

# Big Bird: Resilient Privacy Budgeting Across Untrusted Web Domains

Pierre Tholoniati\*  
Columbia University

Alison Caulfield\*  
Columbia University

Giorgio Cavicchioli  
Columbia University

Mark Chen  
Columbia University

Benjamin Case  
Meta Platforms, Inc.

Asaf Cidon  
Columbia University

Roxana Geambasu\*  
Columbia University

Mathias Lécuyer  
University of British Columbia

Martin Thomson  
Mozilla

## Abstract

The W3C Attribution API is an emerging standard for privacy-preserving advertising measurement. Its current privacy architecture enforces individual differential privacy (IDP) independently for each domain (e.g., an advertiser) issuing queries. We show that this guarantee is unsound under realistic system behavior: it fails under cross-querier data adaptivity and can also fail when shared limits are enforced across queriers. The issue is not the on-device accounting model itself – device-epoch IDP – but treating each querying domain in isolation. We propose Big Bird, a privacy-budget manager that makes global device-epoch IDP – enforced jointly across all domains – both sound and deployable for Attribution. Big Bird addresses the main obstacle to global enforcement in open multi-querier systems: denial-of-service depletion of a shared global budget by Sybil web domains. Its key insight is that benign Attribution workloads have a stock-and-flow structure: impressions create potential privacy loss, conversions realize it, and meaningful budget consumption should be tied to genuine user actions across distinct web domains. Big Bird enforces this structure with privacy-loss-based quotas on impression and conversion sites and a per-user-action cap on how many quotas can be activated, ensuring that adversarial impact scales with genuine user interactions rather than with the number of Sybil domains. We implement Big Bird in Rust, integrate it into Firefox’s Attribution prototype, and evaluate it theoretically and empirically on real ad-tech data. We show that Big Bird provides rigorous global device-epoch IDP, formal resilience to depletion attacks, and utility for benign queriers under attack.

\*Contributions made at Columbia University. P.T. is now at Google. A.C. and R.G. were also affiliated with Microsoft and Meta, respectively.



This work is licensed under a Creative Commons Attribution 4.0 International License.

SOSP '26, Prague, Czech Republic

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2585-2/26/09

<https://doi.org/10.1145/3830418.3843897>

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

**Keywords:** Differential Privacy, Web, Measurement

## ACM Reference Format:

Pierre Tholoniati, Alison Caulfield, Giorgio Cavicchioli, Mark Chen, Benjamin Case, Asaf Cidon, Roxana Geambasu, Mathias Lécuyer, and Martin Thomson. 2026. Big Bird: Resilient Privacy Budgeting Across Untrusted Web Domains. In *Symposium on Operating Systems Principles (SOSP '26)*, September 29–October 2, 2026, Prague, Czech Republic. ACM, New York, NY, USA, 35 pages. <https://doi.org/10.1145/3830418.3843897>

## 1 Introduction

An important new standard is taking shape at the W3C: the Attribution API, designed to enable privacy-preserving advertising measurement at web scale [33, 43]. Advertising measurement is a core economic function of the web, yet today it relies heavily on cross-site tracking mechanisms such as third-party cookies and fingerprinting, which undermine user privacy and have proven difficult to regulate or replace. Attribution aims to provide a principled alternative: a browser-mediated interface for cross-site measurement that replaces raw data collection with structured, differentially private aggregation.

At a high level, Attribution enables tasks such as conversion attribution measurement – linking ads viewed on content sites to purchases on seller sites – while ensuring that only noisy aggregate statistics are released. Browsers mediate all raw-data flows, store user activity locally, and release encrypted reports that are aggregated with differential privacy (DP), eliminating the need for cross-site tracking or centralized user-level data collection.

A key feature of Attribution is device-side privacy accounting based on individual differential privacy (IDP), which we developed for Attribution in prior work, Cookie Monster [43]. Each browser enforces a per-epoch privacy budget for each querier (e.g., an advertiser), deducting privacy loss before releasing reports. This design offers strong utility by charging privacy loss only when a user meaningfully contributes to a query, and it has been adopted as the foundation of the emerging standard – including its per-querier enforcement scope. The standard is being developed with participation

from all major browsers; as of August 2026 it is in Wide-Area Review, en route to Candidate Recommendation.

Given its role as a *web standard* – not merely an implementation, and likely to be deployed across billions of devices – scrutinizing Attribution’s privacy guarantees is essential and urgent. We therefore ask: are its guarantees sound, and can they be enforced robustly in practice?

**Contribution 1:** We show that Attribution’s central privacy guarantee – device-epoch IDP enforced per querier – is fundamentally unsound under realistic system behavior. The issue is not IDP or the device-epoch unit, but per-querier enforcement. In multi-querier systems, privacy guarantees must account for interactions across queriers. We identify two failure modes: (1) cross-querier data adaptivity in streaming settings causes privacy loss to exceed per-querier bounds, even without collusion; and (2) introducing shared limits across queriers, a common defense against collusion, also violates per-querier guarantees. We prove these results in a general DP system model, showing that the problem is not specific to Attribution but reflects a fundamental mismatch between per-querier enforcement and deployment realities.

Focusing on Attribution, we therefore advocate global device-epoch IDP – enforced across all queriers – as the sound privacy foundation. However, global enforcement introduces a key systems challenge. In open environments where any domain can invoke the API, a shared global privacy budget becomes vulnerable to denial-of-service depletion: adversarial domains can exhaust users’ budgets, preventing benign measurement.

**Contribution 2:** We present Big Bird, a global privacy-budget manager for Attribution that provides both rigorous global IDP enforcement and resilience to depletion attacks by untrusted web domains. Big Bird is based on a key structural insight: legitimate Attribution workloads exhibit a stock-and-flow pattern of privacy consumption, where impressions create potential privacy loss (stock) and conversions realize it (flow), tied to genuine user actions across sites. Attacks violate this structure by generating stock and flow without corresponding user activity.

Big Bird restores this structure using coordinated, IDP-aligned quotas on impression sites, conversion sites, and per-user-action domain participation. These constraints ensure that privacy-budget consumption – whether benign or adversarial – scales with genuine user interactions rather than with the number of attacker-controlled (Sybil) domains.

We implement Big Bird in Rust as a generic IDP budget management library, *pdslib*, integrate it into Firefox’s Attribution prototype, analyze it theoretically, and evaluate it empirically on real ad-tech data. Our results show that Big Bird enforces rigorous global device-epoch IDP, achieves formal resilience to depletion attacks, and preserves utility for benign workloads under attack.

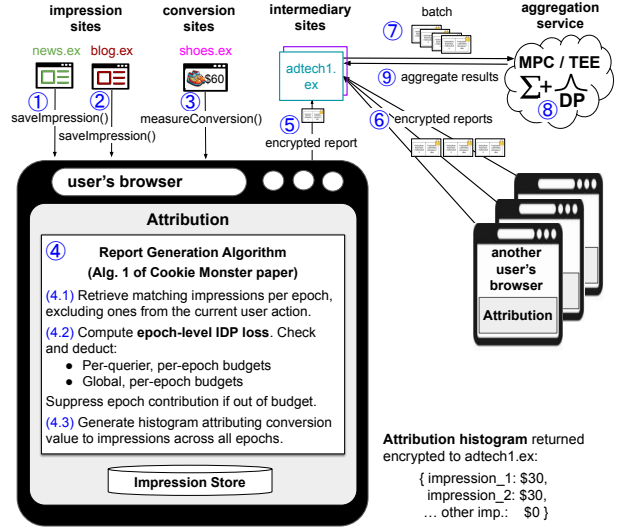


Fig. 1. Attribution API architecture.

