Realisation-Level Privacy Filtering

Sophie Taylor, Praneeth Kumar Vippathalla, and Justin P. Coon

***Abstract*—We study differentially private data release, where a database is accessed through successive, possibly adaptive queries and mechanisms. Existing composition theorems and privacy filters combine worst case per-round privacy parameters, leaving room for more refined accounting based on realised leakage, which we term realisation-level accounting. We propose a realisation-level filtering approach to determine stopping times for data releases, and design one such filter. Despite technical challenges arising from conditioning on realisations and stopping time, we prove that the filter guarantees (ϵ, δ)-differential privacy, with ϵ and δ chosen by the data handler. Through numerical evidence, we demonstrate that realisation-level filtering provides a complementary path to better utility beyond mechanism-level methods. Furthermore, our proposed filter applies to arbitrary mechanisms, including those that are badly behaved under Rényi differential privacy.**

I. Introduction

In most privacy preserving applications, a database is subject to successive queries, and is therefore accessed more than once. In modern settings, queries and mechanisms are often adaptive, meaning they depend on previously released outputs. A key example is federated learning, where model training involves repeated access to data. In such systems, the sequence of queries and mechanisms may be generated by an adaptive training procedure. Moreover, each data access incurs a privacy loss, making adaptive privacy accounting essential. Hence, it is crucial that a system designer can quantify how privacy guarantees compose under multiple adaptive mechanism uses. Throughout this work, we take differential privacy (DP) as the privacy notion.

Existing approaches to privacy composition [1]–[7] provide powerful guarantees in many contexts, but can be overly conservative in certain settings. Classical composition theorems bound the cumulative privacy loss without knowledge of mechanism outputs, commonly by combining known per-mechanism parameters [1]–[3]. Notably, Rényi differential privacy (RDP) provides strong composition guarantees by leveraging the distribution of the privacy leakage [4]. In practice, RDP is used as an accounting tool, with privacy guarantees converted back to DP for reporting. These guarantees must be computed in advance and are independent of the realised mechanism outputs. Given a privacy budget, the number of allowable releases must be determined uniformly over all possible mechanism choices, rather than tailored to the specific sequence used. This can lead to very conservative stopping rules in adaptive scenarios.

To tackle the adaptive setting, researchers have proposed the use of privacy filters, which may be DP based [8], [9] or RDP based [9], [10], and keep a running total of privacy loss to adaptively decide when to stop releases to stay within a privacy budget. Existing privacy filters track the privacy loss of the sequence of realised mechanisms by combining their per round parameters. This contrasts classical composition, which operates without knowledge of the particular realisations of adaptively chosen mechanisms. We call this *mechanism-level* accounting. Despite its advantages, mechanism-level accounting relies on worst case privacy parameters, and does not exploit the fact that the realised leakage may be significantly smaller. In this work, we propose a filtering approach that tracks leakage pointwise, operating at the *realisation-level*. While this may appear natural, doing so poses significant technical challenges, as differential privacy can be violated through conditioning on realised outputs. In particular, designing a stopping rule to ensure a privacy guarantee requires accounting for the privacy loss from halting the filter.

In this work, we propose a privacy filtering approach based on realisation-level accounting. We design one such filter and prove that it satisfies (ϵ, δ)-DP. The filter is generally applicable, and does not assume a particular class of mechanism. Finally, we discuss its utility implications in terms of the allowed number of data releases.

II. Privacy Filtering

Consider the adaptive data privacy problem setup in Figure 1. An analyst sends a data request R_1 from the set of allowable requests $\mathcal{R}_1$ to a database X. This is input to a privacy mechanism $\mathcal{M}_1$, which produces a random output

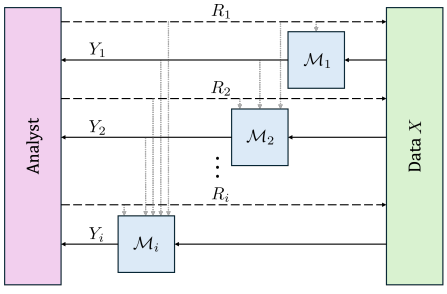

Fig. 1. Adaptive data privacy problem

The authors are with the Department of Engineering Science, University of Oxford, Oxford, U.K (e-mail: sophie.taylor2@balliol.ox.ac.uk; praneeth.vippathalla@eng.ox.ac.uk; justin.coon@eng.ox.ac.uk). This research was funded in whole or in part by the Engineering and Physical Sciences Research Council under grant number EP/W524311/1, and the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-22-1-0070.

