Concurrent Composition for Differentially Private Continual Mechanisms

MONIKA HENZINGER, Institute of Science and Technology, Austria
ROODABEH SAFAVI, Institute of Science and Technology, Austria
SALIL VADHAN, Harvard University, United States

Many intended uses of differential privacy involve a *continual mechanism* that is set up to run continuously over a long period of time, making more statistical releases as either queries come in or the dataset is updated. In this paper, we give the first general treatment of privacy against *adaptive* adversaries for mechanisms that support dataset updates and a variety of queries, all arbitrarily interleaved. It also models a very general notion of neighboring, that includes both event-level and user-level privacy. We prove several *concurrent* composition theorems for continual mechanisms, which ensure privacy even when an adversary can interleave its queries and dataset updates to the different composed mechanisms. Previous concurrent composition theorems for differential privacy were only for the case when the dataset is static, with no adaptive updates. We also give the first interactive and continual generalizations of the "parallel composition theorem" for noninteractive differential privacy. Specifically, we show that the analogue of the noninteractive parallel composition theorem holds if either there are no adaptive dataset updates or each of the composed mechanisms satisfies pure differential privacy, but it fails to hold for composing approximately differentially private mechanisms with dataset updates. Thus, we prove a tight new composition theorem for this case. In addition, we prove concurrent filter compositions theorems for the scenarios in which the privacy parameters are adaptively chosen. We extend these results to other measures of differential privacy, including Rényi DP and f-DP.

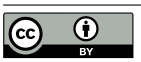
We then formalize a set of general conditions on a continual mechanism $\mathcal{M}$ that runs multiple continual sub-mechanisms such that the privacy guarantees of $\mathcal{M}$ follow directly using the above concurrent composition theorems on the sub-mechanisms, without further privacy loss. This enables us to give a simpler and modular privacy analysis of a recent continual histogram mechanism of Henzinger, Sricharan, and Steiner. In the case of approximate DP, ours is the first proof that shows that its privacy holds against adaptive adversaries. We also provide a framework that simplifies the analysis of local differential privacy when the protocol includes multi-round server-user interactions. Using this result, we simplify the privacy analysis of the core decomposition protocol of Dhulipala, Henzinger, Li, Liu, Sricharan, and Zhu [5].

CCS Concepts: • **Security and privacy**;

Additional Key Words and Phrases: differential privacy, concurrent composition, continual release, continual observation, data streaming, continual mechanisms, concurrent parallel composition, concurrent filter composition

ACM Reference Format:
Monika Henzinger, Roodabeh Safavi, and Salil Vadhan. 2026. Concurrent Composition for Differentially Private Continual Mechanisms. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 99 (May 2026), 26 pages. https://doi.org/10.1145/3801895

Authors' Contact Information: Monika Henzinger, Institute of Science and Technology, Klosterneuburg, Austria, monika.henzinger@ista.ac.at; Roodabeh Safavi, Institute of Science and Technology, Klosterneuburg, Austria, roodabeh.safavi@ista.ac.at; Salil Vadhan, Harvard University, Cambridge, United States, salil_vadhan@harvard.edu.



ACM 2836-6573/2026/5-ART99
https://doi.org/10.1145/3801895

1 Introduction

Differential privacy [8] is the now-standard framework for protecting privacy when performing statistical analysis or machine learning on sensitive datasets about individuals. In addition to having a rich and rapidly growing scientific literature, differential privacy has also seen large-scale adoption by technology companies and government agencies.

The familiar formulation of differential privacy is for a *noninteractive mechanism* $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$, which takes a dataset $x \in \mathcal{X}$ and produces a statistical release (e.g. a collection of statistics or the parameters of a machine learning model) $y \in \mathcal{Y}$. Differential privacy requires that for every two datasets x and x' that "differ on one individual's data," the output distributions $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are "almost indistinguishable" from each other. Formally, we have a *neighboring relation* $\sim$ on $\mathcal{X}$, which specifies which datasets "differ on one individual's data," or more generally what information requires privacy protection. The formal definition of differential privacy is as follows:

Definition 1.1 ((ϵ, δ)-indistinguishability [21]). *For $\epsilon \geq 0$ and $0 \leq \delta \leq 1$, two random variables X_1 and X_2 over the same domain $\mathcal{Y}$ are* (ϵ, δ)-indistinguishable *if for every measurable subset $S \subseteq \mathcal{Y}$,*

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(x') \in S] + \delta.$$

Definition 1.2 ((ϵ, δ)-DP for noninteractive mechanisms [8]). *For $\epsilon \geq 0$ and $0 \leq \delta \leq 1$, a randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is* (ϵ, δ)-differentially private *(or (ϵ, δ)-DP for short) with respect to a neighbor relation $\sim$ if for every two datasets $x_0, x_1 \in \mathcal{X}$ satisfying $x_0 \sim x_1$, the random variables $\mathcal{M}(x_0)$ and $\mathcal{M}(x_1)$ are (ϵ, δ)-indistinguishable.*

Typically we think of ϵ as a small constant, e.g. $\epsilon = .1$, while δ is cryptographically negligible. The case that $\delta = 0$ is referred to as *pure* differential privacy, and otherwise Definition 1.1 is known as *approximate* differential privacy. This simple definition is very useful and easy to work with, but does not capture many scenarios in which we want to apply differential privacy.

Differential Privacy Over Time. Many intended uses of differential privacy involve a system that is set up to run continuously over a long period of time, making more statistical releases as either queries come in or the dataset is updated, for example with people leaving and entering. (The latter case, with dataset updates, is known as differential privacy under *continual observation*, and was introduced in the concurrent works of Dwork, Naor, Pitassi, and Rothblum [9] and Chan, Shi, and Song [2] – see [15] for a survey.) These settings introduce a new attack surface, where an adversary may be *adaptive*, issuing queries or dataset updates based on the responses it receives.

When $\delta > 0$, there are mechanisms that are differentially private against oblivious adversaries but not against adaptive ones.[1] One natural example is the "advanced composition theorem" for differential privacy [11], which is known to fail if the adversary can select the privacy-loss parameters (ϵ_i, δ_i) of the composed mechanisms adaptively [25].

Until recently, adaptivity has been treated in an ad hoc manner in the differential privacy literature. For example, the aforementioned advanced composition theorem of [11] was formalized in a way that allows an adversary to adaptively choose the mechanisms and pairs of adjacent datasets, just not the privacy-loss parameters, and new composition theorems that allow for adaptive choices of privacy-loss parameters (known as privacy *filters* and privacy *odometers*) were given by [25, 28]. Other early examples where adaptive queries were analyzed include the Sparse Vector Technique (SVT) [10, 14, 22] and the Private Multiplicative Weights (PMW) algorithm [14].

[1] A simple artificial example is the following: the mechanism's first release can include a uniformly random number r from $\{1, 2, \ldots, \lceil 1/\delta \rceil\}$, and then if the adversary's next query is r, the mechanism publishes the entire dataset in the clear.

A few years ago, Vadhan and Wang [26] initiated a systematic study of differentially private mechanisms $\mathcal{M}$ that support adaptive queries. Specifically, they formalized the notion of an *interactive mechanism* $\mathcal{M}$, which begins with a dataset as input and maintains a potentially secret state as it answers queries that come in over time. The use of a secret state, as utilized in both SVT and PMW, allows for correlating the answers to queries over time, and often allows for the privacy-loss budget to degrade much more slowly than composing noninteractive mechanisms would allow. They defined $\mathcal{M}$ to be an (ϵ, δ)-*DP interactive mechanism* if for every pair of adjacent datasets $x \sim x'$ and every adaptive adversary $\mathcal{A}$, the *view* of $\mathcal{A}$ when interacting with $\mathcal{M}(x)$ is (ϵ, δ)-indistinguishable from its view when interacting with $\mathcal{M}(x')$, where the *view* of $\mathcal{A}$ consists of the random coin tosses of $\mathcal{A}$ and the transcript of the interaction.

Vadhan and Wang [26] asked whether our existing composition theorems for differential privacy, where the mechanisms being composed are noninteractive mechanisms, extend to the *concurrent composition* of interactive mechanisms. In concurrent composition of interactive mechanisms, an adaptive adversary can interleave its queries to the different interactive mechanisms being composed. This is a realistic attack scenario when multiple interactive differentially private mechanisms are deployed in practice, and is also a useful primitive for analyzing differentially private algorithms that use several interactive mechanisms as building blocks. Vadhan and Wang showed that the basic composition theorem for pure differential privacy [8] extends to concurrent composition. This result was later generalized by Lyu [23] to all composition theorems for approximate differential privacy. Subsequent works gave concurrent composition theorems for other flavors of differential privacy, such as Rényi DP [23] and f-DP [27].

Note that the extension of composition theorems for noninteractive mechanism to concurrent composition theorems for interactive mechanisms is non-trivial: Let us call an interactive mechanism M M-adaptive private if it is differentially private against an adversary that was only interactive with M. Consider an adversary that first interacts adaptively with M_1, then M_2 and then with M_1 again. Note that for the second sequence of interactions with M_1 the adversary takes into account its prior interaction with M_1 as well as with M_2. Thus, the adversary has more information about the data than if it had only interacted with M_1 before and the fact that M_1 is M_1-adaptive private does not mean that it is private against this interleaved composition of M_1 and M_2.

Haney, Shoemate, Tian, Vadhan, Vyrros, Wang, and Xu [13] study the concurrent extension of composition theorems for noninteractive mechanisms where the adversary instantiates arbitrary mechanisms with adaptive privacy parameters. This setting is called *filter composition*, where a *filter* is a function Filt that maps the privacy parameters (ϵ_i, δ_i) of the created mechanisms to $\top$ or $\bot$. Suppose an adversary adaptively creates noninteractive mechanisms such that Filt applied to the privacy parameters of the created mechanisms equals $\top$ at every point in time. The Filt-filter composition of noninteractive mechanisms is said to satisfy (ϵ, δ)-DP if the view of every such adversary is (ϵ, δ)-indistinguishable when all created mechanisms are run on dataset x or its neighboring dataset x'. Haney et al. extend this definition to the concurrent Filt-filter composition of interactive mechanisms, where the adversary may not only create new mechanisms but also interact with existing interactive ones. They prove that if the Filt-filter composition of noninteractive mechanisms is (ϵ, δ)-DP, then the concurrent version for interactive mechanisms is also (ϵ, δ)-DP.

All of the above works are for the case where the *queries* to the mechanisms are adaptive, but the datasets are static, given as input to the mechanisms at the start of the interaction.[2] However, in some

[2]One can extend the concurrent composition theorems to the case where, similarly to the advanced composition theorem [11], an adversary can adaptively select a pair of adjacent datasets (x_i, x_i') for the i'th mechanism $\mathcal{M}_i$, but once the interactive mechanism $\mathcal{M}_i$ is started, its dataset is fixed and no further updates are allowed. Thus, whenever (x_{i+1}, x_{i+1}') are given, a new interactive mechanism $\mathcal{M}_{i+1}$ has to be started that does not know the secret state of $\mathcal{M}_i$.

real-world applications, both queries and dataset updates may be issued adaptively, i.e., the dynamic updates to the dataset might be chosen adaptively. For instance, Ghazi et al. [12] study an advertising problem in which the server receives both private data and queries over time. They design algorithms that accept dataset updates while applying noninteractive mechanisms—such as the discrete Laplace mechanism—to answer queries and analyze the privacy guarantees for their algorithm using existing results on privacy filters for composing noninteractive mechanisms with adaptive privacy-loss parameters. This independent work[3] shows that if a mechanism receiving both queries and dataset updates produces a differentially private output at each time step i (with adaptive parameters ϵ_i, δ_i at step i) such that $\sum_i \epsilon_i \leq \epsilon$ and $\sum_i \delta_i \leq \delta$, then the overall mechanism is (ϵ, δ)-DP. This guarantee can be seen as a special case of our filter composition theorem, where the composition rule is simple summation and each composed mechanism is restricted to a noninteractive mechanism corresponding to a single step. While some continual mechanisms do indeed employ independent noninteractive mechanisms at each step, many important examples—such as the Sparse Vector Technique (SVT)—generate outputs that are correlated across time. For these mechanisms, privacy must be analyzed over the entire output sequence rather than on a step-by-step basis to get the desired guarantees. (If the ϵ-DP SVT mechanism were analyzed on a step-by-step basis, then the privacy guarantee would degrade to $\Theta(q \cdot \epsilon)$-DP, where q is the number of queries issued, defeating the entire point of SVT, which is to have a privacy guarantee and noise scale that is independent of the number of queries.) As a consequence, in the concurrent composition of continual mechanisms (or even interactive mechanisms), one cannot simply reduce the analysis to the composition of per-step noninteractive mechanisms. This is why concurrent composition theorems for interactive mechanisms took several years to establish [13, 23, 26, 27]. Thus, the results of Ghazi et al. [12] do not apply to the more general setting of (filter) concurrent composition of continual mechanisms.[4]

Jain, Raskhodnikova, Sivakumar, and Smith [18] recently gave a formalization of adaptive adversaries under continual observation (where there are dataset updates but the queries are fixed) for the special case of *event-level* privacy, where two streams of updates are considered adjacent if they differ on only one update. Denisov, McMahan, Rush, Smith and Thakurta [4] extended this definition to arbitrary neighbor relations on update sequences. The lack of a general toolkit for reasoning about adaptivity in dataset updates has led to complicated and ad hoc proofs of privacy of continual mechanisms (see e.g. [17]). Rectifying this situation is what motivated our work.

Comparison with algorithms and data structures. We note that the different flavors of mechanisms discussed above are differential privacy analogues of familiar variants of randomized algorithms and data structures. A noninteractive mechanism is simply a *static* (or *batch*) *algorithm*. An interactive mechanism corresponds to a *static data structure*. Our focus in this paper is on continual mechanisms, which allow both queries and updates, and these can be viewed as *dynamic data structures*. Adaptivity is a common concern for the correctness properties of randomized data structures; here we are concerned instead with its effect on privacy properties.

Our contributions (A) We give a formalism that captures privacy against adaptive adversaries for *continual mechanisms*, which support both queries and dataset updates (Definition 3.6). This generalizes interactive mechanisms (which have static datasets) and continual observation (where mechanisms answer fixed queries). We note that specific mechanisms, like variants of SVT, have

[3]The first version of this work was available on arXiv prior to [12].