Both contributions are now part of the Attribution draft.<sup>1</sup> Following the recommendation in §3.4, the specification added a section recording our results as limitations of per-querier semantics and discussing their implications [35]. The draft also adopted the Big Bird algorithm from §4 (Alg. 2) as its budget-management mechanism [34]. The latter resolved an open issue [19]—DoS depletion of the global safety limits—that the editors had flagged as a blocker to the standard’s reaching the Wide-Area Review stage, which it has since reached. We also release our source code as a reference implementation for the standard as well as for broader use cases: *pdslib* at <https://github.com/columbia/pdslib>, Firefox integration at <https://github.com/columbia/pdslib-firefox>, and experimental infrastructure at <https://github.com/columbia/bigbird>. Indeed, while we focus on Attribution, our results have broader implications for DP systems in untrusted querier settings: (1) multi-granularity enforcement is fundamentally unsound, and (2) global budget management requires depletion defenses aligned with the structure of benign workloads. We view Big Bird as one step toward this broader design space.

## 2 Background

### 2.1 Attribution API Architecture

W3C Attribution is a privacy-preserving, browser-based interface for cross-site conversion measurement. Figure 1 shows the workflow, involving four principals, where *.ex* denotes fictional example domains: (1) *impression sites* (e.g., *news.ex*, *blog.ex*) register shown ads via `saveImpression()`; (2) *conversion sites* (e.g., *shoes.ex*) call `measureConversion()` to request encrypted reports of which ads a user saw; (3) *intermediaries* (e.g., *adtech.ex*) receive encrypted reports; and (4) *aggregation services* (e.g., *divviup.org*) aggregate reports

<sup>1</sup>B.C. and M.T. are specification editors [33]. All changes described here were proposed, discussed, and adopted through the W3C public process.

and add DP noise using trusted MPC or TEE.

The browser mediates all information flows: impressions are stored locally, conversions trigger encrypted reports, and aggregators release only DP-protected statistics. A typical sequence is: ①–② impression sites record ads; ③ a conversion site calls `measureConversion()`; ④ the browser fetches relevant impressions, checks local privacy budgets, and ⑤ generates encrypted reports; ⑥–⑨ reports are batched, aggregated, and released with DP noise via an aggregation service.

Critically, *privacy enforcement happens on-device*. Before releasing a report, the browser deducts the corresponding privacy loss from a local *privacy budget store*, returning null reports when budget is exhausted (4.2). This enforcement uses an algorithm presented in the Cookie Monster paper [43].

## 2.2 Cookie Monster Accounting

Our prior work, Cookie Monster [43], proposed using *individual differential privacy (IDP)* [13, 14, 22] for efficient, on-device privacy accounting in Attribution, and the accounting it introduced is the one the standard adopted. We summarize it here because both of this paper’s contributions build on it: §3 revisits the scope of its guarantee, and §4 builds a budget manager on top of its accounting.

Under IDP, each browser maintains its own privacy budgets and deducts privacy loss based on the user’s actual data contribution (the *individual sensitivity*), rather than a population-wide worst-case bound (the *sensitivity*) as in traditional DP. This distinction enables a crucial optimization for Attribution. Advertising signals are sparse: many users see ads but never convert; many conversions have no matching impression; and users purchase different products and therefore participate in different queries. Cookie Monster’s accounting exploits this sparsity. Each device partitions its data into non-overlapping epochs (e.g., one week), and when a conversion occurs, the browser charges privacy loss *only for epochs containing at least one impression relevant to that conversion*. If an epoch contains no such impression, the conversion incurs *zero* privacy loss for that device-epoch. When a charge is applied, it is bounded by  $(\text{value} / \text{maxValue}) \times \epsilon$ , using parameters from the `measureConversion()` call.

This contrasts with traditional DP, where privacy loss is charged for every query based on population-level sensitivity, even if a device contributed no relevant data. For Attribution, avoiding charges in irrelevant epochs is essential for utility: it reduces privacy loss for typical users and allows many measurements before budgets are exhausted.

## 2.3 Queriers and Privacy Guarantees

Attribution performs privacy accounting *with respect to a querier* – the entity on whose behalf aggregation occurs. In today’s API, the querier is the *advertiser* measuring its own conversions. A planned extension will add *intermediaries* as queriers for cross-advertiser queries (e.g., for training

ad-placement models). Each browser therefore maintains *per-epoch budgets per querier*. We showed in Cookie Monster that, *when queriers are modeled in isolation*, such per-querier budgets satisfy device-epoch-level IDP [43].

The Attribution specification acknowledges a well-known limitation: if queriers collude, their privacy losses add. To mitigate this, the API introduces *global safety limits* – per-epoch budgets spanning all queriers. In the current design, per-querier budgets remain the primary enforcement mechanism, with global limits viewed as defense-in-depth.

## 3 Contribution 1: Per-Querier DP Analysis

Attribution’s focus on per-querier enforcement, with global limits as defense-in-depth, is not unique. Several DP systems have similarly proposed per-querier or per-analyst enforcement [15, 16, 24, 43, 44] to align guarantees with organizational boundaries. Such guarantees are intuitively appealing, especially when queriers can reasonably be assumed to act independently and not collude (e.g., competing organizations or scientists under IRB restrictions). Some of these systems also, like Attribution, enforce DP at multiple granularities for defense-in-depth [15, 24, 44].

Our analysis of per-querier guarantees – developed initially for Attribution but generalized here to broader DP systems – shows that their limitations extend beyond collusion. Any DP analysis has three components: (1) a formal privacy property (definition, privacy unit, enforcement scope), (2) a model of system behavior specifying how data is generated and queried, and (3) a proof that the modeled system satisfies the property. If the behavioral model omits deployment-relevant behaviors that affect data generation or querying, the proof may still be correct *relative to the model*, while the stated privacy property fails to capture the system’s actual semantics.

We identify two recurring modeling gaps in analyses of per-querier DP. First, existing models typically treat each querier in isolation, neglecting cross-querier influences on data generation that naturally arise in streaming systems. Second, when per-querier budgets and shared limits coexist, prior work has generally modeled them separately, overlooking how their joint enforcement shapes system behavior.

When these omitted behaviors are modeled, per-querier guarantees become unsound: even without collusion, a single querier’s view can reveal more about the dataset than the per-querier budget permits. This does not render per-querier budgets useless – they remain valuable engineering constraints – but it does mean their formal guarantees rest on stronger assumptions than previously recognized. More broadly, these results reinforce a key takeaway: rigorous privacy analysis relies as much on *faithful system modeling* as on the mathematics of the DP proof itself.

### 3.1 Multi-Querier System Model

We define a generic model for a multi-querier DP system, not specific to Attribution. Alg. 1 shows a system with  $K$  queriers ( $Q^1, \dots, Q^K$ ), each with its own privacy budget  $\mathcal{F}^k$ , interacting with a shared database  $D$  over discrete time steps  $t$ . The model accommodates both streams and static databases: a data generation process  $\mathcal{G}$  may append new data to  $D$  at each step, while in the static case  $\mathcal{G}$  outputs nothing after the initial step. It supports both per-querier enforcement, via  $\mathcal{F}^k$ , and an abstract shared limit  $\mathcal{L}$ , which can represent a global budget. At each step, querier  $k$  selects a query  $Q_t^k$  based only on its own past results, thereby explicitly *excluding intentional collusion*. The query executes only if both  $\mathcal{F}^k$  and  $\mathcal{L}$  approve, in which case the querier receives result  $V_t^k$ ; otherwise it receives a data-independent result  $\perp$ .

To analyze DP, we introduce a *challenge bit*  $\chi$  that governs whether a particular record  $x_0$  is added to the database at time  $t_0$ . Neither the queriers nor  $\mathcal{G}$  see  $\chi$ , but as queriers issue queries, their views may reveal information about it, modeling the leakage of an individual's presence through DP query results. Privacy loss quantifies how hard it is for an adversary to distinguish between  $\chi = 0$  and  $\chi = 1$ , whether the adversary controls a single querier (per-querier guarantees) or all queriers (global guarantees).