$Y_1 = \mathcal{M}_1(R_1, X)$. Given this response, the analyst can make a second request $R_2 \in \mathcal{R}_2$. Importantly, requests can be chosen adaptively, depending on previous outputs. The second data release Y_2 is the output of $\mathcal{M}_2(R_1, R_2, Y_1, X)$. The process continues so that the ith data release Y_i is generated by $\mathcal{M}_i(R_1, \ldots, R_i, Y_1, \ldots, Y_{i-1}, X)$. We assume a fixed series of allowable sets $\mathcal{R}_1, \mathcal{R}_2 \ldots$, and a fixed family of conditional distributions $P_x(Y_1|R_1), P_x(Y_1|R_1, R_2, Y_2), \ldots$ defining the mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots$ a priori; the mechanisms are adaptive in the standard sense that, at each round, the distribution of the output is conditioned on all previous requests and outputs.

We use superscript indexing to denote the first i requests $R^i := (R_1, \ldots, R_i)$ and the first $i-1$ outputs $Y^{i-1} := (Y_1, \ldots, Y_{i-1})$. For $i \geq 1$, we refer to (R^i, Y^{i-1}) as the *partial transcript* and (R^i, Y^i) as the *full transcript* at step i. Conditioned on a realised partial transcript $(R^i, Y^{i-1}) = (r^i, y^{i-1})$ and database $X = x$, the output Y_i is distributed according to $P_x(y_i|r^i, y^{i-1})$. Throughout, the subscript x indicates that the probabilities are conditioned on the database $X = x$.

To safely carry out this process, the data handler must control the privacy loss incurred by the sequence of released outputs. Privacy preservation can be achieved through the use of a privacy filter. A *privacy filter* [11] is a sequential algorithm that, after every request, decides whether to halt the data release process or proceed with accepting a new request. In other words, it tracks leakage to inform a stopping rule. Let T denote the filter's stopping time which is defined as the index of the last released output when the algorithm halts, and is a random variable. In order for the filter to guarantee (ϵ, δ)-DP, we need to consider the following definition.

Definition 1 ((ϵ, δ)-DP Privacy Filter)**.** Given a sequence of mechanisms and allowable requests, a privacy filter is said to be (ϵ, δ)-DP if, for every adversary employing a random and adaptive strategy to choose requests, the following holds:

$$P_x\big((R^T, Y^T) \in \mathcal{S}\big) \leq e^{\epsilon} P_{x'}\left((R^T, Y^T) \in \mathcal{S}\right) + \delta, \quad (1)$$

for all neighbouring databases $x \sim x'$ and all measurable sets $\mathcal{S}$ of full transcripts.

We use the full transcript to define DP, even though requests are chosen by the adversary, since all random variables can be jointly used to infer about the database. Requiring privacy to hold for all measurable sets of full transcripts ensures that no adversary can distinguish neighbouring databases beyond the (ϵ, δ)-DP bound, however they select their requests.

A. Mechanism-Level Privacy Accounting

The classical additive composition result of [1] yields a simple privacy filter. Let (ϵ, δ) be the privacy budget. Suppose mechanisms $(\mathcal{M}_1, \ldots, \mathcal{M}_{k-1})$ have been applied, and a request r_k was made with $\mathcal{M}_k$ as the next mechanism. The so-called *additive privacy filter* [8] checks if $\sum_{i=1}^k \epsilon_i \leq \epsilon$ and $\sum_{i=1}^k \delta_i \leq \delta$, where (ϵ_i, δ_i) are the DP parameters of mechanism $\mathcal{M}_i(r^i, y^{i-1}, X)$. If either condition is violated, the data handler will not apply $\mathcal{M}_k$ to X and stops accepting new requests. Otherwise, she applies the mechanism, releases the corresponding data, and accepts a new request.

Using an advanced composition result for adaptive mechanisms, [8, Thm 5.1] gave an improved privacy filter (in certain regimes) that checks if $\delta' + \sum_{i=1}^k \frac{2\delta_i}{\epsilon_i e^{\epsilon_i}} \leq \delta$ and ϵ exceeds

$$\sum_{j=1}^{k} \frac{\epsilon_i \left(e^{\epsilon_i} - 1\right)}{2} + \sqrt{2\left(\sum_{i=1}^{k} \epsilon_i^2 + \frac{\epsilon^2}{28.04 \log(\frac{1}{\delta'})}\right)} \times \sqrt{\left(1 + \frac{1}{2}\log\left(\frac{28.04 \log(\frac{1}{\delta'}) \sum_{i=1}^k \epsilon_i^2}{\epsilon^2} + 1\right)\right) \log\left(\frac{2}{\delta'}\right)}.$$

We call this an *advanced privacy filter*. Privacy filters based on RDP parameters have also been proposed [9], [10]. As these filters use mechanism parameters to track privacy loss, we call them *mechanism-level* privacy filters. All guarantee (ϵ, δ)-DP.