[4]It is Ghazi et al.'s definition of Individual DP (IDP, their Definition 4.1) and their composition theorem that is framed in terms of IDP that restricts to step-by-step privacy analyses corresponding to composition of noninteractive mechanisms. Their definition of (ϵ, δ)-DP interactive mechanisms (their Definition 3.6) is somewhat more general than their definition of IDP, but it still does not allow for exploiting correlated randomness across queries to reduce privacy loss, as in SVT. (This is because their Definition 3.4 requires the next state of the mechanism to be a function of only the dataset and the query, and not of the randomness used to generate the answer.)

been proposed to handle both dataset updates and queries [19], but without offering a definition of privacy against adversaries that can adaptively select both the updates and queries.
(B) We prove concurrent composition theorems for continual mechanisms, where an adaptive adversary may interleave queries and updates across mechanisms based on the entire history of observed outputs. As discussed above for the concurrent composition of interactive mechanisms, the concurrent setting involves additional adaptivity that is not supported by composition theorems for noninteractive mechanisms.[5] We show tight privacy bounds for the following useful forms of concurrent composition of continual mechanisms.

(1) We show that when the privacy-loss parameters are fixed, the standard composition theorems for noninteractive approximate differential privacy extend to the *concurrent composition of continual mechanisms* (Theorem 4.7). This result also holds for Rényi differential privacy and for f-DP[6] (see the full version).

(2) We extend the classical "parallel composition theorem" for noninteractive mechanisms to the concurrent parallel composition of an *unbounded*[7] number of adaptively chosen *interactive mechanisms* with an adaptively chosen pair of neighboring datasets (Theorem 5.7). In the full version, we show similar extensions for Rényi differential privacy and for f-DP.
(3) We extend the above concurrent parallel composition of interactive mechanisms to continual mechanisms when the composed mechanisms satisfy pure differential privacy (Corollary 5.5). For approximate DP, we construct a counterexample demonstrating that *the concurrent parallel composition of an unbounded number of continual mechanisms* with adaptively chosen dataset updates is not private (Theorem 5.2). Since (ϵ, δ)-DP is a special case of f-DP, this lower bound also applies to f-DP. In the full version, we further construct an analogous counterexample for the case where the composed mechanisms satisfy Rényi DP instead of approximate DP.
(4) By imposing an upper bound on $\prod_i (1 - \delta_i)$, where δ_i is the approximate privacy parameter of the i-th created mechanism, we derive a new *tight* composition theorem for concurrent parallel composition of continual mechanisms (Theorem 5.4).
(5) As an immediate consequence, our new theorem extends the classical parallel composition theorem for noninteractive mechanisms to the concurrent composition of an *unbounded number of adaptively chosen continual mechanisms* satisfying *pure* differential privacy (Corollary 5.5).
(6) In the full version, we show that, for every filter Filt, the concurrent Filt-filter composition of *continual mechanisms* enjoys the same privacy guarantees as the Filt-filter composition of noninteractive mechanisms. This result is also proved for Rényi DP and f-DP.

A summary of the positive results for the aforementioned types of concurrent composition of (ϵ, δ)-DP interactive and continual mechanisms is provided in Table 1.

[5]Existing results on the concurrent composition of *interactive* mechanisms do not apply to the concurrent composition of *continual* mechanisms, since in our model the adversary adaptively selects both (a) the mechanisms receiving neighboring inputs (including their privacy parameters and neighboring definition) and (b) the neighboring update sequences they receive over time, which is not the case for interactive mechanisms.
[6]The f-DP results in this paper assume *finite communication*: for every created continual mechanism, there exists a constant C bounding the sizes of its query and answer spaces and the number of queries it can answer before halting. This assumption was also made for interactive mechanisms in [27].
[7]Throughout this paper, we use the term "unbounded number" to refer to a quantity of unknown (potentially arbitrary) size.

	Concurrent Composition with Fixed Parameters	Concurrent Parallel Composition	Concurrent Filter Composition
CMs	Theorem 4.7 for (ϵ, δ)-DP	Corollary 5.5 for ϵ-DP Theorem 5.4 for (ϵ, δ)-DP	Theorem 12.4 in the full version for (ϵ, δ)-DP
IMs	[26] for ϵ-DP [23, 27] for (ϵ, δ)-DP	Theorem 5.7 for (ϵ, δ)-DP	[13] for (ϵ, δ)-DP

Table 1. Concurrent composition results for (ϵ, δ)-DP continual mechanisms (CMs) and interactive mechanisms (IMs).

(C) In the full version, we formalize a set of general conditions on a continual mechanism $\mathcal{M}$ that runs multiple continual sub-mechanisms such that the privacy guarantees of $\mathcal{M}$ follow directly using the above concurrent composition theorems on the sub-mechanisms, *without further privacy loss*. These conditions are also reformulated as a set of properties for pseudocode describing continual mechanisms.
(D) In the full version, we use our concurrent composition theorems for continual mechanisms, with the aforementioned general conditions, to give a simpler and more modular privacy analysis of a recent continual histogram mechanism [17]. In the case of approximate DP, ours is the first proof that the privacy of the mechanism holds against adaptive adversaries.
(E) In the full version, we model *local protocols* in the *local differential privacy model* as continual mechanisms with modular structure, and demonstrate how our concurrent composition theorems for continual mechanisms—particularly the concurrent parallel composition theorem for interactive mechanisms—can simplify their privacy analysis. As an example, we give a simple privacy analysis for the core decomposition local protocol proposed by Dhulipala, Henzinger, Li, Liu, Sricharan, and Zhu [5].

We elaborate on all of these contributions below.

A general formalism for privacy against adaptive adversaries. Our basic object of study is a *continual mechanism* $\mathcal{M}$, which is an interactive mechanism that is not given any input at the start, but simply receives and sends messages in a potentially randomized and stateful manner. These messages can represent, for instance, queries or dataset updates. To model static datasets (as in standard interactive differential privacy), the entire dataset can be given to $\mathcal{M}$ as its first message.

Towards defining privacy, we consider adaptive adversaries $\mathcal{A}$ that send messages m_i for $i = 1, 2, 3, \ldots$ that are always parsed as pairs $m_i = (m_i[0], m_i[1])$. Privacy is defined with respect to a *verification function* f that takes a sequence of such pairs $(m_1, m_2, \ldots, m_t)$ and outputs either $\top$ or $\bot$. When $f(m_1, m_2, \ldots, m_t) = \top$, we consider the two sequences $m[0] = (m_1[0], m_2[0], \ldots, m_t[0])$ and $m[1] = (m_1[1], m_2[1], \ldots, m_t[1])$ to be *neighboring*.

Now, by the choice of the verification function f, we can easily capture previously studied notions of privacy and more. For example, for a standard interactive mechanism, $f(m_1, m_2, \ldots, m_t)$ will check that $m_1[0] \sim m_1[1]$ are a pair of neighboring datasets, and that $m_i[0] = m_i[1]$ for all $i > 1$ (these represent the adaptive queries). For continual observation under event-level privacy, we interpret all of the $m_i[b]$'s as dataset updates, and f will require that $m[0]$ and $m[1]$ differ in at most one coordinate $i \in [k]$. For continual observation under "user-level" privacy, f will require that the entries in which $m[0]$ and $m[1]$ differ correspond to only one user. For continual mechanisms, we can interleave adaptive queries among the dataset updates by having f require that all $m_i[0] = m_i[1]$ whenever either one is a query (rather than a dataset update).

To go from a verification function f to a definition of privacy against adaptive adversaries, we utilize the notion of an *interactive post-processing mechanism* (IPM) [13], which is a stateful procedure that can stand between an interactive mechanism and an adversary, or two interactive or

continual mechanisms, or even two IPMs, and interact with both of them adaptively, sending and receiving messages to the mechanisms on its left and right. Specifically, as illustrated in Figure 1, we place two IPMs between the adversary $\mathcal{A}$ and the continual mechanism $\mathcal{M}$. The first is a *verifier* $\mathcal{V}[f]$ that, every time it receives a new message pair m_k from $\mathcal{A}$, applies f to the entire history of message pairs $(m_1, \ldots, m_k)$ sent by $\mathcal{A}$, and forwards m_k onward only if f returns $\top$. Otherwise $\mathcal{V}[f]$ halts the interaction. The second IPM is the *identifier* $\mathcal{I}$, which starts out with a secret bit $b \in \{0, 1\}$, and every time it receives a message pair $(m_i[0], m_i[1])$ from $\mathcal{V}[f]$, it forwards $m_i[b]$ on to the mechanism $\mathcal{M}$. The identifier $\mathcal{I}$ with bit b is shown by $\mathcal{I}(b)$. When $\mathcal{M}$ returns an answer a_i, $\mathcal{I}$ and $\mathcal{V}[f]$ just forward it back to $\mathcal{A}$. We denote the interactive mechanism representing the combination of $\mathcal{M}$, I, and $\mathcal{V}[f]$ by $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$.

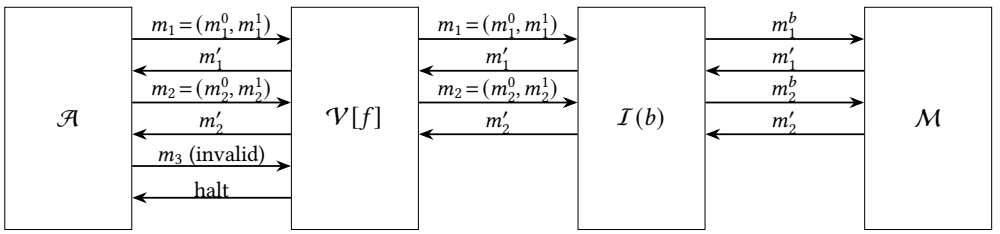


Fig. 1. Illustration of the interactions between $\mathcal{A}$, $\mathcal{V}[f]$, $\mathcal{I}(b)$, and $\mathcal{M}$

This leads to the following definition of differential privacy for continual mechanisms:

Definition 1.3 (DP for continual mechanisms). *Let f be a verification function. A continual mechanism $\mathcal{M}$ is* (ϵ, δ)-DP with respect to f *if for every adaptive adversary $\mathcal{A}$, the view of $\mathcal{A}$ when interacting with $\mathcal{V}[f] \circ^* \mathcal{I}(0) \circ^* \mathcal{M}$ is (ϵ, δ)-indistinguishable from its view when interacting with $\mathcal{V}[f] \circ^* \mathcal{I}(1) \circ^* \mathcal{M}$.*

The idea of defining security against adaptive adversaries by having the adversary send pairs of messages and trying to distinguish whether the first message in each pair or the second message in each pair is being used goes back to the notion of *left-or-right indistinguishability* in cryptography [1]. What is different here is its combination with a verification function f to enforce the neighboring condition, which is necessary to capture differential privacy.

As far as we know, this definition captures all of the previous definitions of privacy against adaptive adversaries as special cases (by appropriate choices of f), as well as modeling adaptive adversaries in even more scenarios (such as continual observation with user-level privacy, and mixtures of adaptively-chosen dataset updates and queries).

Furthermore, it allows us to reduce the analysis of general continual mechanisms to that of the previously studied notion of interactive mechanisms. Indeed, by inspection, we observe that a *continual* mechanism $\mathcal{M}$ is (ϵ, δ)-DP with respect to f if and only if the *interactive* mechanism $\mathcal{M}'(\cdot) = \mathcal{V}[f] \circ^* \mathcal{I}(\cdot) \circ^* \mathcal{M}$ is (ϵ, δ)-DP on the dataset space $\{0, 1\}$ (where $0 \sim 1$).

Concurrent Composition Theorems for Continual Mechanisms. With the above definitions in hand, we can formulate and prove new concurrent composition theorems for continual mechanisms. Recall that with concurrent composition of interactive mechanisms, an adaptive adversary (or honest analyst) can interleave its queries to the different interactive mechanisms being composed; for concurrent composition of *continual* mechanisms, we also allow the dataset *updates* to be interleaved with the queries and each other. See Figure 2 for illustration.

Our first concurrent composition theorem is as follows:

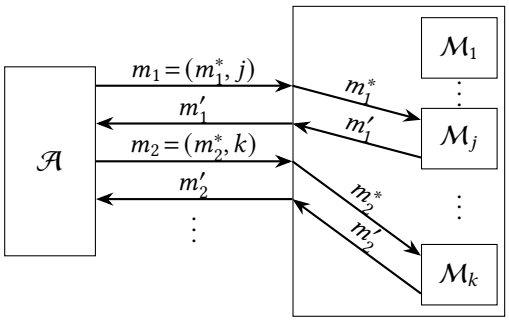


Fig. 2. Illustration of the interactions between an honest adversary $\mathcal{A}$ and the concurrent composition of k continual mechanisms

Theorem 1.4 (concurrent composition of continual mechanisms, informally stated). *The concurrent composition of continual mechanisms $\mathcal{M}_i$ that are (ϵ_i, δ_i)-DP w.r.t. a verification function f_i is (ϵ, δ)-DP for the same (ϵ, δ) that holds for composing noninteractive mechanisms that are (ϵ_i, δ_i)-DP.*

This result allows the adversary to adaptively and concurrently choose the neighboring sequences fed to each $\mathcal{M}_i$ based on the responses received from the other mechanisms. In the full version, analogous results are shown for Rényi differential privacy and, under the finite-communication assumption, for f-DP. A key step in the proof of Theorem 1.4 is to show that for any (ϵ_i, δ_i)-DP continual mechanism $\mathcal{M}$, the interactive mechanism $\mathcal{M}'(\cdot) = \mathcal{V}[f] \circ^* \mathcal{I}(\cdot) \circ^* \mathcal{M}$ can be simulated by post-processing the *randomized response* mechanism $\mathrm{RR}_{\epsilon_i,\delta_i}(\cdot)$. The mechanism $\mathrm{RR}_{\epsilon_i,\delta_i}$ takes a bit $b \in \{0,1\}$ as input and returns a pair according to the following distribution:

$$\mathrm{RR}_{\epsilon_i,\delta_i}(b) = \begin{cases} (\top, b) & \text{w.p. } \delta_i \\ (\bot, b) & \text{w.p. } (1-\delta_i)\frac{e^\epsilon}{1+e^{\epsilon_i}} \\ (\bot, 1-b) & \text{w.p. } (1-\delta_i)\frac{1}{1+e^{\epsilon_i}} \end{cases}$$

It is known that $\mathrm{RR}_{\epsilon_i,\delta_i}$ is (ϵ, δ)-DP with respect to a neighbor relation $\sim$, where $0 \sim 1$ and $1 \sim 0$ [20]. Lyu [23] proves this post-processing result under the assumptions that the answer space of $\mathcal{M}$ is finite and that there exists a predetermined upper bound T on the number of interactions. Vadhan and Zhang [27] additionally assume that the query set is finite. We remove the latter assumption and relax the former to having an infinite (but discrete) answer space.