Formally, Alg. 1 satisfies  $\epsilon$ -DP for querier  $k \in [K]$  if for  $\forall x_0 \in \mathcal{X}, \forall t_0 \in [t_{\max}], \forall \mathcal{G}, \forall (Q^1, \dots, Q^{n+1}), \forall S \subseteq \mathcal{R}^{t_{\max}}$ :

$$\left| \ln \left( \frac{\Pr[V^{k,(\chi=0)} \in S]}{\Pr[V^{k,(\chi=1)} \in S]} \right) \right| \leq \epsilon.$$

By contrast, a global guarantee would bound  $\frac{\Pr[V^{(\chi=0)} \in S]}{\Pr[V^{(\chi=1)} \in S]}$ , if for example  $\mathcal{L}$  were implemented as a global budget.

We next analyze per-querier guarantees and show two orthogonal cases in which they degrade linearly with the number of queriers. These correspond to conditions (A) and (B) in Alg. 1: adaptive data generation in streaming settings, and the presence of a shared limit in addition to per-querier budgets. By toggling these conditions on and off, we isolate the effect of each on per-querier guarantees. Appendix A.3 extends these results to IDP budgets, as used in Attribution.

### 3.2 Failure Mode 1: Adaptive Data Generation

We first consider the case where data is generated over time with no shared limits ( $\mathcal{L}$  always outputs TRUE). Adaptivity in data generation – where new data depends on past DP query results – has been studied in both streaming [11, 18, 21] and interactive [16, 26, 28, 43] systems. Modeling such adaptivity is essential because it both directly affects DP analyses, as prior work shows, and is inevitable in streaming systems. For instance, a movie recommender may use prior DP measurements to suggest a film, and the user's decision to watch it then becomes new data for future measurement.

Line 9 of Alg. 1 captures data adaptivity for multi-querier systems:  $D_t \leftarrow \mathcal{G}(V_{<t}^1, \dots, V_{<t}^K)$ , meaning data can depend not only on one querier's past results but also on others'. This cross-querier dependency arises even without collu-

### Alg. 1 Formal model of a generic multi-querier, pure-DP system

```

1: Input
2: Challenge bit  $\chi \in \{0, 1\}$ 
3: In-or-out record  $x_0$  to be inserted at time  $t_0$ 
4: Data generation process  $\mathcal{G}$ 
5: Queriers  $Q^1, \dots, Q^K$ 
6: Per-querier budgets (modeled as pure-DP filters)  $\mathcal{F}^1, \dots, \mathcal{F}^K$ 
7: Shared limit  $\mathcal{L}$ 
8: for  $t \in [t_{\max}]$  do
  // Data generation model:
  // (A) Cross-querier data adaptivity
9:  $D_t \leftarrow \mathcal{G}(V_{<t}^1, \dots, V_{<t}^K)$  // new data depends on all past results
10: if  $t = t_0$  and  $\chi = 1$  then // insert in-or-out record on challenge
11:    $D_t \leftarrow D_t + x_0$ 
  // Query execution model:
12: for  $k \in [K]$  do
13:    $q_t^k \leftarrow Q^k(V_{<t}^k)$  //  $Q^k$  selects query based on its own past results
  // (B) Shared-limit enforcement
14:   if  $\mathcal{F}^k(q_1^k, \dots, q_t^k) \wedge \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^K)$  then
15:      $V_t^k \leftarrow q_t^k(D_{\leq t})$ 
16:   else
17:      $V_t^k \leftarrow \perp$ 
18: return  $V^1, \dots, V^K$ 

```

sion, through user behavior or mechanisms like auctions. For example, `news.ex` may show an impression for either `shoes.ex` or `hats.ex`, depending on bids shaped by prior DP results from both sites.

We prove that under this model, per-querier budgets (formalized as privacy filters [38]) with capacity  $\epsilon$  do not suffice:

**Theorem 3.1** (Per-querier guarantees are unsound under adaptive data generation). *Consider Alg. 1, with  $K = n+1$  queriers, where each  $\mathcal{F}^k$  is a pure-DP filter with capacity  $\epsilon > 0$ , and where  $\mathcal{L}$  always outputs TRUE (i.e., no shared limit). Denote by  $V^{n+1,(\chi)}$  the view of  $Q^{n+1}$  on challenge bit  $\chi$ . Then, there exists  $\epsilon' > 0$  such that  $\exists x_0 \in \mathcal{X}, \exists t_0 \in [t_{\max}], \exists \mathcal{G}, \exists Q^1, \dots, Q^{n+1}, \exists S \subseteq \mathcal{R}^{t_{\max}}$ :*

$$\left| \ln \left( \frac{\Pr[V^{n+1,(\chi=0)} \in S]}{\Pr[V^{n+1,(\chi=1)} \in S]} \right) \right| \geq \epsilon + n\epsilon'.$$

Thm. 3.1 shows that with per-querier budgets of capacity  $\epsilon$ , privacy loss can exceed  $\epsilon$ , with *excess loss growing linearly in the number of queriers*. This arises not from collusion but from worst-case analysis that DP must account for. The proof, given in Appendix A.1, works by constructing an example of data generation and querying processes that exhibits that much excess loss. While the theorem addresses only pure-DP guarantees (yielding a linear increase in excess loss), the result likely extends to approximate DP (with excess loss growing with the square root of the number of queriers).

Prior analyses overlooked this point. ARA's implementation supports multiple adtechs [17], but its proof models only a single querier [16], effectively defining data generation as  $\mathcal{G}(V_{<t}^k)$  for a fixed  $k$  rather than our more general  $\mathcal{G}(V_{<t}^1, \dots, V_{<t}^K)$ . Cookie Monster, by contrast, allowed multiple queriers but modeled a static dataset in its proof [43].

### 3.3 Failure Mode 2: Shared Limits

We next consider the case where a shared limit is added to a system with per-querier enforcement (line 14 in Alg. 1).

Such limits have been proposed in both static [15, 44] and streaming [24] DP systems as defense-in-depth against collusion. We instantiate  $\mathcal{L}$  as a pure-DP filter shared across all queriers, which lets us quantify the excess privacy loss. The same reasoning, however, applies to other shared-limit mechanisms, such as caps on the total number of queries or rate limiters or quotas spanning subsets of queriers.

We show that even without data adaptivity, per-querier privacy loss can exceed  $\epsilon$  in the presence of a shared budget. To isolate this effect from Thm. 3.1, we restrict attention to static databases ( $\mathcal{G}$  returns  $\emptyset$  after the first time step). In this setting, the shared budget acts as a cross-querier leakage channel, enabling excess loss that, as before, grows linearly with the number of queriers covered by the shared limit:

**Theorem 3.2** (Per-querier guarantees are unsound under adaptive query budgets with shared limits). *Take  $n + 1$  queriers, with per-querier filters with capacity  $\epsilon$ , and a shared filter with capacity  $\epsilon_g < (n + 1)\epsilon$ . There exists  $\epsilon' > 0$  such that for all  $n \geq \frac{4 \ln 2}{(1 - e^{-\epsilon/2})^2}$ , there exists  $\mathcal{G}$  that returns  $\emptyset$  after the first time step, such that  $\exists x_0 \in \mathcal{X}, \exists t_0 \in [t_{\max}], \exists Q^1, \dots, Q^{n+1}, \exists S \subseteq \mathcal{R}^{t_{\max}} :$*

$$\left| \ln \left( \frac{\Pr[V^{n+1, (\chi=0)} \in S]}{\Pr[V^{n+1, (\chi=1)} \in S]} \right) \right| \geq \epsilon + n\epsilon'.$$

The proof constructs an attack that induces this excess loss. While it may appear to rely on collusion, the same leakage can arise inadvertently. At a high level, multiple queriers spend part of their budgets on an initial measurement and issue a follow-up query only when the result is high (e.g., measuring ad exposure and conditionally computing a histogram). The shared filter then encodes this signal: it is more likely to be exhausted when the true value is high, since more queriers issue follow-up queries. A new querier can thus infer this information from whether its own query is accepted or rejected, exceeding the privacy bound of its local budget. For IDP, rejection is not directly observable, requiring a more subtle argument (see §A.2 for DP and IDP proofs).