B. Realisation-Level Privacy Accounting

Mechanism-level privacy accounting relies on worst case parameters for each mechanism, regardless of realised outputs, and may therefore halt when realised leakage is significantly less than the budget. Therefore, we propose another paradigm for privacy filter design based on *realisation-level* privacy accounting. The idea is to track the accumulated privacy loss of the full transcript (r^i, y^i) at each step. The leakage incurred in releasing y_i given the partial transcript (r^i, y^{i-1}) is[1]

$$l_i := \sup_{x \sim x'} \log \frac{P_x(y_i \mid r^i, y^{i-1})}{P_{x'}(y_i \mid r^i, y^{i-1})}.$$

The cumulative privacy loss at step i for the full transcript (r^i, y^i) is given by $l^{(i)} := \sum_{j=1}^i l_j$, with associated random variables L_i and $L^{(i)}$. We refer to these quantities as leakage. A filter tracking leakage this way may be less conservative by adapting to favourable outcomes, allowing more data releases without compromising privacy. This intuition is formalised by our proposed privacy filter in Sec. III and the rest of the paper.

Before presenting ours, we consider design aspects of a general realisation-level privacy filter. Upon stopping, the exact full-transcript leakage is given by the random variable

$$L_\star^{(T)} = \sup_{x \sim x'} \log \frac{P_x(R^T, Y^T)}{P_{x'}(R^T, Y^T)}, \quad (2)$$

where T is the stopping time of the algorithm, and (R^T, Y^T) is the full transcript the adversary has. By noting that R_i cannot depend on X given (R^{i-1}, Y^{i-1}), we can write

$$\begin{aligned} L_\star^{(T)} &= \sup_{x \sim x'} \log \frac{\prod_{i=1}^T P_x(R_i|R^{i-1}, Y^{i-1}) P_x(Y_i|R^i, Y^{i-1})}{\prod_{i=1}^T P_{x'}(R_i|R^{i-1}, Y^{i-1}) P_{x'}(Y_i|R^i, Y^{i-1})} \\ &= \sup_{x \sim x'} \log \frac{\prod_{i=1}^T P_x(Y_i|R^i, Y^{i-1})}{\prod_{i=1}^T P_{x'}(Y_i|R^i, Y^{i-1})} \\ &= \sup_{x \sim x'} \left[\log \frac{P_x(Y_1|R_1)}{P_{x'}(Y_1|R_1)} + \cdots + \log \frac{P_x(Y_T|R^T, Y^{T-1})}{P_{x'}(Y_T|R^T, Y^{T-1})}\right]. \end{aligned}$$

[1]A refined formulation is presented in the full version of the paper [12], which extends privacy filter comparisons beyond conceptually aligned methods to include RDP-based approaches, and yields improved utility.

Therefore, we have $L_\star^{(T)} \leq L^{(T)}$, where the latter is tracked by the filter. So, the filter bounds the exact leakage to an adversary at the stopping time by limiting the cumulative leakage $L^{(T)}$.

Central to the filter is the design of a stopping rule that preserves the desired privacy guarantee. In an attempt to achieve (ϵ, δ)-DP, one might construct the following naive realisation-level privacy filter. At each step i, upon receiving request r_i, it considers the accumulated leakage $l^{(i-1)}$ from the realised full transcript (r^{i-1}, y^{i-1}), and analyses the mechanism $\mathcal{M}_i(r^i, y^{i-1}, X)$. If the worst-case (over databases) probability that releasing Y_i would cause the accumulated privacy loss to exceed ϵ is greater than δ, i.e.,

$$\sup_x P_x\big(L_i > \epsilon - l^{(i-1)} \mid r^i, y^{i-1}\big) > \delta,$$

the filter halts and the mechanism is not executed; otherwise, the mechanism is applied and the output y_i is released. This construction ensures the stopping time leaks no information about X beyond that jointly revealed by requests and outputs.