Next, we explore the concurrent and continual analogue of *parallel composition* of differential privacy. To review parallel composition, suppose that $\mathcal{M}_1, \ldots, \mathcal{M}_\ell$ are each (ϵ_0, δ_0)-DP noninteractive mechanisms with respect to a neighbor relation $\sim^*$, where each $\mathcal{M}_i$ expects a dataset x_i as input, and for a parameter $k \in \{1, \ldots, \ell\}$, we define a neighboring relation $\sim^{(k)}$ on ℓ-tuples as follows: $(x_1, \ldots, x_\ell) \sim^{(k)} (x_1', \ldots, x_\ell')$ if and only if $x_i = x_i'$ for all but at most k indices $i \in [\ell]$, for which $x_i \sim^* x_i'$. We refer to the mechanism $\mathcal{M}'(\cdot) = (\mathcal{M}_1(\cdot), \ldots, \mathcal{M}_\ell(\cdot))$ with neighbor relation $\sim^{(k)}$ as the *k-sparse parallel composition* of $\mathcal{M}_1, \ldots, \mathcal{M}_\ell$. If $\mathcal{M}_1, \ldots, \mathcal{M}_\ell$ are noninteractive mechanisms, then it is known that $\mathcal{M}'$ satisfies the same level of (ϵ, δ)-DP as the (standard) composition of k (rather than ℓ) (ϵ_0, δ_0)-DP mechanisms.

We extend this notion to the *concurrent k-sparse parallel composition of continual mechanisms*, where an adversary (or an honest analyst) may adaptively create continual mechanisms (with no bound ℓ on the number of mechanism), where each mechanism satisfies (ϵ_0, δ_0)-DP with respect to a verification function f^*, and interact with them concurrently. Formally, instead of fixing one dataset

pair (x_i, x_i') for each mechanism in advance, the adversary issues pairs of messages $m = (m^0, m^1)$ to mechanisms over time. All but at most k mechanisms receive identical message pairs throughout the interaction, while for each of the k exceptional mechanisms, the sequence of message pairs σ satisfies $f^*(\sigma) = \top$. If messages are data records to be inserted in the dataset, then this concurrent composition captures adaptive partitioning of data among mechanisms over time. Note that an adversary can decide to designate a mechanism $\mathcal{M}_i$ as one of the k "exceptional" mechanisms, i.e., issue at least one pair of non-identical messages to $\mathcal{M}_i$, after potentially interacting with other mechanisms and sending multiple identical message pairs to $\mathcal{M}_i$ itself.

First, we extend the parallel composition of noninteractive mechanisms to the concurrent composition of interactive mechanisms (where queries are adaptive queries but datasets x_i are static, rather than dynamically updated).[8] We recall that an interactive mechanism is modeled as a continual mechanism receiving a dataset as its first message and queries as subsequent messages. (The verification function f^* ensures that the first message is a pair of neighbor datasets, while the subsequent pairs are identical queries.) The formal statement of the following theorem, in Theorem 5.7, more generally composes interactive mechanisms $\mathcal{M}_i$ with differing (ϵ_i, δ_i) parameters.

Theorem 1.5 (concurrent parallel composition of interactive mechanisms). *For every $\epsilon_0 > 0$ and $0 \le \delta_0 \le 1$ and $k \in \mathbb{N}$, and every (ϵ_0, δ_0)-DP interactive mechanism $\mathcal{M}_0$, the concurrent k-sparse parallel composition of an unbounded number of copies of $\mathcal{M}_0$ satisfies the same (ϵ, δ)-DP guarantees as the composition of k noninteractive (ϵ_0, δ_0)-DP mechanisms.*

We prove analogous results for Rényi DP and, under the finite-communication assumption, for f-DP in the full version. Then we turn to the case of *continual mechanisms*, where there can also be adaptive dataset updates. We prove a negative result for approximate DP ($\delta_0 > 0$):

Theorem 1.6 (counterexample for the concurrent parallel composition of approximate DP continual mechanisms, informally stated). *For every $\delta_0 > 0$, there is an $(0, \delta_0)$-DP continual mechanism $\mathcal{M}_0$ such that for every $\ell \in \mathbb{N}$, the concurrent 1-sparse parallel composition of ℓ copies of $\mathcal{M}_0$ is not $(0, \delta)$-DP for any $\delta < 1 - (1 - \delta_0)^\ell$.*

This theorem says that, at least in terms of the δ parameter, concurrent parallel composition of continual mechanisms does not provide any benefit over non-parallel composition, since the bound of $1 - (1 - \delta_0)^\ell$ is the bound that we would get by just applying composition over all ℓ mechanisms. The high-level idea of this counterexample is to define a continual mechanism $\mathcal{M}_0$ that, upon receiving its first input message (regardless of the value), outputs $\perp$ with probability δ and $\top$ otherwise. If the first output is $\perp$, then on its second input the mechanism simply returns that input, revealing it entirely. Consider an adversary $\mathcal{A}$ that creates ℓ instances of $\mathcal{M}_0$, sends the pair $(0, 0)$ to each instance, and then sends the pair $(0, 1)$ to the first mechanism whose response was $\perp$. With probability $1 - (1 - \delta)^\ell$, such a mechanism exists and its second response reveals its input bit, which equals the secret bit used by the identifier to filter the adversary's message pairs. An analogous counterexample for Rényi DP is provided in the full version.

Inspired by this counterexample, we prove a weaker version of Theorem 1.5 for the concurrent parallel composition of continual mechanisms, stated below. The formal version is given in Theorem 5.4 and applies to (ϵ_i, δ_i)-DP continual mechanisms with possibly different (ϵ_i, δ_i) parameters.

[8] As an intermediate step, Qiu et al.[24] analyzes the composition of infinitely many SVT instances (sharing a secret dataset) that are executed sequentially, where each instance begins only after the previous one halts, and where a differing element between two neighboring datasets can affect the output of at most one instance. Their result is weaker than ours as it does not allow for concurrent interaction of the adversary with the interactive mechanisms.

Theorem 1.7 (concurrent parallel composition of approx DP continual mechanisms, informally stated). *For every $\epsilon_0 > 0$, $0 \le \delta' \le 1$, and $k \in \mathbb{N}$, the concurrent k-sparse parallel composition of (ϵ_0, δ_i)-DP continual mechanisms, where the parameters δ_i are chosen adaptively such that $\prod_i (1-\delta_i) \ge 1-\delta'$, satisfies $(\epsilon, 1-(1-\delta)(1-\delta'))$-DP if the composition of k noninteractive $(\epsilon_0, 0)$-DP mechanisms is (ϵ, δ)-DP.*

The proof of Theorem 1.7 relies on a key property of our new post-processing construction:

Proposition 1.8 (post-processing of randomized response mechanisms, informal statement). *For every (ϵ, δ)-DP interactive mechanism $\mathcal{M}$ and every pair of neighboring datasets x_0, x_1, there exists an interactive post-processing mechanism $\mathcal{P}$ that receives the outputs of $\mathrm{RR}_{\epsilon,0}(b)$ and $\mathrm{RR}_{0,\delta}(b)$ and simulates $\mathcal{M}(x_b)$ identically. As long as the answer distributions of $\mathcal{M}(x_0)$ and $\mathcal{M}(x_1)$ to the queries asked so far are identical, $\mathcal{P}$ does not access the outcome of $\mathrm{RR}_{\epsilon,0}(b)$.*

The first sentence of this proposition was proven by Lyu [23] and Vadhan and Zhang [27] with finiteness condition on the communication. The novelty in our result is that $\mathcal{P}$ does not access the outcome of $\mathrm{RR}_{\epsilon,0}(b)$ until $\mathcal{M}(x_0)$ and $\mathcal{M}(x_1)$ differ. Our construction builds on Lyu's proof, where an IPM is initially given the outcome of $\mathrm{RR}_{\epsilon,\delta}(b)$. Let $\mathcal{M}'$ be a continual mechanism that is (ϵ, δ)-DP with respect to a verification function f. Consider the interaction between $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}'$ and an adversary. As long as the adversary sends pairs of identical messages, the answer distributions of $\mathcal{V}[f] \circ^* \mathcal{I}(0) \circ^* \mathcal{M}'$ and $\mathcal{V}[f] \circ^* \mathcal{I}(1) \circ^* \mathcal{M}'$ will be identical. Thus, by replacing $\mathcal{M}(x_b)$ in Proposition 1.8 with $\mathcal{V}[f] \circ^* \mathcal{I}(b) \circ^* \mathcal{M}'$, we conclude that there exists an IPM $\mathcal{P}$ that simulates $\mathcal{V}[f] \circ^* \mathcal{I}(b) \circ^* \mathcal{M}'$ using only the output of $\mathrm{RR}_{0,\delta}(b)$ until the adversary issues a pair of non-identical messages. At that point, $\mathcal{P}$ requires the output of $\mathrm{RR}_{\epsilon,0}(b)$ to continue its simulation. This result is the core idea to show that in the concurrent k-sparse parallel composition of *continual* mechanisms, the ϵ parameters of the (potentially unboundedly many) composed mechanisms that are always assigned pairs of identical messages do not contribute to the overall privacy loss.

In the concurrent parallel composition of *interactive* mechanisms, the adversary sends to each mechanism a pair of identical or neighboring datasets as first message and identical query pairs as subsequent messages. The key idea in the proof of Theorem 1.5 is that we can determine whether a mechanism is one of the k exceptional ones solely from its first pair, before the mechanism produces any output. Recall that all messages of the adversary are filtered by a secret bit $b \in \{0, 1\}$. If a mechanism is non-exceptional, whether the bit b is zero or one makes no difference, and the first element of every pair can always be forwarded. In this proof, we simulate a mechanism by post-processing of a randomized-response mechanism with input b only if it is exceptional, and show that we incur no privacy cost—neither in ϵ nor in δ—from the non-exceptional mechanisms.

Setting $\delta' = 0$ in Theorem 1.7 yields the following corollary, stated formally (and for differing ϵ_i) in Corollary 1.9.

Corollary 1.9 (concurrent parallel composition of pure DP continual mechanisms, informally stated). *For every $\epsilon_0 > 0$ and integer $k \in \mathbb{N}$, the concurrent k-sparse parallel composition of $(\epsilon_0, 0)$-DP continual mechanisms satisfies the same (ϵ, δ)-DP guarantees as the composition of k noninteractive $(\epsilon_0, 0)$-DP mechanisms.*[9]

Concurrent filter composition theorems for continual mechanisms. Let $\mathrm{Filt} : ([0,\infty) \times [0,1])^* \to \{\top, \bot\}$ be a filter. Consider adversaries that create continual mechanisms with adaptive privacy parameters such that $\mathrm{Filt}(\sigma) = \top$ at every step, where σ is the sequence of parameters (ϵ_i, δ_i) of the created mechanisms $\mathcal{M}_i$. Suppose each $\mathcal{M}_i$ is (ϵ_i, δ_i)-DP with respect to a verification function f_i. In addition to creating new mechanisms, these adversaries may adaptively

[9] In some composition theorems (e.g., advanced composition), composing pure DP mechanisms yields (ϵ, δ)-DP with $\delta > 0$.

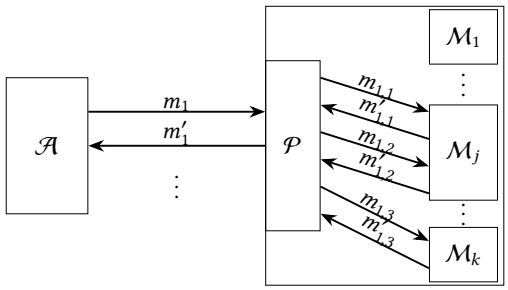


Fig. 3. Illustration of the interactions between an honest adversary $\mathcal{A}$ and a mechanism with k sub-mechanisms

generate pairs of inputs for existing mechanisms such that f_i applied to the sequence of input pairs for each $\mathcal{M}_i$ evaluates to $\top$. The concurrent Filt-filter composition of continual mechanisms is said to be (ϵ, δ)-DP if the view of such adversaries is (ϵ, δ)-indistinguishable when all mechanisms receive either the first inputs from their respective pairs or all receive the second ones.

Theorem 1.10 (Concurrent filter composition for continual mechanisms). *Let* Filt : $([0, \infty) \times [0, 1])^* \to \{\top, \perp\}$ *be a filter. The concurrent* Filt*-filter composition of continual mechanisms satisfies the same* (ϵ, δ)*-DP guarantees as the* Filt*-filter composition of noninteractive mechanisms.*

In the full version, this result is also shown for Rényi DP and, under the finite-communication assumption, for f-DP.

General Conditions for Privacy Analysis of Continual Mechanisms. Composition theorems for differential privacy have two main uses. One is to bound the accumulated privacy loss when DP systems are used repeatedly over time. Another is to facilitate the design of complex DP algorithms from simpler ones, for example in modular software designs. The concurrent composition theorems for interactive mechanisms have already found an application in the privacy analysis of matching algorithms in local edge differential privacy model [6]. (That work also has a result in the continual observation model, but the privacy is only proven for oblivious adversaries, which allows it to be proven without using a composition theorem for continual mechanisms.) Our concurrent composition theorem allows for a simpler and more modular privacy analysis of continual mechanisms built out of other continual mechanisms.

There are various applications where a continual mechanism executes multiple differentially private mechanisms, such as the Sparse Vector Technique (SVT) and continual counters, as sub-mechanisms and concurrently interacts with them. Let $\mathcal{M}$ denote such a composite mechanism. Upon receiving an input, $\mathcal{M}$ may create new sub-mechanisms, interact multiple times with the existing ones, and then return an output. $\mathcal{M}$ generates inputs for its sub-mechanisms based on its own inputs and the previous outputs of all sub-mechanisms. (See Figure 3 for an illustration.) Analyzing the privacy of such mechanism designs is challenging because the creation and inputs of sub-mechanisms are interdependent and adaptive. Even if the inputs of $\mathcal{M}$ are non-adaptive, the inputs to its sub-mechanisms may become adaptive: Let $\mathcal{M}_1$ and $\mathcal{M}_2$ denote the sub-mechanisms of $\mathcal{M}$. If the inputs of $\mathcal{M}_1$ depend on the outputs of $\mathcal{M}_2$ and the inputs of $\mathcal{M}_2$ depend on the outputs of $\mathcal{M}_1$, then $\mathcal{M}_1$ receives inputs correlated with its own past outputs. Later, we see an example of this situation, where a histogram mechanism creates d private continual counters at the initialization and an adaptive number of SVTs and Laplace mechanisms over time. In that design, the inputs of the counters depend on the output of the (active instance of the) SVT and vice versa.