This effect has been missed in prior work on multi-granularity enforcement, either because it was not formalized at all [15] or because per-querier and global limits were analyzed in isolation or under non-adaptive queriers [24, 44]. Conceptually, this is like treating the two conditions in  $\mathcal{F}^k(q_1^k, \dots, q_t^k) \wedge \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^k)$  as if they acted independently, when in real systems they are necessarily enforced jointly.

### 3.4 Siloing Assumption and Takeaways

It is natural to ask whether any conditions – realistic or not – could make per-querier enforcement sound. Appendix A.4 presents one such condition (Thm. A.3), the *siloing assumption*: each querier operates on a completely disjoint data stream, and shared limits are disabled or so loose that they never block queries. Under these constraints, per-querier DP is sound in our general model (Alg. 1).

These constraints are almost certainly unattainable in practice. Worse, attempting to engineer them – for example by weakening or removing shared limits – undermines the

goal of strong user privacy. We therefore strongly recommend *against* doing so.

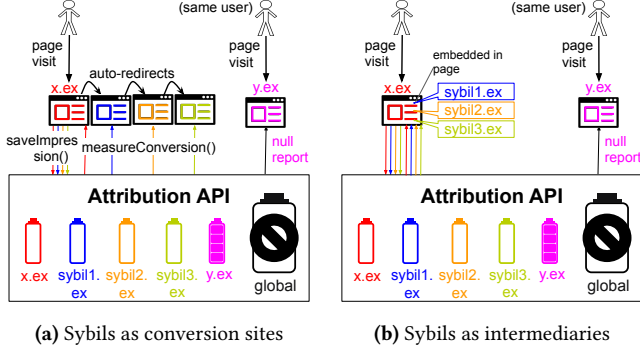
**Instead, the takeaways are three-fold.** First, Attribution should explicitly acknowledge the semantic limitations of per-querier enforcement. Second, per-querier budgets should be complemented with global enforcement; our second contribution – and the remainder of this paper – shows how to do so robustly and efficiently. Attribution has since adopted both: the specification now records the semantic limitations of per-querier enforcement [35], and specifies global enforcement managed by the algorithm of §4 [34]. Third, a broader lesson for the research community: rigorous privacy analysis depends as much on *faithful system modeling* as on DP mathematics. While this is well-understood in formal methods, the DP literature often relies on oversimplified models to keep proofs simple. Our results highlight why modeling fidelity must become a first-class concern in DP systems research.

## 4 Contribution 2: Global IDP with Big Bird

Global enforcement escapes the failures of §3 because both are failures of scope, not of DP accounting. Thm. 3.1 and Thm. 3.2 exhibit leakage that a per-querier budget cannot see: information routed through the data stream or through the shared limit, and thus outside the budget meant to bound it. A global budget has no analogous outside channel—it accounts for the joint information released across all queriers, so cross-querier channels are inside the quantity being bounded, not a way around it. This section therefore claims a global guarantee only, and shows that sound global device-epoch IDP is achievable under a system model we make as comprehensive as we can (Thm. 4.2). Per-querier budgets remain in the design as engineering constraints protecting utility, not privacy; the caveats of §3 apply to them here as anywhere.

Global enforcement does, however, introduce a new systems challenge: in open, multi-querier settings with untrusted participants, a shared global budget becomes vulnerable to *depletion attacks*—a denial-of-service (DoS) on utility rather than privacy. In Attribution, *any* web domain can invoke the API, allowing an attacker controlling many Sybil domains to act as distinct queriers, exhaust users’ budgets, and block benign measurement. Although Attribution introduces global safety limits for privacy, it leaves their management unspecified and identifies DoS depletion as a key roadblock to deployment [19].

We present *Big Bird*, a global budget manager for Attribution that makes device-epoch IDP both sound and deployable. Its key idea is to align budget consumption with the structure of legitimate workloads: impressions create potential privacy loss, conversions realize it, and meaningful consumption should scale with genuine user actions across distinct sites. Big Bird enforces this via IDP-aligned quotas on impression and conversion sites and a per-user-action cap on



**Fig. 2. Example attacks.** Sybils deplete the global budget from a single user action by acting as (a) conversion sites or (b) intermediaries. Per-querier and global budgets are shown as batteries (filled = available, empty = drained). domain participation, while preserving device-epoch IDP.

#### 4.1 Threat Model

Global budgets are a *shared resource*, and on the web, attacker identities are weak (Attribution uses eTLD+1 domain names) and browser API use is generally open. An attacker can exploit this by registering many *Sybil domain names* and coordinating them to deplete a device’s global budget. If this occurs on many devices, benign queriers are locked out of measurement, yielding a DoS based on the privacy layer.

**Example attacks.** Fig. 2 shows two example DoS depletion attacks effective against current Attribution. In Fig. 2a, a malicious site *x.ex* uses aggressive redirection: when a user visits *x.ex*, it registers impressions under many Sybil domains posing as conversion sites, then auto-redirects the user through them. Each Sybil calls `measureConversion()`, consuming global budget. Since Attribution does per-querier checks, it can spend up to  $\epsilon_{\text{querier}}$ ; with  $\epsilon_{\text{global}}/\epsilon_{\text{querier}}$  Sybils, *x.ex* can exhaust a user’s global budget from a single visit. When the user later visits a legitimate merchant (e.g., *y.ex*), that merchant receives null reports – both useless and biased. If *x.ex* attracts many users even once, it can disable Attribution for benign queriers across the epoch.

Attacks need not be so overt. An attacker can instead embed many Sybil intermediaries (Fig. 2b), each requesting reports and drawing from distinct per-querier budgets. More subtly, it can pace Sybil activity over time or across repeated visits to gradually drain users’ budgets, making the attack harder to detect.

**Generic attack model.** The preceding examples illustrate specific tactics that an attacker can use to deplete global budgets in Attribution. Our objective, however, is to build defenses at the budget-management layer that withstand a broader class of attackers, regardless of how tactics evolve. We thus model a generic attacker that controls a set of domains visited by genuine users and aims to degrade the accuracy of benign queriers’ measurements on genuine user data by exhausting the global budget on users’ devices. We characterize the *attacker’s strength* along three dimensions: (i) how many domains it controls; (ii) how many intentional

user actions it can induce across those domains; and (iii) how many distinct genuine users perform these actions.

**Resilience goal.** The attacker’s impact in the Fig. 2 examples is *disproportionate to its strength*: the attacker is weak along one or more dimensions yet still depletes a device’s global budget. Our goal is to *restore proportionality* along all three dimensions of strength: depletion should scale not with Sybil count alone, but with genuine user actions and with the number of users it is able to attract, so that an attacker must be strong along all three dimensions to deplete substantially. We further require that this holds regardless of how the attacker behaves or what it knows, without assuming that benign traffic looks any particular way.

**Assumptions.** We exclude two classes of threats from our model. First, we do not address bot-driven *poisoning* attacks in which adversaries fabricate impressions or conversions (including null reports) to skew aggregates. These attacks degrade data quality rather than deplete global budgets on genuine devices, and defending against them requires mechanisms orthogonal to global-budget management, such as recently proposed fraud prevention for Attribution [6]. Our focus is on preserving measurement availability for genuine-user data – a prerequisite for meaningful measurement even without the poisoning threat.

Second, we protect only benign sites that do not directly or transitively partner with malicious ones (e.g., by embedding attacker-controlled intermediaries or serving their ads). Big Bird operates purely at the global-budget layer and has no visibility into partner relationships or ad-network governance; attacks that exploit such partnerships require ecosystem-level controls (registration, contracts), not budget logic. Our aim is to confine adversarial impact to sites that interact with Sybils, while unrelated benign sites retain utility.