Despite its intuitive nature and handling of stopping decisions, (ϵ, δ)-DP is not guaranteed. To see this, consider repeated application of an identical non-adaptive binary erasure mechanism, where an adversary makes the same fixed request at each step, and observes the output of the mechanism. Here, $Y = X$ with probability $p \leq \delta$ and $Y = \Delta$ otherwise. These events correspond to infinite and zero leakages respectively. The filter halts at step i iff $l^{(i-2)} = 0$ and $l^{(i-1)} = \infty$. In other words, the naive filter continues as long as only Δ's are observed. Therefore, with probability one, the infinite leakage event occurs, and by Def. 1, guarantee is no better than $(\epsilon, 1)$-DP. Owing to the difficulty in ensuring (ϵ, δ)-DP guarantee, a realisation-level privacy filter must be carefully designed.

III. A Realisation-Level Privacy Filter

We introduce a privacy filter that tracks cumulative leakage, and bounds it with a valid stopping rule. In step i, the request r_i is received. What follows is subtle but essential. Rather than deciding whether to release y_i, the algorithm decides whether it will receive the *following* request r_{i+1}, using knowledge of r^i and y^{i-1}. Specifically, it uses a step-wise parameter $\hat{\delta}_{i+1}$ to assess the risk of releasing y_{i+1} under the worst case request r_{i+1} and database x. Formally, $\hat{\delta}_{i+1}$ is given by

$$\inf\Big\{z \in [0,1] : \inf_{\substack{x \\ r_{i+1} \in \mathcal{R}_{i+1}}} P_x\big(Y_i \in \tilde{\mathcal{Y}}_i(z) \mid r^i, y^{i-1}\big) \geq 1-\theta\Big\}, \quad (3)$$

where $\tilde{\mathcal{Y}}_i(z)$ is a function of x and r_{i+1} and is defined as

$$\tilde{\mathcal{Y}}_i(z) = \Big\{y_i : P_x\big(L_{i+1} > \epsilon - l^{(i)} \mid r^i, r_{i+1}, y^{i-1}, y_i\big) \leq z\Big\}.$$

The filter requires checking the condition $\hat{\delta}_{i+1} \leq \tilde{\delta}$. This is most easily done by equivalently confirming whether

$$\inf_{\substack{x \\ r_{i+1} \in \mathcal{R}_{i+1}}} P_x\big(Y_i \in \tilde{\mathcal{Y}}_i(\tilde{\delta}) \mid r^i, y^{i-1}\big) \geq 1 - \theta. \quad (4)$$

Regardless of the outcome, the algorithm executes mechanism i and releases y_i. The privacy filter is outlined in Algorithm 1, with $r_0 = y_0 = \perp$, and $l_0 = 0$.

Algorithm 1: Realisation-level privacy filter

```
Input: x, ε, δ, M_1, M_2, ..., R_1, R_2, ...
Choose δ̃ ∈ [0, δ], θ ∈ [0, 1], and N ∈ ℤ⁺ such that
  δ̃ + θ(1 − δ̃)N ≤ δ
Initialize i = 0, l^(−1) = 0
while i ≤ N do
    Receive r_i
    Execute M_i(r^i, y^{i−1}, x) to obtain and release y_i.
    if δ̂_{i+1} > δ̃ (if (4) is false) then
        break
    end
    Compute l_i and update l^(i) ← l^(i−1) + l_i
    Update i ← i + 1
end
```

The look-ahead design of the stopping rule makes the filter (ϵ, δ)-DP, which is proved in Sec. IV. Formally, it ensures that the stopping event $\{T = i\}$ does not depend on the realised output y_i, which allows the stopping time to be decoupled from the leakage $L^{(i)}$ in the proof of Theorem 1.

Finally, we remark that the algorithm requires a choice of parameters $(\tilde{\delta}, \theta, N)$. Any decision satisfying the condition $\tilde{\delta} + \theta(1 - \tilde{\delta})N \leq \delta$ preserves (ϵ, δ)-DP, but may yield substantially different stopping times. Here, N is the maximum stopping time. The parameter θ introduces a relaxation in (4), allowing the condition to hold with high probability rather than with probability 1. If $\theta = 0$, the leakage induced by y_{i+1} is analysed for the worst possible realisation of y_i. Increasing θ softens this condition. In turn $\tilde{\delta}$ must be reduced from δ to accommodate positive θ. We also see a trade off between θ and N. The maximum stopping time is allowed to be very large if θ is very small. The best parameter choice may vary significantly according to the sequence of mechanisms input to the algorithm.

IV. Privacy Guarantee

In this section, we establish the privacy guarantee of our privacy filter. For brevity, we use $P_x(\mathcal{S})$ to mean $P_x((R^T, Y^T) \in \mathcal{S})$, and $\mathbb{E}_x$ to denote an expectation with respect to P_x.