In such modular designs, we model $\mathcal{M}$ as the *post-processing* of a specific concurrent composition of its (continual) sub-mechanisms. Since $\mathcal{M}$ accesses the raw (secret) data itself, it could in principle incur additional privacy loss beyond that of the concurrent composition of the sub-mechanisms. To rule out such privacy loss, we introduce a set of conditions on $\mathcal{M}$ and prove that, under these conditions, the privacy guarantee of $\mathcal{M}$ is exactly that of the concurrent composition of its sub-mechanisms. Intuitively, $\mathcal{M}$ must restrict its use of raw inputs solely to generating inputs for differentially private sub-mechanisms, while all other actions, such as creating new sub-mechanisms or determining output values, must depend only on the privatized outputs of the sub-mechanisms. This will ensure that $\mathcal{M}$ does not use any "privacy budget" in addition to the "privacy budget" used by the sub-mechanisms.

Continual mechanisms are typically presented using pseudocode (rather than as formal state machines exchanging messages, which are used by our conditions on $\mathcal{M}$), where the pseudocode receives an input, updates its variables, and returns an output. We therefore reformulate our conditions in this setting. We call a variable an *untainted variable* if it is either the output of a private sub-mechanism or computed solely based on other untainted variables. Our conditions are now as follows:

- *Destination condition:* Decisions about whether to produce an output, instantiate a new sub-mechanism (and of which type), or interact with an existing one (and which one) must be based only on untainted variables.
- *Response condition:* The values of $\mathcal{M}$'s outputs must be determined solely by untainted variables.
- *Mapping condition:* Fixing the values of untainted variables, neighboring inputs to $\mathcal{M}$ must be mapped to input sequences for the sub-mechanisms with a structure for which concurrent composition is private.

We show that if these conditions are satisfied, then the privacy guarantee of $\mathcal{M}$ is the same as that of the underlying concurrent composition. For example, if $\mathcal{M}$ executes k continual sub-mechanisms and ensures that its neighboring inputs yield neighboring input sequences for each sub-mechanism, then the privacy guarantee of $\mathcal{M}$ matches that of the concurrent composition of its sub-mechanisms as given in Theorem 1.4. The same conclusion holds for the concurrent k-sparse parallel composition of sub-mechanisms, as established in Theorem 1.5 and Corollary 1.9.

An Algorithmic Application. We give a new privacy proof for the *continual monotone histogram mechanism* of Henzinger, Sricharan, and Steiner (HSS) [17]. This mechanism involves concurrent executions of a simpler continual histogram mechanism and multiple instantiations of the Sparse Vector Technique mechanism. The original privacy analysis is rather intricate even for oblivious adversaries (because the mechanism itself is adaptive in its queries and dataset updates to the building-block mechanisms). Combining our concurrent composition theorems (both Theorem 1.4 and Corollary 1.9), we obtain a much simpler privacy proof. For approximate DP, the original proof was only for oblivious adversaries, whereas our proof also applies to adaptive adversaries.

An Application in Local Differential Privacy (LDP) Consider an untrusted server that concurrently interacts with multiple users, each holding a private dataset, with the goal of computing a function on the users' aggregated data. A *local protocol* specifies the interaction rules between the server and the users, where both the server and the users are modeled as interactive mechanisms. Differential privacy for a local protocol is defined in terms of the distribution of the server's *view*, consisting of its internal randomness and the transcript of all exchanged messages. Specifically, a protocol P is said to satisfy (ϵ, δ)-LDP if the server's view is (ϵ, δ)-indistinguishable for every pair of neighboring sequences of user datasets.

A useful notion of neighboring in local differential privacy (LDP) considers two assignments of datasets to users that are identical for all but at most k users, where the exceptional users datasets

are neighboring with respect to a given binary relation on user datasets. An example of this is a setting in which each user represents a hospital and each patient may have records in at most k hospitals. Analyzing the privacy of local protocols in which the server interacts concurrently with users over multiple rounds can be challenging. We observe that such interactions can be modeled as a *concurrent parallel composition* of interactive mechanisms, and our new privacy results for this concurrent composition directly yield a simplified privacy analysis for local protocols:

Theorem 1.11 (informal statement). *Let P be a local protocol with n users, and let $\sim^*$ be a neighbor relation on (individual) user datasets. Define two sequences of n datasets to be neighbors if they are identical for all but at most $k \in \mathbb{N}$ users, where each differing user's datasets are neighbors with respect to $\sim^*$. Suppose the interactive mechanism corresponding to every user is (ϵ_0, δ_0)-DP with respect to $\sim^*$ against adaptive adversaries. If the composition of k noninteractive (ϵ_0, δ_0)-DP mechanisms is (ϵ, δ)-DP, then the protocol P is (ϵ, δ)-LDP.*

In the formal statement of Theorem 1.11 in the full version, we incorporate the fact that the server in P restricts the queries sent to each user and enforces that they satisfy certain properties. Consequently, each user's privacy guarantee only needs to hold for adversaries whose queries obey these properties. In the full version, we also present a more general statement of Theorem 1.11 for an arbitrary choice of neighbor relation on sequences of user datasets. The concurrent composition theorem used in the general result must be selected based on the given neighbor relation.

We illustrate an application of Theorem 1.11 by simplifying the privacy analysis of the local protocol proposed by Dhulipala et al. [5] for the core decomposition problem. In that work, a sequence of datasets $x = (x_1, \ldots, x_n)$ represents a graph with n vertices, where x_i denotes the set of vertices adjacent to vertex i. (Users are vertices.) Differential privacy is defined with respect to *edge-neighboring*: two dataset sequences x and x' are neighbors if their corresponding graphs differ by a single edge. The edge-neighboring relation can be generalized to a relation $\sim$, where $x \sim x'$ if and only if the adjacency sets x_i and x'_i are identical for all vertices i but at most 2 of them, and for those vertices, x_i and x'_i are *add-remove neighbors*, meaning that they differ in the presence or absence of a single (adjacent) vertex. By defining $\sim^*$ in Theorem 1.11 to be the add-remove neighbor relation and setting $k = 2$, we obtain a simpler privacy analysis for the protocol in [5].

We note that Theorem 1.11 and its generalizations are stated for protocols in which each user holds a static dataset. In many real-world settings, however, users may join or leave, and their data may evolve over time. In such cases, users must be modeled more generally as continual mechanisms rather than interactive mechanisms. The privacy guarantees of these protocols can still be computed using our concurrent composition theorems for continual mechanisms.

We also note that in the above definition, all components (i.e., the server and the users) are assumed to be honest, meaning that they follow the protocol, and non-colluding, meaning that they do not share their internal states outside the protocol. In the presence of possible dishonesty or collusion, all users except the target users who hold the private data, together with the server, must be treated as a single adversary that interacts concurrently with the target users. In this case, privacy of the local protocol follows from the concurrent composition of the interactive or continual mechanisms, depending on whether the datasets evolve over time.

The full version of this paper is available on arXiv and contains all missing details [16]. This extended abstract is organized as follows. Section 2 introduces all necessary notation, including noninteractive mechanisms, interactive mechanisms, and interactive post-processing mechanism. Section 3 defines continual mechanisms, identifiers $\mathcal{I}$, verification functions f, verifiers $\mathcal{V}$ and differential privacy for continual mechanisms. Concurrent composition of continual mechanisms is defined in Section 4 and our first concurrent composition theorem is proven. Section 5 introduces

concurrent k-sparse parallel composition of continual mechanisms and interactive mechanisms, proves the theorems for this type of composition and shows the counterexample.

2 Preliminaries

Recall that a *non-interactive mechanism (NIM)* $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is a randomized function that maps a dataset $x \in \mathcal{X}$ to an element of $\mathcal{Y}$ and halts immediately. An *interactive mechanism (IM)* is defined formally as follows.

Definition 2.1 (Interactive Mechanism). *An* interactive mechanism (IM) *is a randomized state function $\mathcal{M} : S_{\mathcal{M}} \times Q_{\mathcal{M}} \to S_{\mathcal{M}} \times A_{\mathcal{M}}$, where $S_{\mathcal{M}}$ is the state space of $\mathcal{M}$, $Q_{\mathcal{M}}$ is the set of possible incoming messages or queries for $\mathcal{M}$, and $A_{\mathcal{M}}$ is the set of possible outgoing messages or answers generated by $\mathcal{M}$. Upon receiving a query $m \in Q_{\mathcal{M}}$, according to some distribution, $\mathcal{M}$ updates its current state $s \in S_{\mathcal{M}}$ to a new state $s' \in S_{\mathcal{M}}$ and generates an answer $m' \in A_{\mathcal{M}}$, which is described by $(s', m') \leftarrow \mathcal{M}(s, m)$. The set $S^{init}_{\mathcal{M}} \subseteq S_{\mathcal{M}}$ denotes the set of possible initial states for $\mathcal{M}$. For $s^{init} \in S^{init}_{\mathcal{M}}$, $\mathcal{M}(s^{init})$ represents the interactive mechanism $\mathcal{M}$ with the initial state s^{init}. If $S^{init}_{\mathcal{M}}$ is a singleton $\{s\}$, then we write $\mathcal{M}$ instead of $\mathcal{M}(s)$ for short.*

Remark 2.2. *In the differential privacy literature, a query typically refers to a question about the dataset held by a mechanism. However, in this paper, we use query as a shorthand for query message, meaning any input message to a mechanism that requires a response.*

The state of an IM $\mathcal{M}$ serves as its memory. The initial state of $\mathcal{M}$ can include secret information, such as a dataset $x \in \mathcal{X}$, or it can simply be an empty memory. Thus, in some cases, the size of the initial state space $S^{\text{init}}_{\mathcal{M}}$ is as large as the family of datasets $\mathcal{X}$, while in some other cases, $S^{\text{init}}_{\mathcal{M}}$ is a singleton consisting of a single state representing the empty memory.

Interactive randomized response. Recall that the standard randomized response mechanism $\text{RR}_{\epsilon,\delta}$ is an NIM that outputs a pair from the space $\{\top, \bot\} \times \{0, 1\}$. We define the *interactive randomized response*, $\text{IRR}_{\epsilon,\delta}$, to output the pair in two separate steps: it first produces an element from $\{\top, \bot\}$, and later returns a bit in $\{0, 1\}$, such that the joint distribution of the two outputs matches that of $\text{RR}_{\epsilon,\delta}$. Specifically, $\text{IRR}_{\epsilon,\delta}$ is an IM with a single query message q^*. Given an initial state $b \in \{0, 1\}$, $\text{IRR}_{\epsilon,\delta}(b)$ responds to the first query q^* with $\bot$ with probability δ and with $\top$ with probability $1 - \delta$. Upon receiving the query q^* for the second time, the mechanism acts as follows: If its first answer was $\bot$, then it returns the answer b with probability 1. Otherwise, if its first response was $\top$, then it outputs b with probability $\frac{e^\epsilon}{1+e^\epsilon}$ and $m = 1 - b$ with probability $\frac{1}{1+e^\epsilon}$. $\text{IRR}_{\epsilon,\delta}(b)$ halts the communication upon receiving q^* for the third time.

Interaction of IMs. Two interactive mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ satisfying $Q_{\mathcal{M}_1} = A_{\mathcal{M}_2}$ and $Q_{\mathcal{M}_2} = A_{\mathcal{M}_1}$ can interact with each other through message passing. For initial states $s_1 \in S^{\text{init}}_{\mathcal{M}_1}$ and $s_2 \in S^{\text{init}}_{\mathcal{M}_2}$, the interaction between $\mathcal{M}_1(s_1)$ and $\mathcal{M}_2(s_2)$ is defined as follows: It starts with round 1. Mechanism $\mathcal{M}_1(s_1)$ sends a message to $\mathcal{M}_2(s_2)$ in the odd rounds, and $\mathcal{M}_2(s_2)$ responds in the subsequent even rounds. The interaction continues until one of the mechanisms halts the communication. A mechanism halts the communication by sending the specific message halt at its round. Throughout the paper, whenever discussing the interaction between two IMs, we implicitly assume that the query and answer sets of the involved mechanisms match without mentioning it. [10]

Consider the interaction between $\mathcal{M}_1(s_1)$ and $\mathcal{M}_2(s_2)$, initiated by a message from $\mathcal{M}_1(s_1)$. The *view* of $\mathcal{M}_1(s_1)$ in this interaction, denoted $\text{View}(\mathcal{M}_1(s_1), \mathcal{M}_2(s_2))$, is defined as

$$\text{View}(\mathcal{M}_1(s_1), \mathcal{M}_2(s_2)) = (r_{\mathcal{M}_1}, m_1, m_2, m_3, \dots),$$

[10] In Definition 2.1, we defined an IM as a stateful randomized algorithm mapping its current state and an input message to a new state and an output message. To initiate the interaction and send the first message m_1, we can assume $\mathcal{M}_1(s_1)$ invokes $\mathcal{M}_1(s_1, \lambda)$, where λ is the empty message.

where $r_{\mathcal{M}_1}$ represents the internal randomness of $\mathcal{M}_1$, and $(m_1, m_2, \dots)$ is the transcript of messages exchanged between $\mathcal{M}_1$ and $\mathcal{M}_2$. We define $\Pi_{\mathcal{M}_1,\mathcal{M}_2}$ as the family of all possible views over all initial states $s_1 \in S^{\text{init}}_{\mathcal{M}_1}$ and $s_2 \in S^{\text{init}}_{\mathcal{M}_2}$.

To define differential privacy for an IM $\mathcal{M}$, we compare the views of a so-called *adversary* interacting with $\mathcal{M}(s)$ and $\mathcal{M}(s')$ for neighboring initial states s, s'. An (adaptive) adversary $\mathcal{A}$ is simply an IM with a singleton initial state space. Throughout this paper, adversaries always send the first message. Moreover, since adversaries are stateful algorithms, they can choose their messages adaptively based on the history of messages exchanged, meaning that all adversaries in this paper are *adaptive*.

Existing works [13, 23, 26, 27] view the initial state of $\mathcal{M}$ as a dataset, i.e., $S^{\text{init}}_{\mathcal{M}} = \mathcal{X}$, and define neighbor relations as binary relations on $\mathcal{X}$. However, for us it will be convenient to conceptually separate $S^{\text{init}}_{\mathcal{M}}$ from the data (which is changing continuously over time), but still define DP w.r.t. neighbor relations on $S^{\text{init}}_{\mathcal{M}}$. This leads to the following definition of differential privacy for IMs.