#### 4.2 Candidate Approaches

The research challenge is to add DoS depletion resilience without overconstraining benign workloads, violating device-epoch IDP, or undermining its optimizations. Naïve approaches fail on at least one of these fronts.

**Rate-limit API calls.** One could cap how many impressions a site may log or how many reports it may request per time unit. However, API calls are a poor proxy for privacy loss. Under IDP, consumption depends on a user’s actual data contribution, and many events incur zero loss: most impressions never lead to conversions, and many conversions have no attributable impression. These would still count against an API-call limit, disabling IDP’s sparsity benefits. Worse, sites cannot know in advance which events will incur loss, so such caps force conservative throttling of legitimate activity and degrade utility. Any fixed limit therefore either suppresses benign workloads or must be set so loosely that attackers operate beneath it.

**Rate-limit total budget consumption.** A more aligned approach is to cap budget consumption itself, e.g., by limiting how much global budget can be spent over short time windows. This blocks bursty attacks like those in Fig. 2, but fails against paced strategies that spread consumption over time. For example, a site can exploit repeated or prolonged user presence (e.g., an open tab) to gradually drain budget via redirects or intermediary requests. Extending the time window to catch such attacks risks making the limiter itself a DoS target. Moreover, setting the cap below  $\epsilon_{\text{querier}}$  harms benign workloads, especially small advertisers that may legitimately consume their budget in bursts.

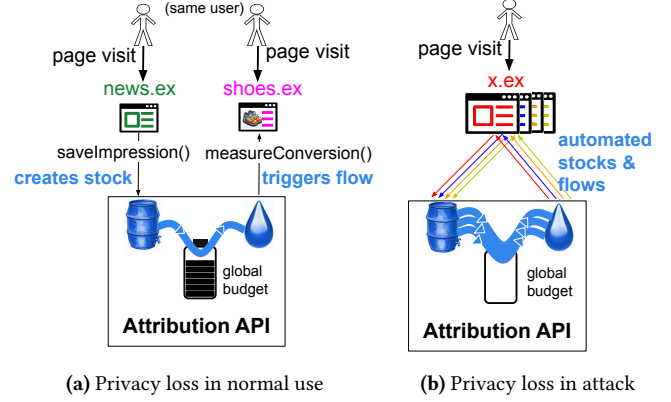
**Cap total budget consumption per user action.** A key weakness in Fig. 2 is that a single user action can drain a user’s entire global budget. A natural fix is to cap budget per user action: each action initializes a budget  $\epsilon_{\text{action}}$ , shared across all queriers triggered by that action. This ties consumption to a meaningful unit, but still allows a single high-engagement site to deplete budget over many interactions with the same user. Among candidate approaches, this is closest to Big Bird.

**Interactions of caps with device-epoch IDP.** We find it easy to define caps that violate device-epoch IDP by coupling decisions across epochs. For example, §C presents a natural but IDP-violating variant of the preceding approach: capping total budget consumption across all epochs following a user action. Any defense must therefore ensure that both checks and consequences are confined to a single epoch; in this paper, we show how to do so.

### 4.3 Big Bird Design

Based on the limitations of alternative approaches, we establish four principles for Big Bird’s design. **P1:** Limits should track *IDP budget consumption*, not proxies such as API-call counts, since IDP budget consumption is not always correlated to the number of API calls. **P2:** Defenses should target *minimal, intrinsic properties of legitimate Attribution use*, not specific attack tactics (e.g., redirection chains, embedded intermediaries, fast-fire vs. slow-burn attacks) or speculative assumptions about benign workloads (e.g., that benign traffic will never involve redirections or more than  $x$  intermediaries). This keeps defenses robust as both attacks and benign workloads evolve. **P3:** Global-budget use should be *ultimately tied to intentional user actions across distinct first-party sites*, since that is what Attribution is meant to measure. **P4:** Attribution’s device-epoch IDP must be preserved.

**Key insight: stock-and-flow.** Fig. 3 illustrates this pattern. The Attribution API is intended to measure *user interactions across distinct first-party sites*, such as impression and conversion sites. In normal Attribution use (Fig. 3a), privacy loss should only be consumed when there is something to measure: a user visits one site (and sees an ad) and later visits a distinct site (and converts there). That creates a stock-and-flow pattern for privacy budget consumption: (1) *stock* is

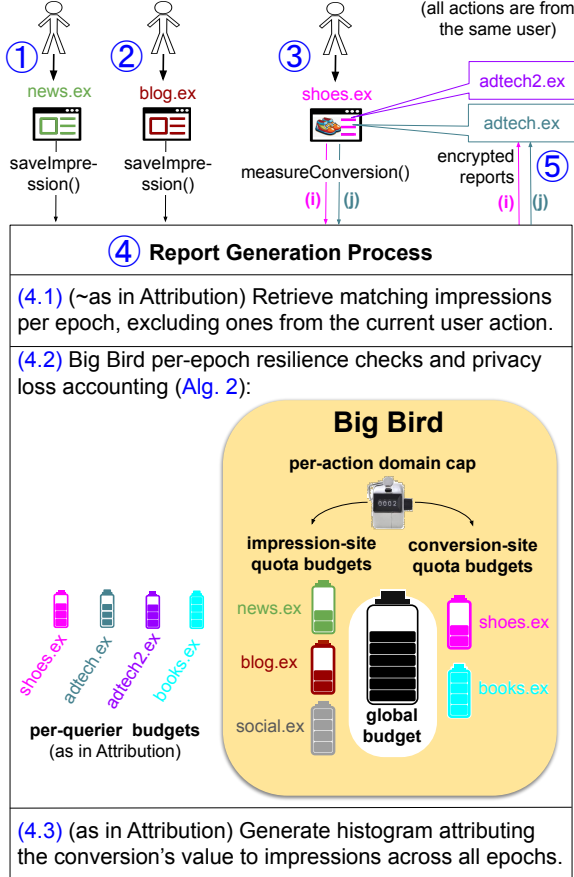


**Fig. 3. Stock-and-flow pattern of budget consumption in Attribution.** (a) Normal use: user actions on impression sites create stock and user actions on conversion sites trigger flows. (b) Attacks break this pattern by automatically creating many stocks and flows from very few user actions.

created when the user visits an impression site and the site saves an impression; and (2) *flow* is triggered when a user visits a conversion site and the site requests a report. Non-zero privacy loss is realized only when *both steps occur*, a property afforded by the IDP mechanism. By contrast, attacks disrupt this structure (Fig. 3b) by coalescing domains or automating stock creation and flow triggering (e.g., via automatic redirections or embedded intermediaries), allowing substantial global budget consumption from minimal user actions. Big Bird restores this pattern with privacy-loss-based quotas that cap *stock creation* and *flow triggering* for every caller alike, ultimately tying both to *intentional user actions*.

**Big Bird architecture.** Fig. 4 shows the Big Bird architecture. It sits as a management layer around Attribution’s global budgets and protects them from depletion by enforcing quotas that restore the expected stock-and-flow pattern, in each epoch: (1) *impression-site quotas* limit how much stock any one domain can contribute; (2) *conversion-site quotas* limit how much flow a domain can trigger; and (3) a *per-action domain cap* limits how many new domains can create quotas in the context of a single user action.