Theorem 1. *The privacy filter described in Algorithm 1 is (ϵ, δ)-DP.*

Proof. Let $\mathcal{A} := \{L^{(T)} \leq \epsilon\}$, and let $\mathcal{A}_\star := \{L_\star^{(T)} \leq \epsilon\}$. To prove the (ϵ, δ)-DP guarantee, it is enough to show that

$$P_x(\mathcal{A}^c) \leq \delta \quad (5)$$

for all x, because of the following argument. The exact leakage $L_\star^{(T)}$ to the adversary is no greater than $L^{(T)}$, the leakage quantity tracked by the algorithm. Thus, $\mathcal{A} \subseteq \mathcal{A}_\star$. Additionally, if $(R^T, Y^T) \in \mathcal{A}_\star$ then $P_x(R^T, Y^T) \leq e^\epsilon P_{x'}(R^T, Y^T)$ for all neighbours $x \sim x'$. Hence, for any measurable set $\mathcal{S}$, we have $P_x(\mathcal{S} \cap \mathcal{A}) \leq e^\epsilon P_{x'}(\mathcal{S} \cap \mathcal{A})$. This combined with (5) yields

$$P_x(\mathcal{S}) = P_x(\mathcal{S} \cap \mathcal{A}) + P_x(\mathcal{S} \cap \mathcal{A}^c)$$

$$\leq e^{\epsilon} P_{x'}(\mathcal{S} \cap \mathcal{A}) + P_x(\mathcal{A}^c)$$
$$\leq e^{\epsilon} P_{x'}(\mathcal{S}) + P_x(\mathcal{A}^c) \leq e^{\epsilon} P_{x'}(\mathcal{S}) + \delta, \quad (6)$$

for all neighbours $x \sim x'$, proving the (ϵ, δ)-DP guarantee.

Now we show (5). To achieve this, define the event $\mathcal{A}_i := \left\{L^{(i)} \leq \epsilon\right\}$ and the random variable $F_i := \mathbf{1}_{\mathcal{A}_i}/(1-\hat{\delta}_i)$. for $i \geq 1$. Note that the stopping event $T = i$ can be written as

$$\{T = i\} = \left\{\hat{\delta}_1 \leq \tilde{\delta}, \ldots, \hat{\delta}_i \leq \tilde{\delta},\ \hat{\delta}_{i+1} > \tilde{\delta}\right\}.$$

Now consider the expectation

$$\mathbb{E}_x\left[F_i \mathbf{1}_{\{T=i\}}\right] = \mathbb{E}_x\left[\mathbb{E}_x\left[\frac{\mathbf{1}_{\mathcal{A}_i}}{1-\hat{\delta}_i}\mathbf{1}_{\{T=i\}}\middle| R^i, Y^{i-1}\right]\right]$$
$$= \mathbb{E}_x\left[\mathbf{1}_{\{T=i\}}\frac{1}{1-\hat{\delta}_i}P_x\left(\mathcal{A}_i \mid R^i, Y^{i-1}\right)\right] \quad (7)$$
$$= \mathbb{E}_x\left[\mathbb{E}_x\left[\mathbf{1}_{\{T=i\}}Z \mid R^i, Y^{i-2}\right]\right], \quad (8)$$

where (7) uses that fact that $\hat{\delta}_i$ is a function of R^{i-1}, Y^{i-2} and $\mathbf{1}_{\{T=i\}}$ is a function of R^i, Y^{i-1}, and for brevity, in (8) we use $Z := P_x(\mathcal{A}_i|R^i, Y^{i-2}, Y_{i-1})/(1-\hat{\delta}_i)$. Since $Z \geq \mathbf{1}_{\{Z\geq 1\}}$,

$$\mathbb{E}_x\left[\mathbf{1}_{\{T=i\}}Z|R^i, Y^{i-2}\right]$$
$$\geq P_x\left(\{T=i\} \cap \{Z \geq 1\} \mid R^i, Y^{i-2}\right)$$
$$\geq P_x\left(\{T=i\} \mid R^i, Y^{i-2}\right) + P_x\left(\{Z \geq 1\} \mid R^i, Y^{i-2}\right) - 1$$
$$\geq P_x\left(T=i \mid R^i, Y^{i-2}\right) - \theta, \quad (9)$$

where (9) follows from the definition of $\hat{\delta}_i$ (3) that given (R^{i-1}, Y^{i-2}), $Z \geq 1$ with probability at least $1-\theta$ for all R_i. By combining (8), and (9), we finally get

$$\mathbb{E}_x\left[F_i \mathbf{1}_{\{T=i\}}\right] \geq E_x\left[P_x\left(T=i \mid R^i, Y^{i-2}\right) - \theta\right]$$
$$= P_x(T=i) - \theta. \quad (10)$$