Definition 2.3 ((ϵ, δ)-DP for IMs). *Let $\mathcal{M}$ be an interactive mechanism, and let $\sim$ be a neighbor relation on $S^{init}_{\mathcal{M}}$. For $\epsilon \geq 0$ and $0 \leq \delta \leq 1$, $\mathcal{M}$ is (ϵ, δ)-differentially private (or (ϵ, δ)-DP for short) with respect to (w.r.t.) $\sim$ against adaptive adversaries if for every pair of neighboring initial states $s_0, s_1 \in S^{init}_{\mathcal{M}}$ satisfying $s_0 \sim s_1$ and every adaptive adversary $\mathcal{A}$, the random variables* $\text{View}(\mathcal{A}, \mathcal{M}(s_0))$ *and* $\text{View}(\mathcal{A}, \mathcal{M}(s_1))$ *are (ϵ, δ)-indistinguishable.*

Definition 2.4 (Equivalent IMs). *Let $\mathcal{M}$ and $\mathcal{M}'$ be two IMs with $Q_{\mathcal{M}} = Q_{\mathcal{M}'}$ and $A_{\mathcal{M}} = A_{\mathcal{M}'}$. Let s and s' be initial states for $\mathcal{M}$ and $\mathcal{M}'$, respectively. The mechanisms $\mathcal{M}(s)$ and $\mathcal{M}'(s')$ are* equivalent *if for every adversary $\mathcal{A}$ with $A_{\mathcal{A}} = Q_{\mathcal{M}}$ and $Q_{\mathcal{A}} = A_{\mathcal{M}}$, the random variables* $\text{View}(\mathcal{A}, \mathcal{M}(s))$ *and* $\text{View}(\mathcal{A}, \mathcal{M}'(s'))$ *are identically distributed.*

Interactive Post-Processing (IPM) For NIMs, a randomized post-processing function $\mathcal{T} : \mathcal{Y} \to \mathcal{Z}$ maps the output of a mechanism $\mathcal{N} : \mathcal{X} \to \mathcal{Y}$ to an element of $\mathcal{Z}$. Differential privacy for NIMs is known to be preserved under post-processing. Haney et al. [13] generalize post-processing functions to *interactive post-processing mechanisms (IPMs)* as follows and show that interactive post-processing preserves privacy guarantees for IMs.

Definition 2.5 (Interactive Post-Processing Mechanism [13]). *An interactive post-processing mechanism (IPM) is a stateful randomized algorithm $\mathcal{P} : S_{\mathcal{P}} \times \left((\{\mathsf{L}\} \times Q^L_{\mathcal{P}}) \cup (\{\mathsf{R}\} \times Q^R_{\mathcal{P}})\right) \to S_{\mathcal{P}} \times \left((\{\mathsf{L}\} \times A^L_{\mathcal{P}}) \cup (\{\mathsf{R}\} \times A^R_{\mathcal{P}})\right)$ that stands between two interactive mechanisms, referred to as the left and right mechanisms, communicating with them via message passing. The state space of $\mathcal{P}$ is denoted by $S_{\mathcal{P}}$, while $Q^L_{\mathcal{P}}$ and $Q^R_{\mathcal{P}}$ represent the sets of possible messages received by $\mathcal{P}$ from the left and right mechanisms, respectively. Similarly, $A^L_{\mathcal{P}}$ and $A^R_{\mathcal{P}}$ denote the sets of possible messages $\mathcal{P}$ sends to the left and right mechanisms. $\mathcal{P}$ randomly maps its current state $s \in S_{\mathcal{P}}$, an indicator $v \in \{\mathsf{L}, \mathsf{R}\}$ specifying whether the incoming message is from the left or right mechanism, and a message $m \in Q^L_{\mathcal{P}} \cup Q^R_{\mathcal{P}}$ to a new state $s' \in S_{\mathcal{P}}$, an indicator $v' \in \{\mathsf{L}, \mathsf{R}\}$ specifying whether the outgoing message is for the left or right mechanism, and a message $m' \in A^L_{\mathcal{P}} \cup A^R_{\mathcal{P}}$. This is described by $(s', v', m') \leftarrow \mathcal{P}(s, v, m)$.*

The set $S^{init}_{\mathcal{P}} \subseteq S_{\mathcal{P}}$ denotes the set of possible initial states for $\mathcal{P}$. For $s^{init} \in S^{init}_{\mathcal{P}}$, $\mathcal{P}(s^{init})$ represents the IPM $\mathcal{P}$ with the initial state s^{init}. If $S^{init}_{\mathcal{P}}$ is a singleton $\{s\}$, then we write $\mathcal{P}$ instead of $\mathcal{P}(s)$.

Post-processing of an IM $\mathcal{M}$ by an IPM $\mathcal{P}$. The post-processing of an NIM $\mathcal{N}$ by a non-interactive post-processing function $\mathcal{T}$, denoted as $\mathcal{T} \circ \mathcal{N}$, is an NIM that first runs $\mathcal{N}$ on the input dataset and then applies $\mathcal{T}$ to $\mathcal{N}$'s output, returning $\mathcal{T}$'s output. In contrast, *the post-processing of an IM* $\mathcal{M}$ by an IPM $\mathcal{P}$, denoted by $\mathcal{P} \circ^* \mathcal{M}$, is an IM that runs $\mathcal{P}$ and $\mathcal{M}$ internally. In an interaction with

another IM $\mathcal{A}$, $\mathcal{P} \circ^* \mathcal{M}$ forwards its input messages from $\mathcal{A}$ to $\mathcal{P}$ as left messages. Then, $\mathcal{P}$ and $\mathcal{M}$ communicate multiple times, with $\mathcal{M}$ acting as the right mechanism, until $\mathcal{P}$ returns a response for its left mechanism, at which point $\mathcal{P} \circ^* \mathcal{M}$ returns this response to $\mathcal{A}$. Note that $\mathcal{P}$ may return an answer for its left mechanism immediately after receiving a left message, without interacting with $\mathcal{M}$.

When defining $\mathcal{P} \circ^* \mathcal{M}$, we implicitly assume that the query and answer spaces of $\mathcal{P}$ and $\mathcal{M}$ match, i.e., $Q^R_{\mathcal{P}} = A_{\mathcal{M}}$ and $A^R_{\mathcal{P}} = Q_{\mathcal{M}}$. By definition, the IM $\mathcal{P} \circ^* \mathcal{M}$ has the state space $S_{\mathcal{P} \circ^* \mathcal{M}} = S_{\mathcal{M}} \times S_{\mathcal{P}}$, initial state space $S^{\text{init}}_{\mathcal{P} \circ^* \mathcal{M}} = S^{\text{init}}_{\mathcal{M}} \times S^{\text{init}}_{\mathcal{P}}$, query space $Q^L_{\mathcal{P}}$, and answer space $A^L_{\mathcal{P}}$. For readability, when $\mathcal{P} \circ^* \mathcal{M}$ is initialized with the state $(s^{\mathcal{M}}, s^{\mathcal{P}}) \in S^{\text{init}}_{\mathcal{P} \circ^* \mathcal{M}}$, we denote it as $\mathcal{P}(s^{\mathcal{P}}) \circ^* \mathcal{M}(s^{\mathcal{M}})$ instead of $(\mathcal{P} \circ^* \mathcal{M})\left((s^{\mathcal{M}}, s^{\mathcal{P}})\right)$.

Lemma 2.6 (Post-Processing Lemma for IMs [13]). *Let $\mathcal{M}$ be an IM with an initial state space $S^{init}_{\mathcal{M}}$ and $\mathcal{P}$ be an IPM with a singleton initial state space. For every pair of initial states $s_0, s_1 \in S^{init}_{\mathcal{M}}$, the following holds: If for every adversary $\mathcal{A}$, the random variables $\text{View}(\mathcal{A}, \mathcal{M}(s_0))$ and $\text{View}(\mathcal{A}, \mathcal{M}(s_1))$ are (ϵ, δ)-indistinguishable, then for every adversary $\mathcal{A}$, the random variables $\text{View}(\mathcal{A}, \mathcal{P} \circ^* \mathcal{M}(s_0))$ and $\text{View}(\mathcal{A}, \mathcal{P} \circ^* \mathcal{M}(s_1))$ are (ϵ, δ)-indistinguishable.* [11]

IPMs are not limited to interacting with IMs. Two IPMs $\mathcal{P}_1$ and $\mathcal{P}_2$ can interact with each other as well. The post-processing of $\mathcal{P}_2$ by $\mathcal{P}_1$, denoted by $\mathcal{P}_1 \circ^* \mathcal{P}_2$, is an IPM that forwards its left queries to $\mathcal{P}_1$ and its right queries to $\mathcal{P}_2$. The IPM $\mathcal{P}_1$ acts as the left mechanism of $\mathcal{P}_2$, and the IPM $\mathcal{P}_2$ acts as the right mechanism of $\mathcal{P}_1$. Upon receiving a query, these two IPMs (potentially) interact with each other multiple times until either $\mathcal{P}_1$ outputs a left answer or $\mathcal{P}_2$ outputs a right answer. In the former case, $\mathcal{P}_1 \circ^* \mathcal{P}_2$ outputs that answer to the left, and in the latter case, $\mathcal{P}_1 \circ^* \mathcal{P}_2$ outputs teh answer to the right.

Note that when defining $\mathcal{P}_1 \circ^* \mathcal{P}_2$, we implicitly assume that the query and answer spaces of $\mathcal{P}_1$ and $\mathcal{P}_2$ match, i.e., $Q^R_{\mathcal{P}_1} = A^L_{\mathcal{P}_2}$ and $A^R_{\mathcal{P}_1} = Q^L_{\mathcal{P}_2}$. By definition, the IPM $\mathcal{P}_1 \circ^* \mathcal{P}_2$ has state space $S_{\mathcal{P}_1 \circ^* \mathcal{P}_2} = S_{\mathcal{P}_1} \times S_{\mathcal{P}_2}$, left query space $Q^L_{\mathcal{P}_1 \circ^* \mathcal{P}_2} = Q^L_{\mathcal{P}_1}$, left answer space $A^L_{\mathcal{P}_1 \circ^* \mathcal{P}_2} = A^L_{\mathcal{P}_1}$, right query space $Q^R_{\mathcal{P}_1 \circ^* \mathcal{P}_2} = Q^R_{\mathcal{P}_2}$, and right answer space $A^R_{\mathcal{P}_1 \circ^* \mathcal{P}_2} = A^R_{\mathcal{P}_2}$. For readability, when $\mathcal{P}_1 \circ^* \mathcal{P}_2$ is initialized with $(s^1, s^1) \in S^{\text{init}}_{\mathcal{P}_1 \circ^* \mathcal{P}_2}$, we write $\mathcal{P}_1(s^1) \circ^* \mathcal{P}_2(s^2)$ instead of $(\mathcal{P}_1 \circ^* \mathcal{P}_2)\left((s^1, s^2)\right)$.

3 Continual Mechanisms

In this section, we introduce *continual mechanisms (CMs)* and propose a general definition of differential privacy (DP) for them. This definition unifies and extends DP for IMs and DP under continual observation. Its generality and simplicity allow us to prove (filter) concurrent composition theorems for CMs without requiring complicated proofs. In the full version, we apply these theorems to simplify existing analyses and strengthen results in the continual observation setting.

This section is based on two crucial observations: (i) A message, as sent or received by IMs and IPMs, is a general concept capable of encoding various types of instructions, and (ii) the essence of DP is to protect the privacy of a single bit distinguishing neighboring states, sequences, etc. The second observation has frequently appeared in the DP literature and was used in [18] to model adaptivity in continual observation. Using these key observations, we reformulate DP for IMs and continual observation, and introduce continual mechanisms with a formal DP definition for them.

An Alternative Definition of DP for IMs. By Definition 2.3, an IM $\mathcal{M}$ with initial state space $S^{\text{init}}_{\mathcal{M}}$ is (ϵ, δ)-DP w.r.t. a neighboring relation $\sim$ if for every two initial states $s_0, s_1 \in S^{\text{init}}_{\mathcal{M}}$ such

[11] In [13], the definitions of IMs and IPMs are more restrictive. Specifically, for any IM $\mathcal{M}$, $S_{\mathcal{M}} = Q_{\mathcal{M}} = A_{\mathcal{M}} = \{0,1\}^*$, and for any IPM $\mathcal{P}$, $S_{\mathcal{P}} = M_{\mathcal{P}} = \{0,1\}^*$; however, their results do not rely on these restrictions, and the interactive post-processing lemma holds for arbitrary sets $S_{\mathcal{M}}$, $Q_{\mathcal{M}}$, $A_{\mathcal{M}}$, $S_{\mathcal{P}}$, and $M_{\mathcal{P}}$, which are not necessarily countable.

that $s_0 \sim s_1$ and every adversary $\mathcal{A}$, the views of $\mathcal{A}$ in interaction with $\mathcal{M}(s_0)$ and $\mathcal{M}(s_1)$ are (ϵ, δ)-indistinguishable. This means that for the worst-case (i.e., adversarial) choice of neighboring initial states, an adversary, by adaptively asking queries, should not be able to determine whether $\mathcal{M}$ is running on s_0 or s_1. In an alternative definition, an adversary sends a pair of neighboring initial states (s_0, s_1) as its first message and adaptively asks a sequence of queries $q_1, q_2, \ldots$ in the subsequent messages. In this setting, $\mathcal{M}$ receives one of s_0 and s_1 as its first message and responds with an acknowledgment message $\top$. Thereafter, $\mathcal{M}$ answers queries based on the received state, and the goal of the adversary is to determine whether $\mathcal{M}$ was initialized with s_0 or s_1. Here the actual initial state of $\mathcal{M}$ (not the one in the first message) contains no private information and acts only as an empty memory, so $S^{\text{init}}_{\mathcal{M}}$ consists of a single fixed state.

Formally, we restrict adversaries to sending a pair of neighboring initial states (s_0, s_1) as the first message and pairs of identical queries (q_i, q_i) thereafter. We introduce an IPM $\mathcal{I}$ with $S^{\text{init}}_{\mathcal{I}} = \{0, 1\}$ placed between the adversary and $\mathcal{M}$ to filter these pairs according to its initial state. For $b \in \{0, 1\}$, in the interaction between $\mathcal{A}$ and $\mathcal{I}(b) \circ^* \mathcal{M}$, when $\mathcal{A}$ sends (m_0, m_1), $\mathcal{I}(b)$ forwards m_b to $\mathcal{M}$ and returns $\mathcal{M}$'s response unchanged to $\mathcal{A}$. With this formulation, $\mathcal{M}$ is (ϵ, δ)-DP if for every such adversary $\text{View}(\mathcal{A}, \mathcal{I}(0) \circ^* \mathcal{M})$ and $\text{View}(\mathcal{A}, \mathcal{I}(1) \circ^* \mathcal{M})$ are (ϵ, δ)-indistinguishable.

Finally, we enforce the above adversary restrictions via another IPM, the *verifier* $\mathcal{V}$. The verifier checks that the first message is a pair of neighboring datasets and that subsequent messages are of the form (q, q); only if these checks pass are the messages forwarded to $\mathcal{I}$.