The first two quotas are privacy budgets implemented as IDP filters, just like global and per-querier budgets. However, their purpose is not privacy; they protect the global budget from depletion. These quotas account for individual privacy loss in a manner aligned with the global budget (principle **P1**), and constrain stock creation and flow triggering according to the expected structure of benign behavior (**P2**). The per-action domain cap is not a privacy budget; it simply restricts the number of quota budgets activated under different domains from a single user action. Together, these three quotas constrain consumption of the global budget, forcing the attacker to behave within the bounds of expected workload, and tie this restricted consumption to something tangible, such as intentional user actions across sites (**P3**). §4.4 formalizes the resilience properties afforded by this quota architecture, while our evaluation quantifies



**Fig. 4. Big Bird architecture.** Depicts budget structure for a single epoch, though in reality privacy and quota budgets exist for each epoch. them empirically. Finally, our quota enforcement is done separately for each epoch, ensuring that Big Bird preserves device-epoch IDP, a property we discuss in §4.5 (P4).

**Example execution.** Fig. 4 shows the workflow. ①–② A user sees ads on `news.ex` and `blog.ex`. ③ After purchasing on `shoes.ex`, two `measureConversion()` calls are issued: (i) one for `shoes.ex` (to measure a creative) and (j) one for `adtech2.ex` (to optimize its placement model). ④ Big Bird processes each call by selecting relevant impressions and attempting to deduct budgets atomically per epoch: querier, global, conversion-site, and matching impression-site quotas. If any deduction fails, that epoch’s contribution is dropped. ⑤ Two encrypted reports are returned, one per querier.

Reports (i) and (j) use parameters:  $\epsilon_{\text{privacy}}=0.2$  (privacy budget requested),  $\text{value}=75$  (conversion value), and  $\text{maxValue}=150$  (maximum conversion value). Assume the relevant impressions occur in epochs  $e1$  and  $e2$ , while the conversion occurs in  $e3$ . Tab. 1 shows the remaining budget and quota capacities after releasing reports (i) and (j), assuming all budgets start at the listed capacities. For report (i), following the IDP rule used in Attribution, each relevant epoch incurs privacy loss  $(\text{value}/\text{maxValue}) \cdot \epsilon = 0.1$ , while other epochs incur none. Only sites with matching impressions (`news.ex` in  $e1$  and `blog.ex` in  $e2$ ) consume

| epoch | querier<br>$\epsilon_{\text{querier}} = 0.5$ |           |            | global<br>$\epsilon_{\text{global}} = 4$ | imp-quota<br>$\epsilon_{\text{imp-quota}} = 2$ |         | conv-quota<br>$\epsilon_{\text{conv-quota}} = 0.75$ |
|-------|----------------------------------------------|-----------|------------|------------------------------------------|------------------------------------------------|---------|-----------------------------------------------------|
|       | shoes.ex                                     | adtech.ex | adtech2.ex |                                          | news.ex                                        | blog.ex |                                                     |
| e1    | 0.4                                          | 0.4       | 0.4        | 3.8                                      | 1.8                                            | 2       | 0.55                                                |
| e2    | 0.4                                          | 0.4       | 0.4        | 3.8                                      | 2                                              | 1.8     | 0.55                                                |
| e3    | 0.5                                          | 0.5       | 0.5        | 4                                        | 2                                              | 2       | 0.75                                                |

**Tab. 1. Quota and budget states after example execution.**

impression-site quota. Since all budgets for  $e1$  and  $e2$  have sufficient capacity, deductions commit atomically in each epoch, and the histogram is generated from these epochs. Report (j) performs the same computation but deducts from `adtech2.ex`’s querier budget instead of `shoes.ex`’s.

**Big Bird algorithm.** Alg. 2 enforces quotas while preserving device-epoch IDP. Each user action creates a context  $\text{uaCtx}$ , shared across all API calls causally triggered by that action. Browsers already have notions of intentional user action, namely the user-activation API [5], often used to gate access to various APIs. Within each  $\text{uaCtx}$ , Big Bird bounds the *number of distinct first-party domains*, not API calls. This prevents attackers from spawning many Sybil domains while allowing arbitrary activity within each domain. This matches normal web behavior: a benign user action may involve one or two redirects (e.g., email links or short URLs), but is not expected to traverse many domains. As a result, small per-action domain cap values (e.g., 2–3) should comfortably support benign workloads while preventing attackers from spawning many distinct quotas to fuel depletion.

Privacy accounting occurs in `measureConversion()`. For each epoch in the attribution window, the function (i) matches impressions, (ii) computes device-epoch IDP loss for the conversion and the finer-grained device-epoch–impression-site losses used by impression-site quotas, and (iii) attempts to deduct all relevant budgets in a single *atomic transaction*. The finer-grained accounting prevents wasting quota on impression sites that displayed no relevant impressions. If any deduction fails, that epoch’s data contribution is dropped with no budget impact; otherwise, all deductions commit atomically. This prevents budget waste, preserves the stock-and-flow structure, and ensures that decisions based on one epoch’s data never affect another’s privacy accounting.

A subtle but critical constraint shapes Alg.2’s structure. A natural alternative would be to enforce the per-action domain cap by computing, inside `measureConversion()`, how many impression-site quotas supply stock and how many conversion-site quotas trigger flow for the report, and to nullify the entire report if this exceeds a threshold. Big Bird deliberately avoids this. A single report may aggregate impressions and conversions from multiple epochs, and any check that reasons about those epochs jointly would couple their outcomes and thus their privacy losses. Because device-epoch IDP requires each epoch’s privacy loss to depend only on that epoch’s data, such cross-epoch checks are unsound.

Instead, Big Bird enforces the per-action domain cap using *strict per-epoch separation*. The count of activated quotas

## Alg. 2 Big Bird algorithm

```

def onUserAction(uaCtx)
  accessedSites[uaCtx][e] ← ∅ for each epoch e

def saveImpression(uaCtx, impSite, impression)
  e = getCurrentEpoch()
  if |accessedSites[uaCtx][e] ∪ {impSite}| > κaction then
    return // can't allow new quotas from this user action
  accessedSites[uaCtx][e] ∪ = {impSite}
  ImpressionStore[e].add(uaCtx, impression) // save uaCtx
  return

def measureConversion(uaCtx, impSites, querier, convSite, params)
  // Separately for each epoch, match impressions and do budget
  // checks and privacy accounting (Steps (4.1)-(4.3) in Fig. 4).
  for e ∈ params.attributionWindow do
    // Check the per-action domain cap.
    if |accessedSites[uaCtx][e] ∪ convSite| > κaction then
      Ie ← ∅ // drop epoch's impression data from consideration
      CONTINUE
    accessedSites[uaCtx][e] ∪ = {convSite}
    // Match impressions, filtering out impressions that querier and convSite
    // are not permitted to measure, including impressions from this uaCtx.
    Ie ← MatchingFn(uaCtx, ImpressionStore[e], params)
    // Compute IDP loss at two granularities w/ Cookie Monster algo.
    εe ← computeEpochLoss(Ie, params)
    for i ∈ impSites do
      εei ← computeEpochImpSiteLoss(Ie[i], params)
    // Execute budget checks & deductions as a transaction.
    tx = beginTransaction()
    if querierBudget[e][querier].tryConsume(εe, tx) and
      globalBudget[e].tryConsume(εe, tx) and
      convQuota[e][convSite].tryConsume(εe, tx) and
      (impQuota[e][i].tryConsume(εei, tx) ∀ i ∈ impSites) then
      tx.commit() // all budgets deducted successfully
    else // a budget deduction failed
      tx.abort() // undo any other budget deductions
      Ie ← ∅ // drop epoch's impression data from consideration
    // Generate attribution histogram from remaining impressions
  return AttributionFn(Ie1, ..., Iew) // ei ∈ params.attributionWindow

```

is maintained not only per `uaCtx` but also *per epoch*. If the cap is exceeded in epoch  $e$ , Big Bird nullifies only epoch  $e$ 's data contribution, forcing its privacy loss to zero while leaving all other epochs unaffected. No check aggregates privacy consumption across epochs, and no failure in one epoch propagates to another. This preserves device-epoch IDP while still enforcing meaningful DoS limits tied to user actions.