For an upper bound on the expression on the left-hand side of (10), note that if $T = i$, then $1-\hat{\delta}_i \geq 1-\tilde{\delta}$. Thus,

$$\mathbb{E}_x\left[F_i \mathbf{1}_{\{T=i\}}\right] = \mathbb{E}_x\left[\frac{\mathbf{1}_{\mathcal{A}_i \cap \{T=i\}}}{1-\hat{\delta}_i}\right]$$
$$\leq \frac{1}{1-\tilde{\delta}} P_x(\mathcal{A}_i \cap \{T=i\}). \quad (11)$$

Combining (10) and (11) yields

$$P_x(\mathcal{A}_i|T=i) \geq 1 - \tilde{\delta} - \frac{(1-\tilde{\delta})\theta}{P_x(T=i)}, \quad (12)$$

for all x. We now return to $P_x(\mathcal{A}^c)$. Noting that $P_x(\mathcal{A}_0^c) = 0$,

$$P_x(\mathcal{A}^c) = \sum_{i=0}^{N} P_x(\mathcal{A}_i^c|T=i)P_x(T=i)$$
$$\leq \sum_{i=1}^{N}\left(\tilde{\delta} + \frac{(1-\tilde{\delta})\theta}{P(T=i)}\right)P_x(T=i)$$
$$= \tilde{\delta} + \theta(1-\tilde{\delta})N \leq \delta,$$

where the first inequality follows from (12), and the second inequality is true by construction, proving the theorem.

□

V. Utility of Realisation-Level Accounting

Beyond ensuring differential privacy, a privacy filter is more useful if it can run for a longer, enabling many data releases without privacy compromise. Utility is thus naturally characterised by the stopping time T, the number of database queries before access is cut off. We focus on survival probabilities $P_x(T \geq k)$. In this section, we contrast the utility of conceptually aligned accounting methods: *classical* DP composition, *mechanism-level* DP filters, and *realisation-level* DP filters. All operate directly within the (ϵ, δ)-differential privacy framework, rather than using Rényi based accounting. In doing so, we aim to isolate the effect of realisation accounting, to determine its potential as an unexplored axis for improvement.

A. Pure Differential Privacy

We first consider $(\epsilon, 0)$-DP approaches, comparing their utility by formalising the release conditions. Assume that the filters operate on fixed sequences of pure DP mechanisms $\mathcal{M}_1, \mathcal{M}_2, ...$ and allowable sets $\mathcal{R}_1, \mathcal{R}_2, ...$ Among mechanism-level approaches, we adopt the additive privacy filter described in Sec. II since additive composition is tight for pure DP while advanced composition relies on $\delta > 0$. For step i, classical composition concerns the ϵ parameter of the full mechanism $\mathcal{M}_i(R^i, Y^{i-1}, X)$, whilst the additive filter tracks that of the realised mechanism $\mathcal{M}_i(r^i, y^{i-1}, X)$.

Let $\epsilon_{\mathcal{C}}^k, \epsilon_{\mathcal{M}}^k$, and $\epsilon_{\mathcal{R}}^k$ denote privacy loss under classical composition, mechanism-level accounting and realisation-level accounting respectively. If $\mathcal{M}_i(R^i, Y^{i-1}, X)$ is ϵ_i-DP, classical composition gives $\sum_{i=1}^k \epsilon_i$-DP. Writing

$$\epsilon_{\mathcal{C}}^k = \sum_{i=1}^{k} \epsilon_i = \sum_{i=1}^{k} \sup_{r^i, y^i,} l_i, \quad (13)$$

y_k is released if $\epsilon_{\mathcal{C}}^k \leq \epsilon$. The additive filter on the other hand considers the realised mechanism $\mathcal{M}_i(r^i, y^{i-1}, X)$, which is pure ϵ_i'-DP uniformly over y_i. With

$$\epsilon_{\mathcal{M}}^k = \sum_{i=1}^{k} \epsilon_i' = \sum_{i=1}^{k} \sup_{y_i} l_i, \quad (14)$$

y_k is released if $\epsilon_{\mathcal{M}}^k \leq \epsilon$. Finally, consider realisation-level accounting as in Algorithm 1. When $\delta = 0$, $\theta = \tilde{\delta} = 0$, and the continuation criteria for output k becomes $\epsilon_{\mathcal{R}}^k \leq \epsilon$, where

$$\epsilon_{\mathcal{R}}^k \leq \sum_{i=1}^{k-2} l_i + \sup_{r_k, y_{k-1}, y_k} (l_{k-1} + l_k). \quad (15)$$