An Alternative Definition of DP under Continual Observation. Dwork, Naor, Pitassi, and Rothblum [9] introduced *differential privacy under continual observation*, where the dataset is not fixed at the initialization time and evolves over time. At each step, the mechanism receives either an empty record or a data record; if the record is non-empty, it is added to the dataset. The mechanism then answers a predetermined query about the current dataset.

To capture "event-level privacy" (where the privacy unit is a single record), Dwork et al. [9] define *neighboring* as a binary relation on sequences of data and empty records: two sequences are neighboring if they are identical except for the presence or absence of a record in at most one step. They called a mechanism (ϵ, δ)*-DP under continual observation* if the distributions of answer sequences are (ϵ, δ)-indistinguishable for every two neighboring input sequences. This definition does not allow data sequences chosen adaptively based on released answers.

Jain, Raskhodnikova, Sivakumar, and Smith [18] formalized DP against adaptive adversaries in this model: An adaptive adversary $\mathcal{A}$ chooses a single (data or empty) record at every step, except for one *challenge* step, where $\mathcal{A}$ sends a pair of records. They called a mechanism $\mathcal{M}$ (ϵ, δ)*-DP against adaptive adversaries* if for every such $\mathcal{A}$, the adversary's views—its internal randomness and message transcript—are (ϵ, δ)-indistinguishable when the first or the second element of the pair in the challenge step is handed to $\mathcal{M}$. Denisov, McMahan, Rush, Smith and Thakurta [4] generalize this definition by allowing the adversary to send a pair of (data or empty) records at *every* step, provided the sequences of first and second records are neighbors under an arbitrary relation $\sim$ on sequences of records. A mechanism $\mathcal{M}$ is said to satisfy (ϵ, δ)-DP against an adaptive adversary if, for every adversary $\mathcal{A}_\sim$ adaptively sending pairs forming $\sim$-neighbor sequences, the views of $\mathcal{A}_\sim$ are (ϵ, δ)-indistinguishable when $\mathcal{M}$ receives the first versus the second elements.

A mechanism $\mathcal{M}$ in the continual observation model can be viewed as an IM with a single initial state, representing an empty dataset. The input message set of $\mathcal{M}$ consists of the empty record and all possible data records. Under this view, the above definition can be restated as: for every adversary $\mathcal{A}_\sim$, the random variables $\text{View}(\mathcal{A}_\sim, \mathcal{I}(0) \circ^* \mathcal{M})$ and $\text{View}(\mathcal{A}_\sim, \mathcal{I}[1] \circ^* \mathcal{M})$ are (ϵ, δ)-indistinguishable, where $\mathcal{I}$ is the IPM defined earlier. As before, we enforce the restriction on the adversary via another IPM $\mathcal{V}$.

In this formulation, every adversary message is a pair of data or empty records, and the sequences of first and second elements are neighbors. Recall that in the alternative definition of DP for IMs, the first adversary message must be a pair of neighboring initial datasets, while subsequent messages must be pairs of identical queries. We next generalize the DP definition by allowing arbitrary conditions on the sequence of pairs sent by adversaries, without restricting the messages' content.

Introducing CMs. As discussed in the alternative formulation of IMs, we can assume that the initial state space of an IM is a singleton and encode its initial state as a message. A continual mechanism is defined as follows:

Definition 3.1 (Continual Mechanism). *A* continual mechanism (CM) *is an interactive mechanism with a singleton initial state space.*

To the best of our knowledge, all mechanisms in DP literature can be represented as CMs:

- Every NIM $\mathcal{N}$, such as the *Laplace mechanism*, can be modeled as a CM that receives a dataset x as its first message and outputs $\mathcal{N}(x)$. Upon any subsequent input message, the CM halts.
- Every IM $\mathcal{M}$ with an initial state space $S_{\mathcal{M}}^{\text{init}}$ can be modeled as a CM that receives $s \in S_{\mathcal{M}}^{\text{init}}$ as its first message and responds with an acknowledgment message $\top$. This CM then answers the next input messages the same as $\mathcal{M}(s)$.
- Mechanisms such as the Sparse Vector Technique (SVT) [10, 14, 22] and the continual counter [2, 3, 7, 8] in the streaming settings can also be represented as CMs receiving two types of messages: question queries, requesting information about the current dataset, and data-update queries, consisting of instructions for modifying the dataset.

DP for CMs. To formulate the general definition of DP for a CM $\mathcal{M}$, we introduce two IPMs: an identifier $\mathcal{I}$ and a verifier $\mathcal{V}$. As described earlier, an identifier receives a secret bit b as its initial state and maps messages of the form (m_0, m_1) to m_b. To avoid defining a new family of adversaries for each problem or model, we use verifiers. A verifier $\mathcal{V}$ with a *verification function* f, denoted by $\mathcal{V}[f]$, stands between an adversary $\mathcal{A}$ and the IM $\mathcal{I} \circ^* \mathcal{M}$. Upon receiving a message from $\mathcal{A}$, $\mathcal{V}[f]$ applies f to check whether the sequence of pairs sent by the adversary so far satisfies specific properties. Comparing the sequences formed by the first and second elements of the pairs, f can encode a neighboring relation. If the sequences satisfy the properties, $\mathcal{V}[f]$ forwards $\mathcal{A}$'s message to $\mathcal{I} \circ^* \mathcal{M}$; otherwise, it halts the communication.

Definition 3.2 (Identifier $\mathcal{I}$). *An* identifier $\mathcal{I}$ *is an IPM with state space $S_{\mathcal{I}}$ and initial state space $S_{\mathcal{I}}^{init} = \{0, 1\}$. $\mathcal{I}$ stands between two IMs, called the left and right mechanisms. Messages from the right mechanism are unrestricted, while messages from the left must have the form (m_0, m_1). Upon receiving (m_0, m_1) from the left mechanism, $\mathcal{I}$ with initial state $b \in \{0, 1\}$ forwards m_b to the right mechanism. $\mathcal{I}(b)$ forwards the message from the right mechanism unchanged to the left mechanism.*

Definition 3.3 (Verification Function). *A verification function f maps a sequence of messages to $\top$ or $\bot$. A sequence $m_1, \ldots, m_k$ is f-valid if $f(m_1, \ldots, m_k) = \top$.*

Remark 3.4. *Since the set of all possible messages does not exist, we assume that every message space considered here is a subset of a universal set $\mathcal{U}$ (e.g., $\{0, 1\}^*$ in [13]). Verification functions therefore have domain $\mathcal{U}^*$.*

Definition 3.5 (Verifier $\mathcal{V}$). *For a verification function f, a verifier $\mathcal{V}[f]$ is an IPM with singleton initial state space $S_{\mathcal{V}[f]}^{init} = \{()\}$, where $()$ denotes the empty sequence. Hence, we denote $\mathcal{V}[f]\,(())$ simply by $\mathcal{V}[f]$. The IPM $\mathcal{V}[f]$ stands between two IMs, called the left and right mechanisms. Its state is the sequence of messages received from the left mechanism so far. Upon receiving a message m from the left, $\mathcal{V}[f]$ appends m to its state. If the resulting sequence is f-valid, m is forwarded to the*

right mechanism; otherwise, communication halts. Messages from the right mechanism are forwarded unchanged to the left.

By definition, for every CM $\mathcal{M}$ and verification function f, the mechanism $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ is an IM with two possible initial states. Since $\mathcal{I} \circ^* \mathcal{M}$ accepts messages in $Q_{\mathcal{M}} \times Q_{\mathcal{M}}$, we implicitly assume that f returns $\perp$ whenever a message in its input sequence is not in $Q_{\mathcal{M}} \times Q_{\mathcal{M}}$. This implies that $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ can interact with any IM. Thus, in the following definition of DP for CMs, we generally consider all adversaries, without restricting the adversary family:

Definition 3.6 ((ϵ, δ)-DP for CMs). *Let $\mathcal{M}$ be a CM and f a verification function. We say that $\mathcal{M}$ is (ϵ, δ)-DP w.r.t. f against adaptive adversaries if for every adversary $\mathcal{A}$, the random variables* $\mathrm{View}(\mathcal{A}, \mathcal{V}[f] \circ^* \mathcal{I}(0) \circ^* \mathcal{M})$ *and* $\mathrm{View}(\mathcal{A}, \mathcal{V}[f] \circ^* \mathcal{I}(1) \circ^* \mathcal{M})$ *are (ϵ, δ)-indistinguishable.*

Let $S^{\text{init}}_{\mathcal{M}} = \{s^{\text{init}}\}$ be the initial state space of the CM $\mathcal{M}$ in Definition 3.6. The initial state spaces of $\mathcal{I}$ and $\mathcal{V}[f]$ are $\{0, 1\}$ and $()$, respectively, so the initial state space of $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ is $\{((), 0, s^{\text{init}}), ((), 1, s^{\text{init}})\}$. Define a universal relation $\sim_{01}$ on this set where every two elements are neighboring. Comparing Definitions 3.6 and 2.3 yields the following corollary.

Corollary 3.7. *A CM $\mathcal{M}$ is (ϵ, δ)-DP w.r.t. a verification function f against adaptive adversaries if and only if the IM $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ is (ϵ, δ)-DP w.r.t. the neighboring relation $\sim_{01}$ against adaptive adversaries.*

CMs vs. IMs. CMs are a special case of IMs with a singleton initial state space, and conversely IMs can be represented as CMs. Both CMs and IMs are defined by a randomized function mapping a state and query message to a new state and answer message, where messages are a general abstraction. The main difference lies in the DP definition associated with each. By Definition 2.1, IMs may have multiple initial states and DP is defined via neighboring initial states. In contrast, by Definition 3.6, CMs have a singleton initial state, and a verifier restricts the adversary to sending pairs whose first and second sequences satisfy specified properties. The definition of DP for CMs is based on a bit held by the identifier.

4 Concurrent Composition of Continual Mechanisms

Composition of NIMs. For $1 \le i \le k$, let $\mathcal{N}_i$ be an NIM with input domain $\mathcal{X}_i$, and let $\sim_i$ be a neighboring relation on $\mathcal{X}_i$. The *composition of the NIMs* $\mathcal{N}_1, \ldots, \mathcal{N}_k$ is represented by an NIM $\mathrm{Comp}[\mathcal{N}_1, \ldots, \mathcal{N}_k]$ with the domain $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, which on input $x = (x_1, \ldots, x_k)$ outputs

$$\mathrm{Comp}[\mathcal{N}_1, \ldots, \mathcal{N}_k](x) = (\mathcal{N}_1(x_1), \ldots, \mathcal{N}_k(x_k)).$$

Define a neighboring relation $\sim$ on $\mathcal{X}$ as $x \sim x'$ iff $x_i \sim_i x'_i$ for all i. The composition of $\mathcal{N}_1, \ldots, \mathcal{N}_k$ is said to be (ϵ, δ)-DP if the NIM $\mathrm{Comp}[\mathcal{N}_1, \ldots, \mathcal{N}_k]$ is (ϵ, δ)-DP w.r.t. $\sim$ (see Definition 1.2).

Concurrent Composition of IMs. Vadhan and Wang [26] extended the composition of NIMs to the *concurrent composition of IMs*, where an adversary interacts simultaneously with k IMs $\mathcal{M}_1, \ldots, \mathcal{M}_k$, adaptively generating messages and selecting their recipient $\mathcal{M}_j$ based on all the previous messages exchanged between the adversary and the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$. They formalize this using an IM $\mathrm{ConComp}[\mathcal{M}_1, \ldots, \mathcal{M}_k]$ that runs $\mathcal{M}_1, \ldots, \mathcal{M}_k$ internally and, on input (q, j), forwards q to $\mathcal{M}_j$ and returns its response. They show that if each IM $\mathcal{M}_i$ is ϵ_i-DP w.r.t. $\sim_i$ and the composition of NIMs $\mathrm{RR}_{\epsilon_1,0}, \ldots, \mathrm{RR}_{\epsilon_k,0}$ is (ϵ, δ)-DP, then $\mathrm{ConComp}[\mathcal{M}_1, \ldots, \mathcal{M}_k]$ is also (ϵ, δ)-DP w.r.t. a neighbor relation $\sim$, defined as above for non-interactive composition, against an adaptive adversary. Lyu [23] and Vadhan and Zhang [27] extend this result for (ϵ_i, δ_i)-DP IMs.

In the concurrent composition of IMs, queries to each mechanism may depend not only on its past responses but also on those of the other mechanisms. Nevertheless, [23, 27] show that such concurrency and adaptivity do *not* compromise privacy guarantees, and all (ϵ, δ)-DP composition

theorems for NIMs—e.g., basic [8], advanced [11], and optimal [20] composition theorems—also apply to the concurrent composition of IMs.

Concurrent Composition of CMs. We extend the concurrent composition of IMs to CMs, where an adversary can create new CMs or send pairs of queries to existing ones. Depending on the composition type, the created mechanisms are constrained, and the sequence of query pairs to each CM satisfies certain properties. Based on a secret bit $b \in \{0, 1\}$, each CM receives either the first or second element of every query pair. To formalize this, we define a CM ExtConComp, generalizing ConComp [26] to allow mechanism creation in addition to queries.

A *creation query* $(\mathcal{M}, s, \epsilon, \delta, f)$ is a message that requests the creation of a continual mechanism $\mathcal{M}$ with the initial state s and asserts that $\mathcal{M}$ is (ϵ, δ)-DP w.r.t. the verification function f. The mechanism ExtConComp accepts creation queries and messages of the form (q, i), where q is a query for the i-th created mechanism:

Definition 4.1. Extended concurrent composition, ExtConComp, *is a CM receiving two types of messages: (i) a creation query $(\mathcal{M}, s, \epsilon, \delta, f)$, where $\mathcal{M}$ is a CM with the (single) initial state s that is (ϵ, δ)-DP w.r.t. f, and (ii) a query-index pair (q, i), where $i \in \mathbb{N}$ and q is any message. Upon receiving $(\mathcal{M}, s, \epsilon, \delta, f)$, ExtConComp creates an instance of $\mathcal{M}$ with the initial state s. Upon (q, i), it forwards q to the i-th created mechanism and returns its response, halting if fewer than i mechanisms exist. The state of ExtConComp is the sequence of pairs $(\mathcal{M}_i, s_i')$ of created mechanisms and their current states. The single initial state space of ExtConComp is the empty sequence $()$. Both query and answer sets of ExtConComp equal the universal message set $\mathcal{U}$ from Remark 3.4.*[12]

Different variants of concurrent composition of CMs impose different constraints on created mechanisms and queries. We encode such constraints via a verification function f controlling the communication of the adversary with ExtConComp:

Definition 4.2 (f-Concurrent Composition of CMs). *Let f be a verification function. The f-*concurrent composition of CMs *is the IM $\mathcal{V}[f] \circ^* \mathcal{I} \circ^*$ ExtConComp. For $\epsilon \geq 0$ and $0 \leq \delta \leq 1$,* the f-concurrent composition of CMs is (ϵ, δ)-DP against adaptive adversaries *if the IM $\mathcal{V}[f] \circ^* \mathcal{I} \circ^*$ ExtConComp is (ϵ, δ)-DP w.r.t. $\sim_{01}$, i.e., ExtConComp is (ϵ, δ)-DP w.r.t. f against adaptive adversaries.*

Although the verification function f varies across compositions, it must ensure that adversary messages are compatible with $\mathcal{I} \circ^*$ExtConComp. The adversary either creates a new CM or sends a query pair to an existing one. Since creation is independent of $\mathcal{I}$'s initial state, the adversary must send a pair of identical creation queries to $\mathcal{V}[f]$. Moreover, to query q_0 or q_1 from the i-th mechanism depending on $\mathcal{I}$'s state, the adversary must send $\big((q_0, i), (q_1, i)\big)$ to $\mathcal{V}[f]$. We formalize these common constraints below. Throughout the paper, we denote a pair of identical creation queries $(\mathcal{M}, s, \epsilon, \delta, f)$ by $(\mathcal{M}, s, \epsilon, \delta, f)^2$.

