a_1)\right]$$
$$+(1-\delta)\cdot\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\Pr\left[\vec{T}(1,q) = (a_0,a_1)\right] + \delta\cdot\Pr\left[\vec{T}(\texttt{`Iam1'},q) = (a_0,a_1)\right]$$

Each $\Pr\left[\vec{T}(c,q) = (a_0,a_1)\right]$ is an unknown parameter here, where $c \in \{0, 1, \texttt{`Iam0'}, \texttt{`Iam1'}\}$. We also enforce them formulating valid distributions:

$$\forall c, q, a_0, a_1, \Pr\left[\vec{T}(c,q) = (a_0,a_1)\right] \geq 0$$

$$\forall c, \mathcal{A}, \sum_{a_0,a_1} \Pr\left[\vec{T}(c,\mathcal{A}(a_0)) = (a_0,a_1)\right] = 1$$

Besides, to construct a valid two-round mechanism, the probability of outputting a_0 in the first round should not depend on the future query q_1:

$$\forall c, a_0, \sum_{a_1} \Pr\left[\vec{T}(c,0) = (a_0,a_1)\right] = \sum_{a_1} \Pr\left[\vec{T}(c,1) = (a_0,a_1)\right]$$

We use the linear programming solver from SciPy [24] for solving the linear equation systems.

In all of our trials, we find a feasible $\vec{T}$, concluding that each of the mechanisms $\mathcal{M}_{\boldsymbol{p}}$ can be simulated by the post-processing of randomized response of the same (ε, δ) parameters.

Based on the above findings, we conjecture that the concurrent composition of interactive DP mechanisms may still have the same bound as the composition for noninteractive DP mechanisms. Besides, we might be able to prove it through a similar construction of interactive post-processing mechanisms as we did in Theorem 1.10. This means that every interactive DP mechanisms can be reduced to noninteractive randomized response. We leave the resolution of these conjectures for future work.

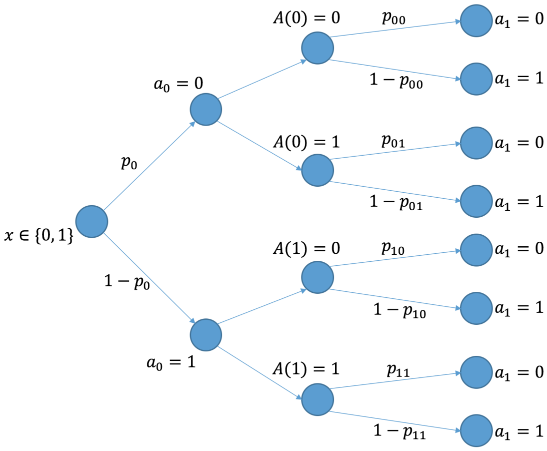

Fig. 1. 2-round mechanism we use in the experiment.

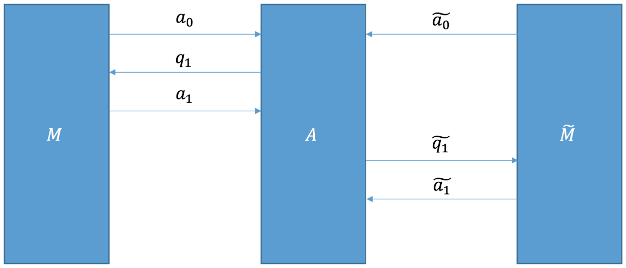

Fig. 2. Concurrent Composition of 2-round Mechanisms

7 Acknowledgement

We sincerely thank Boaz Barak, Cynthia Dwork, Marco Gaboardi, Michael Hay, Weiwei Pan, and Andy Vyrros for helpful comments and discussions.

References

1. Beimel, A., Nissim, K., Omri, E.: Distributed private data analysis: Simultaneously solving how and what. In: Annual International Cryptology Conference. pp. 451–468. Springer (2008)
2. Bun, M., Dwork, C., Rothblum, G.N., Steinke, T.: Composable and versatile privacy via truncated cdp. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing. pp. 74–86 (2018)
3. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of Cryptography Conference. pp. 635–658. Springer (2016)
4. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 639–648 (1996)
5. Chen, L., Ghazi, B., Kumar, R., Manurangsi, P.: On distributed differential privacy and counting distinct elements. arXiv preprint arXiv:2009.09604 (2020)
6. Dong, J., Roth, A., Su, W.J.: Gaussian differential privacy. arXiv preprint arXiv:1905.02383 (2019)
7. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 486–503. Springer (2006)
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. pp. 265–284. Springer (2006)
9. Dwork, C., Naor, M., Reingold, O., Rothblum, G.N., Vadhan, S.: On the complexity of differentially private data release: efficient algorithms and hardness results. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 381–390 (2009)
10. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3-4), 211–407 (2014)
11. Dwork, C., Rothblum, G.N.: Concentrated differential privacy. arXiv preprint arXiv:1603.01887 (2016)
12. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. pp. 51–60. IEEE (2010)
13. Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Conference on the Theory and Application of Cryptology. pp. 526–544. Springer (1989)
14. Gaboardi, M., Hay, M., Vadhan, S.: A programming framework for opendp (2020)
15. Goldreich, O.: Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali. Morgan & Claypool (2019)
16. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM Journal on Computing **25**(1), 169–192 (1996)
17. Hardt, M., Rothblum, G.N.: A multiplicative weights mechanism for privacy-preserving data analysis. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. pp. 61–70. IEEE (2010)
18. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. In: International conference on machine learning. pp. 1376–1385. PMLR (2015)

19. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? SIAM Journal on Computing **40**(3), 793–826 (2011)
20. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). pp. 263–275. IEEE (2017)
21. Murtagh, J., Vadhan, S.: The complexity of computing the optimal composition of differential privacy. In: Theory of Computing. pp. 157–175. Theory of Computing (2016)
22. Roth, A., Roughgarden, T.: Interactive privacy via the median mechanism. In: Proceedings of the forty-second ACM symposium on Theory of computing. pp. 765–774 (2010)
23. Vadhan, S.: The complexity of differential privacy. In: Tutorials on the Foundations of Cryptography, pp. 347–450. Springer (2017)
24. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods **17**(3), 261–272 (2020)
25. Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association **60**(309), 63–69 (1965)