The kth output is released if $\epsilon_{\mathcal{R}}^i \leq \epsilon$ for all $i \leq k$. Combining (13), (14) and (15) reveals that, for all k, $\epsilon_{\mathcal{C}}^k \geq \epsilon_{\mathcal{M}}^k$ and $\epsilon_{\mathcal{C}}^k \geq \epsilon_{\mathcal{R}}^k$. Thus, for any adversary, classical composition admits the fewest data releases, highlighting the benefit of privacy filtering. No general ordering exists between $\epsilon_{\mathcal{M}}^k$ and $\epsilon_{\mathcal{R}}^k$. A sufficient condition for $\epsilon_{\mathcal{M}}^k \geq \epsilon_{\mathcal{R}}^k$ is

$$\sum_{i=1}^{k-2}\left(\sup_{y_i} l_i - l_i\right) \geq \sup_{r_k, y_{k-1}, y_k} l_k - \sup_{y_k} l_k.$$

The inequality compares a single contribution at step k with an accumulation of earlier effects. While the right-hand side may dominate for small k, the left-hand side grows with k. Once the filters pass early stopping thresholds, the condition increasingly favours the realisation-level filter. A similar effect appears numerically in Sec. V-B in the approximate DP setting.

B. Approximate Differential Privacy

In this section, we numerically compare the utility of the conceptually aligned *mechanism-level* DP filters of [8] with our *realisation-level* DP filter, described by Algorithm 1.

We simulate both the additive and advanced privacy filters outlined in Sec. II. A given mechanism $\mathcal{M}_i(r_i, y_{i-1}, X)$ may admit multiple valid (ϵ_i, δ_i) pairs, meaning a specific choice must be made at each round. To allow fair comparison, we choose natural parameters that avoid overly conservative accounting. We simulate two sensible approaches for the additive filter. The first, which we call *uniform* δ_i, sets $\delta_i = \delta/N$ and chooses the corresponding minimum ϵ_i. This choice ensures that the additive filter shares the same maximum stopping time N as our filter, allowing survival probabilities to be compared on a level footing for $T \leq N$. The second, which we call *ratio* δ_i, jointly selects (ϵ_i, δ_i) by fixing the ratio δ_i/ϵ_i at each round to match the ratio of the remaining δ and ϵ budgets. This ensures that ϵ and δ are consumed proportionally and exhausted simultaneously. Also, it returns a (ϵ_i, δ_i) whenever any feasible pair exists, and will not stop unless the remaining budget is insufficient. For the advanced filter, we follow similar intuition. Each round, we split the δ budget equally between δ' and $\sum_{i=1}^{k} \frac{2\delta_i}{\epsilon_i e^{\epsilon_i}}$. More precisely, we fix $\delta' = \frac{\delta}{2}$, and choose (ϵ_i, δ_i) such that $\sum_{i=1}^{k} \frac{2\delta_i}{\epsilon_i e^{\epsilon_i}} = \frac{\delta}{2}$. By contrast, our filter tracks realised leakage and avoids this choice.

We simulate adaptive Gaussian mechanisms with $Y_i = X + Z_i$, where $Z_i \sim \mathcal{N}(0, \sigma_i^2)$, with adaptive variance:

$$\sigma_i = \min\left\{ \frac{\sigma_{\text{base}}}{1 + \frac{1}{i-1}\sum_{j=1}^{i-1} |y_j|}, \sigma_{\min} \right\}.$$

We consider an adversary attempting to distinguish $X = 0$ and $X = 1$, with true value $X = 0$. We set $\sigma_{\text{base}} = 10$ and $\sigma_{\min} = 1$. For the realisation-level filter, we use $N = 20$, $\theta = 0.0035$, and the corresponding minimum $\tilde{\delta}$. Curves and stopping checks (4) are computed via Monte Carlo simulation.

Results are presented in Figure 2. The realisation-level filter generally dominates the mechanism-level filters, particularly for larger k, with a mean stopping time of 9.18 compared to 6.88 for the best mechanism-level filter. The realisation-level filter admits a lower survival probability than the additive filters for a small number of early k values. This is consistent with its reliance on realised leakage values, which may be unusually high, occasionally triggering early stopping. The effect is outweighed in the long term. We also remark the poor performance of the advanced filter, which is down to the particular mechanisms simulated. The advanced composition theorem [2], [3] significantly outperforms additive composition in the small ϵ_i regime, which is not where this simulation operates. Overall, the results highlight the potential for utility gains from privacy accounting at the realisation level.