Definition 4.3 (Suitable Sequence). *A message sequence $m_1, \ldots, m_k$ is* suitable *if for every $1 \leq j \leq k$:*

(i) Each m_j is either (a) a pair of identical creation queries $(\mathcal{M}, s, \epsilon, \delta, f)^2$, or (b) a query-pair $\big((q_0, \ell), (q_1, \ell)\big)$ with common index ℓ.

(ii) If $m_j = \big((q_0, \ell), (q_1, \ell)\big)$, then $m_1, \ldots, m_{j-1}$ contain at least ℓ creation pairs.

(iii) If $m_j = \big((q_0, \ell), (q_1, \ell)\big)$ and $\mathcal{M}_\ell$ is the ℓ-th mechanism requested to be created in $m_1, \ldots, m_{j-1}$, then $q_0, q_1 \in Q_{\mathcal{M}_\ell}$.

(iv) If $m_j = (\mathcal{M}, s, \epsilon, \delta, f)^2$, then $\mathcal{M}$ is a CM with (single) initial state s that is (ϵ, δ)-DP w.r.t. f.

[12] By Russell's paradox, a set of all CMs cannot exist. Thus creation queries select mechanisms and verification functions from well-defined (possibly infinite) sets so that Definition 4.1 and the query/answer sets of ExtConComp are well defined. The set may include ExtConComp itself, allowing recursive creation.

(v) If $m_j = (\mathcal{M}, s, \epsilon, \delta, f)^2$ is the ℓ-th creation pair in $m_1, \ldots, m_{j-1}$, then the sequence of query pairs (q_0, q_1) from messages $\big((q_0, \ell), (q_1, \ell)\big)$ in $m_1, \ldots, m_k$ is f-valid.

(ϵ, δ)-DP IMs and $\mathrm{RR}_{\epsilon,\delta}$ To prove that the concurrent composition of IMs inherits the privacy guarantees of NIM composition, Lyu [23] and Vadhan and Zhang [27] show that for every (ϵ, δ)-DP IM $\mathcal{M}$ and neighboring initial states s_0, s_1, there exists an IPM $\mathcal{T}$ (with a single initial state) such that for each $b \in 0, 1$, the mechanisms $\mathcal{T} \circ^* \mathrm{RR}_{\epsilon,\delta}(b)$ and $\mathcal{M}(s_b)$ are equivalent. Although not explicitly stated, the construction of $\mathcal{T}$ in [23] requires finite answer sets for $\mathcal{M}(s_0)$ and $\mathcal{M}(s_1)$. Vadhan and Zhang [27] additionally assume a finite query space. We instead require the following property:

Definition 4.4. *An IM $\mathcal{M}$ has* discrete answer distributions *if $A_{\mathcal{M}}$ is countable, and for every initial state s, integer $t \in \mathbb{N}$, history $(q_1, a_1, \ldots, q_{t-1}, a_{t-1}) \in (Q_{\mathcal{M}} \times A_{\mathcal{M}})^{t-1}$, and new query $q_t \in Q_{\mathcal{M}}$, the distribution of $\mathcal{M}(s)$'s response to q_t conditioned on having the history $(q_1, \ldots, a_{t-1})$ has a probability mass function (PMF).*

This is generally not a restrictive assumption in differential privacy. In practice, every distribution sampled computationally is inherently discrete due to the binary nature of computers and the finiteness of memory. Theoretically, real-valued outputs can be discretized by rounding to multiples of a sufficiently small constant α, a post-processing that preserves privacy and negligibly affects accuracy. An example of this approach is provided in the full version. Lyu [23] also assumes a fixed upper bound c on the number of queries answered by $\mathcal{M}$. In the full version, we show how to remove this bound and allow countably infinite answer sets.

Lemma 4.5 (slight extension of [23, 27]). *For $\epsilon > 0$ and $0 \le \delta \le 1$, let $\mathcal{M}$ be a CM that is (ϵ, δ)-DP w.r.t. a verification function f. If $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ has discrete answer distributions, then there exists an IPM $\mathcal{T}$ with a single initial state such that for each $b \in \{0, 1\}$, the IMs $\mathcal{T} \circ^* \mathrm{RR}_{\epsilon,\delta}(b)$ and $\mathcal{V}[f] \circ^* \mathcal{I}(b) \circ^* \mathcal{M}$ are equivalent.*

Concurrent Composition of CMs with fixed parameters. In the *concurrent composition of CMs with fixed parameters* $(\epsilon_1, \delta_1), \ldots, (\epsilon_k, \delta_k)$, an adversary selects and creates k CMs $\mathcal{M}_1, \ldots, \mathcal{M}_k$ adaptively over time, where each mechanism $\mathcal{M}_i$ is (ϵ_i, δ_i)-DP w.r.t. a verification function f_i, and adaptively issues query pairs for them such that the sequence of query pairs for each $\mathcal{M}_i$ is f_i-valid. (Creating $\mathcal{M}_i$ means instantiating $\mathcal{M}_i(s_i)$, where s_i is its unique initial state.) For instance, the adversary may first create a CM $\mathcal{M}_3$ that is (ϵ_3, δ_3)-DP w.r.t. a verification function f_3, interact with it for some time, and then select a mechanism $\mathcal{M}_1$ from the class of (ϵ_1, δ_1)-DP CMs. Note that, in this concurrent composition, the multiset of privacy parameters $(\epsilon_1, \delta_1), \ldots, (\epsilon_k, \delta_k)$ is fixed in advance, but the CMs $\mathcal{M}_i$ and verification functions f_i are chosen by the adversary.

Given a secret bit $b \in \{0, 1\}$, each CM receives either the first or second element of every query pair addressed to it, and the adversary attempts to infer b from the responses of all k mechanisms. For technical reasons, each $\mathcal{V}[f_i] \circ^* \mathcal{I} \circ^* \mathcal{M}_i$ must have discrete answer distributions. To formalize this concurrent composition, we define a verification function $f^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$ enforcing these restrictions.

Definition 4.6 ($f^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$). *For every $t \in \mathbb{N}$ and message sequence $m_1, \ldots, m_t$, the function $f^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$ returns $\top$ if and only if all of the following conditions hold:*

(i) The sequence $m_1, \ldots, m_t$ is suitable.

(ii) There are at most k pairs of identical creation queries in $m_1, \ldots, m_t$. Let $\mathcal{M}'_i$ be the i-th created mechanism with privacy parameters (ϵ'_i, δ'_i) and verification function f'_i. Then $\mathcal{V}[f'_i] \circ^* \mathcal{I} \circ^* \mathcal{M}'_i$ has discrete answer distributions, and the multiset of the privacy parameters (ϵ'_i, δ'_i) of all created mechanisms is a sub-multiset of the fixed privacy parameters $(\epsilon_1, \delta_1), \ldots, (\epsilon_k, \delta_k)$. (Here (ϵ'_i, δ'_i) are chosen adaptively by the adversary, while (ϵ_i, δ_i) are predetermined public parameters.)

We refer to the $f^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$-concurrent composition of CMs as the *concurrent composition of CMs with privacy parameters* $((\epsilon_i,\delta_i))_{i=1}^k$. Therefore, this concurrent composition is (ϵ,δ)-DP against adaptive adversaries if ExtConComp is (ϵ,δ)-DP w.r.t. $f^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$ against adaptive adversaries.

Recall that in Section 3, we generalized neighboring relations for IMs to verification functions for CMs. The neighboring initial states for an IM are chosen non-adaptively at initialization. However, the choice of messages forming a valid sequence w.r.t. a verification function is adaptive and happens over time. That is, an adversary generates two different query sequences for a CM, and the choice of differing sequences is made adaptively throughout the interaction.

In the concurrent composition of IMs, queries to each mechanism may depend on its past responses and those of the others, but *the choice of neighboring initial states is non-adaptive.* However, in the concurrent composition of CMs, *the choice of differences between query sequences, i.e., the neighboring query sequences, is adaptive*, based not only on the responses of the corresponding mechanism but also on the responses of other mechanisms. (CMs have a single fixed initial state; hence, no choice of initial state is involved.) Despite this additional adaptivity, the following theorem shows that the privacy results of [23, 26] for concurrent composition of k IMs extend to the concurrent composition of CMs with k fixed privacy parameters.

Theorem 4.7. *For $\epsilon_1,\ldots,\epsilon_k > 0$, $\epsilon \geq 0$ and $0 \leq \delta,\delta_1,\ldots,\delta_k \leq 1$, if the composition of NIMs $\mathrm{RR}_{\epsilon_1,\delta_1},\ldots,\mathrm{RR}_{\epsilon_k,\delta_k}$ is (ϵ,δ)-DP, then the concurrent composition of k CMs with the privacy parameters $((\epsilon_i,\delta_i))_{i=1}^k$ is also (ϵ,δ)-DP against adaptive adversaries.*

The proof of Theorem 4.7, provided in the full version, simulates each created CM $\mathcal{M}'_j$ with privacy parameters (ϵ'_j,δ'_j) by the IPM $\mathcal{T}_j$ from Lemma 4.5. By that lemma, for each $b \in \{0,1\}$, the mechanism $\mathcal{T}_j$ interacts with $\mathrm{RR}_{\epsilon'_j,\delta'_j}(b)$ once and simulates $\mathcal{V}[f'_j] \circ^* \mathcal{I}(b) \circ^* \mathcal{M}'_j$ identically.

5 Concurrent Parallel Composition of CMs

The *concurrent k-sparse parallel composition* extends the concurrent composition of CMs with k fixed parameters by allowing an *arbitrary* number of CMs. Given a multiset `params` of k privacy parameters $(\epsilon_1,\delta_1),\ldots,(\epsilon_k,\delta_k)$, the adversary can create unlimited CMs with arbitrary parameters but must send pairs of identical queries to all except at most k mechanisms. Let $\mathcal{M}_1,\ldots,\mathcal{M}_k$ denote these exceptional mechanisms. Each $\mathcal{M}_i$ must be (ϵ_i,δ_i)-DP w.r.t. a verification function f_i, and its query-pair sequence must be f_i-valid. To formalize this composition, we define the verification function $f_\infty^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$, where the subscript ∞ indicates that arbitrarily many CMs may be created.

Definition 5.1 ($f_\infty^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$). *For every $t \in \mathbb{N}$ and sequence of messages $m_1,\ldots,m_t$, the function $f_\infty^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$ returns $\top$ if and only if all of the following conditions hold:*

(i) The sequence $m_1,\ldots,m_t$ is suitable.

(ii) Let $\mathcal{M}'_i$ be the i-th created mechanism with privacy parameters (ϵ'_i,δ'_i) and verification function f'_i. Let $\mathcal{J}$ be the set of indices j for which there exists a message $\big((q_0,j),(q_1,j)\big)$ with $q_0 \neq q_1$. Let $\mathtt{params}_{\mathcal{J}}$ be the multiset of the privacy parameters (ϵ'_j,δ'_j) for each $j \in \mathcal{J}$. Then $\mathtt{params}_{\mathcal{J}}$ must be a sub-multiset of the fixed parameters $(\epsilon_1,\delta_1),\ldots,(\epsilon_k,\delta_k)$ (hence $|\mathcal{J}| \leq k$).

We call the $f_\infty^{\epsilon_1,\delta_1,\ldots,\epsilon_k,\delta_k}$-concurrent composition of CMs the *concurrent k-sparse parallel composition of CMs with privacy parameters* $((\epsilon_i,\delta_i))_{i=1}^k$. The following theorem shows that for $k=1$, $\epsilon_1 = 0$, and any $\delta_1 \in (0,1]$, the f_∞^{0,δ_1}-concurrent composition of CMs is not differentially private:

Theorem 5.2. *For every $\delta \in (0,1]$, there exists an adversary $\mathcal{A}_\delta$ s.t. the views of $\mathcal{A}_\delta$ interacting with $\mathcal{V}[f^{0,\delta}] \circ^* \mathcal{I}(0) \circ^* ExtConComp$ and $\mathcal{V}[f^{0,\delta}] \circ^* \mathcal{I}(1) \circ^* ExtConComp$ have disjoint supports.*

The proof idea is to construct a mechanism $\mathcal{M}_\delta$ that takes a bit as a query. On the first query (independent of its value), it outputs $\perp$ with probability δ and $\top$ otherwise. On the second query, it returns the query bit if the first answer was $\perp$, and otherwise behaves as on the first query. The adversary repeatedly creates instances of $\mathcal{M}_\delta$ and sends the query pair $(0,0)$ until one outputs $\perp$. It then sends $(0,1)$ to that instance; the response reveals the secret bit used by the identifier. The formal proof is provided in the full version.

In this counterexample, the adversary runs unboundedly many $(0,\delta)$-DP CMs but sends non-identical queries to only one, revealing the secret bit with probability 1. Each mechanism fails with probability δ, and these failures accumulate to full disclosure. To prevent such leakage, we restrict the second privacy parameters δ'_j of all created mechanisms to satisfy an additional condition.

Definition 5.3 ($f^{\epsilon_1,\dots,\epsilon_k}_{\infty,\delta'}$). *For every $t \in \mathbb{N}$ and every input sequence of messages $m_1,\dots,m_t$, the function $f^{\epsilon_1,\dots,\epsilon_k}_{\infty,\delta'}$ returns $\top$ if and only if all of the following conditions hold:*

(i) The sequence $m_1,\dots,m_t$ is suitable.