Finally, we note a subtlety of the atomic check-and-deduct procedure, specifically for impression-site quota deductions within an epoch. As shown in §5, atomicity strengthens resilience by forcing attackers to be selective about which Sybil impressions they declare as relevant: a single impression whose impression-site quota is exhausted suffices to nullify the entire epoch's contribution, preventing any global-budget consumption. This same mechanism can, however, harm benign workloads: if multiple impression sites are relevant in an epoch and one exhausts its quota, the querier cannot obtain attribution information from the remaining sites even if they retain budget. Supporting finer-grained

### “Normal” workload parameters:

**M**: max # of impression sites contributing nonzero loss in an epoch.  
**N**: max # of conversion sites requesting nonzero loss from an epoch.  
**n**: max # of conversion sites requesting nonzero loss from (epoch, impSite).  
**r**: max budget consumed by an intermediary's cross-advertiser queries on a single conversion site, as a fraction of the intermediary's  $\epsilon_{\text{querier}}$ .

| Cap                   | Capacity configuration                                                          |
|-----------------------|---------------------------------------------------------------------------------|
| Per-querier filter    | $\epsilon_{\text{querier}}$ : configuration parameter                           |
| Global filter         | $\epsilon_{\text{global}} = \max(N, n \cdot M)(1 + r)\epsilon_{\text{querier}}$ |
| Impression-site quota | $\epsilon_{\text{imp-quota}} = n(1 + r)\epsilon_{\text{querier}}$               |
| Conversion-site quota | $\epsilon_{\text{conv-quota}} = (1 + r)\epsilon_{\text{querier}}$               |
| per-action domain cap | $\kappa_{\text{action}} = 2 \text{ or } 3$                                      |

Tab. 2. Big Bird budget configurations.

behavior – e.g., selectively nullifying only the depleted impression site's contribution while preserving device-epoch IDP – appears feasible, but we leave its design to future work. Our strongest-attacker experimental results in §5 provide an upper bound on resilience if impression-site quotas are removed from the atomic transaction.

**Quota configuration.** We choose quota capacities via a utility-first approach: specify workload requirements, then select the smallest values that satisfy them. Taking the querier budget  $\epsilon_{\text{querier}}$  and intermediary fraction  $r$  as base parameters, and using browser-tunable parameters that reflect expected benign workload scale, we define four constraints: (i) a conversion site must be able to spend its full querier budget, and intermediaries may spend a fraction  $r$  of that, giving  $\epsilon_{\text{conv-quota}} \geq (1 + r)\epsilon_{\text{querier}}$ ; (ii) no single conversion site should drain the global budget, so with  $N$  expected conversion sites per epoch,  $\epsilon_{\text{global}} \geq N\epsilon_{\text{conv-quota}}$ ; (iii) likewise, the global budget should exceed an impression site's stock, so with  $M$  expected impression sites,  $\epsilon_{\text{global}} \geq M\epsilon_{\text{imp-quota}}$ ; and (iv) because multiple conversion sites may query the same impression site, a fan-out parameter  $n$  requires  $\epsilon_{\text{imp-quota}} \geq n\epsilon_{\text{conv-quota}}$ .

We then minimize  $\epsilon_{\text{global}}$  subject to these constraints, yielding the capacities in Tab. 2 and a tight global-privacy guarantee consistent with utility. Configuring Big Bird is thus a three-step procedure: (1) choose the base parameters  $\epsilon_{\text{querier}}$  and  $r$ ; (2) conservatively estimate the workload parameters  $M, N, n$  from a production sample at a chosen percentile; and (3) derive all quota capacities in closed form from Tab. 2. The percentile is the only real knob, trading utility against the tightness of  $\epsilon_{\text{global}}$ . §5.2 instantiates this procedure on Criteo.

## 4.4 Resilience Guarantee

Big Bird restores proportionality between attacker impact and strength along the three dimensions in §4.1. Its resilience, stated in Thm. 4.1, stems from: (a) per-domain quota limits, (b) user-action-based quota creation, and (c) bounded-sensitivity DP aggregation. Indeed, Attribution serves population-level *DP queries* in which each device-epoch contributes only through a report with *bounded sensitivity*. As a result, even if an attacker depletes budget on some device-epochs, the resulting error in a benign query

scales only with the number of affected device-epochs.

**Theorem 4.1** (Resilience to DoS depletion – proof in B.4). *Consider an attacker who (1) creates  $M^{\text{adv}}$  and  $N^{\text{adv}}$  *imp-quota* and *conv-quota* quotas, respectively; (2) collects  $U^{\text{adv}}$  user actions across the quotas’ domains for device  $d$ ; and (3) depletes budget from  $k$  device-epochs.*

*Denote by  $\epsilon_{\text{global}}^{\text{adv}}$  the maximum budget that the attacker can consume from the global budget on any epoch from  $d$ . Denote by  $\|Q(D) - \tilde{Q}(D)\|$  the error induced by the attacker, for a query  $Q$  where reports have sensitivity  $\Delta$  and at most  $m$  reports belong to any single device. Then, we have:*

- (i)  $\epsilon_{\text{global}}^{\text{adv}} \leq \min(M^{\text{adv}} \epsilon_{\text{imp-quota}}, N^{\text{adv}} \epsilon_{\text{conv-quota}})$
- (ii)  $\epsilon_{\text{global}}^{\text{adv}} \leq (1 + r) \epsilon_{\text{querier}} \cdot \kappa_{\text{action}} \cdot U^{\text{adv}}$
- (iii)  $\|Q(D) - \tilde{Q}(D)\| \leq m \cdot k \cdot \Delta$ , and  $k \leq U^{\text{adv}} / \lceil \frac{\epsilon_{\text{global}}}{n(1+r) \epsilon_{\text{querier}} \cdot \kappa_{\text{action}}} \rceil$ .

(i) Even if user actions were free, the global budget that an attacker can drain on a given device-epoch is capped by how many impression-site and conversion-site quotas it has managed to create, scaled by their capacities. This is the stock-and-flow constraint: to move global budget, the attacker must both create stock (impression-site quotas) and trigger flow (conversion-site quotas), and each is individually bounded. (ii) We connect this to intentional user actions via the per-action domain cap. Each user action can create at most  $\kappa_{\text{action}}$  new quotas of each type, and only within a single epoch. Across  $U^{\text{adv}}$  actions, the attacker can therefore unlock at most  $O(\kappa_{\text{action}} \cdot U^{\text{adv}})$  useful quotas, which yields the stated bound on  $\epsilon_{\text{global}}^{\text{adv}}$ . This enforces proportionality to user actions: substantial depletion now requires many genuine user interactions, not just many cheap domains. (iii) We lift the guarantee from a single device-epoch to the query level. With bounded report sensitivity  $\Delta$ , even if the attacker fully depletes the global budget on  $k$  device-epochs, its impact on any aggregate is linear in  $k$ .  $k$  itself is bounded by the number of actions across devices.

**Limitations.** We identify two limitations of the resilience guarantees provided by Big Bird. First, resilience assumes the attacker cannot scale simultaneously along all three strength dimensions. Enforcing this ultimately requires API-governance mechanisms beyond budget management. For instance, browsers could require API invokers to register with trusted authorities to limit Sybil domains, or adopt a conservative definition of *user activation* to ensure that interactions the user did not intend as attribution actions (e.g., clickjacking, deceptive UI, automatic redirects, or invisible frames) do not trigger budget consumption. Our contribution is to *expose the necessary control points* within the privacy-budget layer so that, when combined with governance mechanisms, Attribution can be rigorously hardened against attacks.