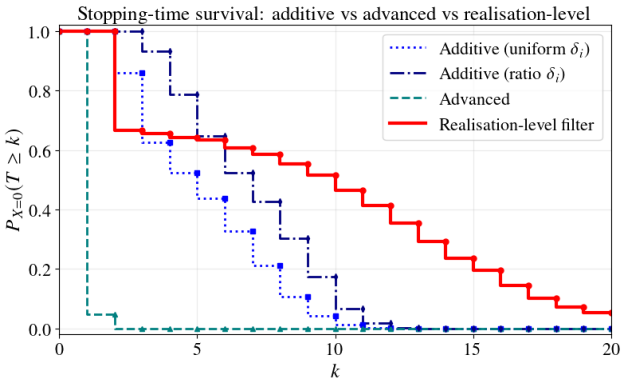

Fig. 2. Stopping time survival $P_{X=0}(T \geq k)$ of mechanism-level DP privacy filters compared with our realisation-level DP privacy filter.

VI. Discussion and Future Direction

In this work, we propose privacy filtering at realisation level, design one such privacy filter, and prove that it guarantees (ϵ, δ)-differential privacy. Contrasting prior approaches, the filter tracks realised leakage, rather than composing worst case per mechanism parameters. It can be applied to arbitrary mechanisms, whether they be independent or adaptive, with discrete or continuous outputs that may have differing supports. This extends beyond settings where Rényi divergence is well behaved, as required by RDP filters. This work demonstrates that realisation-level accounting provides a previously unexplored axis for utility improvement, by exploiting knowledge of the realised leakage. The main implementation challenge of the filter is in computing $\hat{\delta}_{i+1}$. Whilst analytical computation is feasible in simple settings, efficient approximations or bounds may be required for more elaborate mechanisms. Existing filters face similar scaling issues to ours as bounds must be controlled over all neighbouring datasets, which is expensive for complex mechanisms. Our filter additionally requires bounds to hold for all requests in $\mathcal{R}_{i+1}$. Unlike datasets, this set may be controlled by the data handler, who can reject other requests and safely continue. The effect of this strategy on utility warrants further study, particularly in adaptive learning settings where requests may be difficult to predict. Also, optimal selection of filter parameters $(\tilde{\delta}, \theta, N)$ remains open.

Exploring filters that combine realisation-level accounting with Rényi-based composition is a promising direction for future work. Rényi differential privacy is known to provide very strong composition guarantees for light-tailed mechanisms, such as those based on Gaussian noise, and existing RDP filters leverage this property to achieve high utility for some common mechanisms. The simulation results in this work illustrate that, among conceptually aligned privacy filters, realisation-level accounting can yield improved stopping time behaviour by exploiting knowledge of realised privacy loss. In summary, our work provides evidence that realisation-level accounting provides a complementary path to higher utility.

REFERENCES

[1] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

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

[3] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, ser. Foundations and Trends in Theoretical Computer Science. Now Publishers Inc., 2014, vol. 9, no. 3–4.

[4] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 263–275.

[5] A. Koskela, J. Jälkö, and A. Honkela, "Computing tight differential privacy guarantees using fft," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2560–2569.

[6] A. Koskela, J. Jälkö, L. Prediger, and A. Honkela, "Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3358–3366.

[7] W. Alghamdi, J. F. Gomez, S. Asoodeh, F. Calmon, O. Kosut, and L. Sankar, "The saddle-point method in differential privacy," in *International Conference on Machine Learning*. PMLR, 2023, pp. 508–528.

[8] R. Rogers, A. Roth, J. Ullman, and S. Vadhan, "Privacy odometers and filters: Pay-as-you-go composition," 2021. [Online]. Available: https://arxiv.org/abs/1605.08294

[9] V. Feldman and T. Zrnic, "Individual privacy accounting via a renyi filter," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 080–28 091, 2021.

[10] M. Lécuyer, "Practical privacy filters and odometers with rényi differential privacy and applications to differentially private deep learning," *arXiv preprint arXiv:2103.01379v2*, 2021.

[11] R. Rogers, A. Roth, J. Ullman, and S. Vadhan, "Privacy odometers and filters: pay-as-you-go composition," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 1929–1937.

[12] S. Taylor, P. Vippathalla, and J. Coon, "Realisation-level privacy filtering," 2026. [Online]. Available: https://arxiv.org/abs/2604.08630