(ii) Let $\mathcal{M}'_i$ be the i-th created mechanism with privacy parameters (ϵ'_i,δ'_i). Let $\mathcal{J}$ be the set of indices j for which there exists a message $\big((q_0,j),(q_1,j)\big)$ with $q_0 \neq q_1$. Let $\mathtt{params}_{\mathcal{J}}$ be the multiset of ϵ'_j for $j \in \mathcal{J}$. Then $\mathtt{params}_{\mathcal{J}}$ must be a sub-multiset of the fixed privacy parameters $\epsilon_1,\dots,\epsilon_k$.

(iii) $1-\prod_j(1-\delta'_j) \le \delta'$, where the product is taken over the indices of the created mechanisms.

Theorem 5.4. *For $\epsilon_1,\dots,\epsilon_k > 0$, $\epsilon \ge 0$ and $0 \le \delta,\delta' \le 1$, if the composition of NIMs $\mathrm{RR}_{\epsilon_1,0},\dots,\mathrm{RR}_{\epsilon_k,0}$ is (ϵ,δ)-DP, then the $f^{\epsilon_1,\dots,\epsilon_k}_{\infty,\delta'}$-concurrent composition of CMs is $(\epsilon, 1-(1-\delta)(1-\delta'))$-DP against adaptive adversaries.*

It is known that the composition of $\mathrm{RR}_{\epsilon_1,0},\dots,\mathrm{RR}_{\epsilon_k,0}$ is $\sum_{i=1}^k \epsilon_i$-DP. Hence, letting $\epsilon_1,\dots,\epsilon_k \to 0$ implies that the $f^{0,\dots,0}_{\infty,\delta'}$-concurrent composition of CMs is $(0,\delta')$-DP. The proof of Theorem 5.2 shows this composition cannot be $(0,\delta'')$-DP for $\delta'' < \delta'$, so Theorem 5.4 is *tight*.

Setting $\delta' = 0$ in Theorem 5.4 requires all created mechanisms to be purely differentially private.

Corollary 5.5. *For $\epsilon_1,\dots,\epsilon_k > 0$, $\epsilon \ge 0$ and $0 \le \delta \le 1$, if the composition of NIMs $\mathrm{RR}_{\epsilon_1,0},\dots,\mathrm{RR}_{\epsilon_k,0}$ is (ϵ,δ)-DP, then the $f^{\epsilon_1,\dots,\epsilon_k}_{\infty,0}$-concurrent composition of CMs is (ϵ,δ)-DP against adaptive adversaries. That is, the concurrent k-sparse parallel composition of purely differentially private CMs with privacy parameters $((\epsilon_i,0))_{i=1}^k$ is (ϵ,δ)-DP against adaptive adversaries.*

Recall that in the proof of Theorem 4.7, each created CM $\mathcal{M}'_j$ with privacy parameters (ϵ'_j,δ'_j) is simulated by an IPM $\mathcal{T}_j$ that interacts with $\mathrm{RR}_{\epsilon'_j,\delta'_j}$ once. We next construct a new $\mathcal{T}_j$ interacting with $\mathrm{IRR}_{\epsilon'_j,\delta'_j}$ instead of $\mathrm{RR}_{\epsilon'_j,\delta'_j}$ and satisfying additional properties required for the proof of Theorem 5.4. The proof of the following lemma is provided in the full version.

Lemma 5.6. *For $\epsilon > 0$ and $0 \le \delta \le 1$, let $\mathcal{M} : S_{\mathcal{M}} \times Q_{\mathcal{M}} \to S_{\mathcal{M}} \times A_{\mathcal{M}}$ be an (ϵ,δ)-DP IM w.r.t. a neighbor relation $\sim$, and suppose $\mathcal{M}$ has discrete answer distributions. Then for every two neighboring initial states s_0 and s_1, there exists an IPM $\mathcal{T}$ such that for each $b \in \{0,1\}$, (i) The IMs $\mathcal{M}(s_b)$ and $\mathcal{T} \circ^* \mathrm{IRR}_{\epsilon,\delta}(b)$ are equivalent, and (ii) for every $k \in \mathbb{N}$ and query sequence $(q_1,\dots,q_k) \in Q^k_{\mathcal{M}}$, if the answer distributions of $\mathcal{M}(s_0)$ and $\mathcal{M}(s_1)$ to $(q_1,\dots,q_k)$ are identical, then when $\mathcal{T} \circ^* \mathrm{IRR}_{\epsilon,\delta}(b)$ receives the queries $q_1,\dots,q_k$, the IPM $\mathcal{T}$ interacts with $\mathrm{IRR}_{\epsilon,\delta}(b)$ only once.*

Let $\mathcal{M}$ be a CM that is (ϵ,δ)-DP w.r.t. a verification function f. Suppose $\mathcal{V}[f] \circ^* \mathcal{I} \circ^* \mathcal{M}$ has discrete answer distributions. Lemma 5.6 implies that there exists an IPM $\mathcal{T}$ such that, for each $b \in \{0,1\}$, (i) $\mathcal{V}[f] \circ^* \mathcal{I}(b) \circ^* \mathcal{M}$ and $\mathcal{T} \circ^* \mathrm{IRR}_{\epsilon,\delta}(b)$ are equivalent, and (ii) for every $k \in \mathbb{N}$ and message sequence $m_1,\dots,m_k$ where each message m_j is a pair of identical messages, if $\mathcal{T} \circ^* \mathrm{IRR}_{\epsilon,\delta}(b)$ receives $m_1,\dots,m_k$, then the IPM $\mathcal{T}$ interacts with $\mathrm{IRR}_{\epsilon,\delta}(b)$ at most once. Therefore, by the definition of

$\mathrm{IRR}_{\epsilon,\delta}$, $\mathcal{M}$ can be simulated by post-processing the outputs of two randomized response mechanisms $\mathrm{RR}_{\epsilon,0}$ and $\mathrm{RR}_{0,\delta}$. While simulating $\mathcal{M}$ generally requires both outputs, if the adversary asks only identical query pairs, the output of $\mathrm{RR}_{\epsilon,0}$ is unnecessary.

The high-level proof idea of Theorem 5.4 is to simulate each created CM $\mathcal{M}'_j$ with privacy parameters (ϵ'_j, δ'_j) by the IPM $\mathcal{T}_j$ from Lemma 5.6. By the definition of $f_{\infty,\delta'}^{\epsilon_1,\ldots,\epsilon_k}$, the multiset of parameters ϵ'_j for mechanisms receiving non-identical query pairs is a sub-multiset of $\epsilon_1, \ldots, \epsilon_k$. For those mechanisms, $\mathcal{T}_j$ requires the outputs of both $\mathrm{RR}_{\epsilon'_j,0}$ and $\mathrm{RR}_{0,\delta'_j}$, while for the rest of the created mechanisms, $\mathcal{T}_j$ only needs $\mathrm{RR}_{0,\delta'_j}$. We prove Theorem 5.4 by separately composing $\mathrm{RR}_{\epsilon_1,0}, \ldots, \mathrm{RR}_{\epsilon_k,0}$ and $\mathrm{RR}_{0,\delta'_1}, \mathrm{RR}_{0,\delta'_2}, \ldots$, and then composing these two sets of composed mechanisms. The full proof is provided in the full version.

Concurrent k-sparse Parallel Composition of IMs Consider a setting where an adversary creates (an unlimited number of) IMs over time while adaptively querying existing ones. When creating an IM $\mathcal{M}_i$ that is (ϵ_i, δ_i)-DP w.r.t. a neighbor relation $\sim_i$, the adversary also selects a pair of neighboring initial states for $\mathcal{M}_i$, depending on the history of queries and answers so far. A secret bit b determines which state in each pair is used: if $b = 0$ all mechanisms use the first state, and if $b = 1$ the second. The adversary aims to guess b.

As discussed in Section 3, each $\mathcal{M}_i$ can be represented by a CM $\mathcal{M}'_i$ (with initial state null) that takes an initial state $s \in S_{\mathcal{M}}^{\mathrm{init}}$ as its first message and then behaves as $\mathcal{M}_i(s)$. If $\mathcal{M}_i$ is DP w.r.t. a neighbor relation $\sim_i$, then the corresponding CM is DP w.r.t. a verification function $f_{\sim_i}$, ensuring that the first message is a pair of $\sim_i$-neighbor initial states and the subsequent messages are pairs of identical queries for $\mathcal{M}_i$. Thus, the concurrent parallel composition of IMs can be viewed as a restricted version of the concurrent parallel composition of CMs where only the first message pair sent to each $\mathcal{M}'_i$ may contain non-identical messages.

Recall that in the counterexample of Theorem 5.2, by sending a pair of identical messages, the adversary leveraged the first outcome of a $(0, \delta)$-DP CM to decide whether to spend its budget on issuing non-identical queries to that mechanism. Intuitively, when $\delta > 0$, there might exist certain "bad first answers" that occur with low probability, but conditioning on them makes the second answer no longer differentially private. By the above discussion, in the concurrent parallel composition of IMs, the adversary cannot exploit such conditioning. Using this observation, we obtain the following theorem; the formal proof is provided in the full version.

Theorem 5.7. *For $\epsilon_1, \ldots, \epsilon_k > 0$, $\epsilon \geq 0$ and $0 \leq \delta, \delta_1, \ldots, \delta_k \leq 1$, if the composition of NIMs $\mathrm{RR}_{\epsilon_1,\delta_1}, \ldots, \mathrm{RR}_{\epsilon_k,\delta_k}$ is (ϵ, δ)-DP, then the concurrent k-sparse parallel composition of IMs with privacy parameters $((\epsilon_i, \delta_i))_{i=1}^k$ is also (ϵ, δ)-DP.*

Acknowledgments

[1]Salil Vadhan was supported by NSF grant BCS-2218803, a grant from the Sloan Foundation, and a Simons Investigator Award. Work began while a Visiting Researcher at the Bocconi University Department of Computing Sciences, supported by Luca Trevisan's ERC Project GA-834861.

[2]Monika Henzinger and Roodabeh Safavi were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019564), and the Austrian Science Fund (FWF) under grants DOI 10.55776/Z422, DOI 10.55776/I5982, and DOI 10.55776/P33775. For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997* (1997), IEEE Computer Society, pp. 394–403.

[2] Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur. 14*, 3 (2011), 26:1–26:24.

[3] Denisov, S., McMahan, B., Rush, K., Smith, A. D., and Thakurta, A. G. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *arXiv preprint arXiv:2202.08312* (2022).

[4] Denisov, S., McMahan, H. B., Rush, J., Smith, A., and Guha Thakurta, A. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *Advances in Neural Information Processing Systems 35* (2022), 5910–5924.

[5] Dhulipala, L., Henzinger, M., Li, G. Z., Liu, Q. C., Sricharan, A., and Zhu, L. Near-optimal differentially private graph algorithms via the multidimensional abovethreshold mechanism. *arXiv preprint arXiv:2508.02182* (2025).

[6] Dinitz, M., Li, G. Z., Liu, Q. C., and Zhou, F. Differentially private matchings. *CoRR abs/2501.00926* (2025).

[7] Dvijotham, K. D., McMahan, H. B., Pillutla, K., Steinke, T., and Thakurta, A. Efficient and near-optimal noise generation for streaming differential privacy. In *Foundations of Computer Science* (2024).

[8] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3* (2006), Springer, pp. 265–284.

[9] Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proc. 42nd STOC* (2010), pp. 715–724.

[10] Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and 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* (2009), pp. 381–390.

[11] Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science* (2010), pp. 51–60.

[12] Ghazi, B., Harrison, C., Hosabettu, A., Kamath, P., Knop, A., Kumar, R., Leeman, E., Manurangsi, P., Raykova, M., Sahu, V., and Schoppmann, P. On the differential privacy and interactivity of privacy sandbox reports. *Proceedings on Privacy Enhancing Technologies (PoPETs) 2025*, 3 (2025), 382–397.

[13] Haney, S., Shoemate, M., Tian, G., Vadhan, S., Vyrros, A., Xu, V., and Zhang, W. Concurrent composition for interactive differential privacy with adaptive privacy-loss parameters. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (2023), pp. 1949–1963.

[14] Hardt, M., and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (Oct 2010), pp. 61–70.

[15] Henzinger, M., and Safavi, R. Securing dynamic data: A primer on differentially private data structures (invited talk). In *33rd Annual European Symposium on Algorithms, ESA 2025, Warsaw, Poland, September 15-17, 2025* (2025), A. Benoit, H. Kaplan, S. Wild, and G. Herman, Eds., vol. 351 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 2:1–2:14.

[16] Henzinger, M., Safavi, R., and Vadhan, S. Concurrent composition for differentially private continual mechanisms. *arXiv preprint arXiv:2411.03299* (2026).

[17] Henzinger, M., Sricharan, A. R., and Steiner, T. A. Differentially private data structures under continual observation for histograms and related queries. *CoRR abs/2302.11341* (2023).

[18] Jain, P., Raskhodnikova, S., Sivakumar, S., and Smith, A. D. The price of differential privacy under continual observation. In *Proc. 40th ICML* (2023), pp. 14654–14678.

[19] Jain, P., Smith, A., and Wagaman, C. Time-aware projections: Truly node-private graph statistics under continual observation*. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024), pp. 127–145.

[20] Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. In *International Conference on Machine Learning (ICML)* (2015), PMLR, pp. 1376–1385.

[21] Kasiviswanathan, S. P., and Smith, A. On the 'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality 6*, 1 (2014).

[22] Lyu, M., Su, D., and Li, N. Understanding the sparse vector technique for differential privacy. *arXiv preprint arXiv:1603.01699* (2016).

[23] Lyu, X. Composition theorems for interactive differential privacy. *Advances in Neural Information Processing Systems 35* (2022), 9700–9712.

[24] Qiu, Y., and Yi, K. Differential privacy on dynamic data. arXiv preprint arXiv:2209.01387, 2022.

[25] Rogers, R. M., Roth, A., Ullman, J., and Vadhan, S. Privacy odometers and filters: Pay-as-you-go composition. *Advances in Neural Information Processing Systems 29* (2016).

[26] Vadhan, S., and Wang, T. Concurrent composition of differential privacy. In *Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part II 19* (2021), Springer, pp. 582–604.

[27] Vadhan, S., and Zhang, W. Concurrent composition theorems for differential privacy. In *55th Annual ACM Symposium on Theory of Computing* (2023).

[28] Whitehouse, J., Ramdas, A., Rogers, R., and Wu, S. Fully-adaptive composition in differential privacy. In *International conference on machine learning* (2023), PMLR, pp. 36990–37007. Available at https://jwhitehouse11.github.io/icml_2023_adaptive.pdf.

Received December 2025; accepted February 2026