Second, Thm. 4.1 bounds depletion of the *global* budget, but quota budgets themselves may still be targeted – albeit only in ways excluded by our threat model (§4.1). For example, a malicious intermediary could exhaust its own conversion-site quota to block other intermediaries, or target the impression-site quota of a site that embeds its ads. These attacks are localized: the impact is confined to queriers inter-

acting (directly or transitively) with the affected site, while unrelated sites retain access to the global budget. Moreover, our quota configuration dampens such effects: exhausting an impression-site quota requires fan-out across  $n$  conversion sites, while exhausting a conversion-site quota requires  $(1 + r)$  queriers. Fully addressing quota-targeting attacks will require Attribution to support mechanisms for managing and subdividing quotas (e.g., per-partner allocation) to prevent abuse. We leave this to future work, as it likely requires integration with ecosystem-level constructs such as business relationships and auction pricing.

#### 4.5 Privacy Guarantee

Our privacy analysis of Big Bird aims for a high-fidelity model that provides confidence that the design satisfies global device-epoch-level IDP. We capture: individual DP; on-device report generation with centralized noise addition; all public information exposed by the protocol; query, budget, and data adaptivity; multiple epochs and queriers; the atomic interaction of multiple budgets within a single transaction; and user-action-based limits on how many quotas may be invoked. To our knowledge, this is one of the most faithful system models in the DP literature. Appendix B gives two versions: a simplified one for educational use (§B.5) and the complete version for formal analysis (§B.2). Using the complete model, we prove that Big Bird preserves *global device-epoch IDP*. Here and below,  $\Pr[V = v]$  denotes the joint density or probability mass at  $v$ , as appropriate.

**Theorem 4.2** (Global IDP Guarantee – proof in §B.3). *Consider  $x \in \mathcal{X}$  with global budget capacity  $\epsilon_{\text{global}}$ . Then, Big Bird satisfies individual device-epoch  $\epsilon_{\text{global}}$ -DP for  $x$  under public information  $C$ . That is, for every data generation process  $\mathcal{G}$  and all queriers  $\mathcal{A}^1, \dots, \mathcal{A}_k$ , the output  $V^{(\chi)}$  of Alg. 7 under challenge bit  $\chi$  satisfies, for all  $v \in \mathcal{R}^{\epsilon_{\text{max}} \times (t_{\text{max}} + 1) \times B}$ :*

$$\left| \ln \left( \frac{\Pr[V^{(0)} = v]}{\Pr[V^{(1)} = v]} \right) \right| \leq \epsilon_{\text{global}}.$$

#### 4.6 Prototype as Reference Implementation

We implement Big Bird both as a reference implementation for the emerging W3C standard and as a foundation for broader deployments, including in mobile OS location analytics. The implementation has two components. (1) **pdslib** is a general-purpose IDP Rust library that subsumes Cookie Monster and Big Bird while exposing a generic interface beyond Attribution. Clients (websites or mobile apps) register events (e.g., ad views, location visits), request reports (e.g., attributions, model updates), and receive encrypted, privacy-budgeted responses. (2) An integration in Firefox’s Attribution prototype [32] replaces report counting with full IDP and resilient enforcement based on pdslib. Mobile recommendation workloads, which share sparsity and structure with advertising, are a promising future instantiation.

### 5 Evaluation

We evaluate Big Bird’s utility and resilience on a real-world Criteo dataset, simulating DoS attackers inspired by Fig. 2a.

Our evaluation asks two questions: (1) How do quota configurations affect benign-query utility, both without attack and under a fixed-strength attack? (2) How does attacker strength along different resource dimensions translate into damage to benign queries?

### 5.1 Methodology

**Dataset.** CriteoPrivateAd [40] is a 30-day snapshot of production traffic (104M impressions across 220k publisher sites and 10k conversion sites) released for benchmarking W3C’s Attribution API. To our knowledge, it is the only real-world dataset with multiple conversion sites—a requirement for exercising Big Bird’s conversion-site quotas. The dataset includes timing, contextual features, conversions, and a daily-reset device ID. Because impressions are subsampled, most devices appear only once. Using Criteo’s resampling method, we reconstruct the device-level distribution, yielding *1.4M devices*, *4.6M impressions*, and *5.6M conversions*. The median device has 2 impressions and 4 conversions. While realistic, the dataset reflects a single intermediary and likely underestimates load in multi-intermediary deployments.

**Benign workload.** We evaluate single-advertiser measurement, where each batch of conversions produces a DP histogram over five contextual buckets. We evaluate utility using  $\text{RMSRE}_\tau$  [2], the root mean squared relative error across buckets, clipping counts below  $\tau = 5\%$  of batch size. This captures both DP noise (common across baselines) and bias from budget/quota depletion. To ensure low-noise baselines, we (i) restrict to advertisers with  $\geq 100$  conversions/day (526 total), (ii) batch  $\sim 10$  days of conversions (capped at 8,000), and (iii) set  $\epsilon$  to target  $\text{RMSRE}_\tau \approx 5\%$ , similar to how we expect an advertiser would tune their measurement.

**Attack workload.** We inject synthetic adversarial traffic atop real events. The attacker is given impression and conversion streams for the 10 busiest sites, reusing timestamps and device IDs to simulate organic traffic. Each event is treated as a user action enabling budget depletion. The attacker controls a pool of Sybil domains (25 by default) and, per user action, triggers a redirection chain (bounded by  $\kappa_{\text{action}}$ ). Each Sybil registers one fresh impression and requests a report on a previously unqueried impression with  $\epsilon = \epsilon_{\text{querier}}$ .

A naive attacker that references all Sybils per action is neutralized by Big Bird’s atomic deductions: a single over-quota Sybil invalidates the entire request, consuming zero global budget. We confirm this empirically and focus on two stronger strategies. The **Random Attacker** samples a subset of Sybils per conversion (35% by default, minimizing atomic rejection), modeling limited tracking capabilities. The **Omniscient Attacker** has perfect knowledge of each device’s remaining quotas and uses only Sybils with remaining quota, eliminating rejection waste and representing the optimal attack within the Attribution interface. Because these capabilities—perfect repeat-user recognition and full visibility into every device’s quota state—are ones real attackers

will likely lack, we read omniscient-attacker error as a deliberate worst-case ceiling rather than a typical operating point; real attackers fall between the two extremes.

**Baselines and defaults.** We compare Big Bird to two baselines spanning the utility–privacy spectrum: (1) *Attribution w/o global budget* (Cookie Monster [43]), which enforces per-querier IDP but no shared limit. It is immune to depletion and serves as the *utility ceiling*, but lacks global privacy guarantees. (2) *Attribution w/ global budget*, which adds a shared privacy filter but no quota mechanisms, representing the strongest defense without Big Bird.

Unless stated otherwise, we set  $\epsilon_{\text{querier}} = 1.0$ ,  $\epsilon_{\text{global}} = 8.0$ ,  $\epsilon_{\text{imp-quota}} = 2.0$ , and  $\epsilon_{\text{conv-quota}} = 1.0$ . We use  $\kappa_{\text{action}} = 2$  and 25 Sybils at maximum popularity. Epochs are one day, matching Criteo’s daily device-ID reset.

### 5.2 Quota Configuration on Criteo

We instantiate the three-step configuration procedure of §4.3 (“Quota configuration” header) on Criteo. *Step (1):* we use  $\epsilon_{\text{querier}} = 1$  (our default from §5.1) and, because our evaluation

Tab. 3. Workload params in Criteo.

| %ile | $\tilde{N}$ | $\tilde{M}$ | $\tilde{n}$ | $\epsilon_{\text{global}}$ | $\epsilon_{\text{imp-quota}}$ |
|------|-------------|-------------|-------------|----------------------------|-------------------------------|
| p50  | 2           | 1           | 2           | 2                          | 2                             |
| p80  | 4           | 2           | 2           | 4                          | 2                             |
| p85  | 4           | 2           | 4           | 8                          | 4                             |
| p90  | 4           | 3           | 4           | 12                         | 4                             |
| p95  | 6           | 3           | 4           | 12                         | 4                             |
| p99  | 8           | 4           | 8           | 32                         | 8                             |

models only single-advertiser queries,  $r = 0$ , giving  $\epsilon_{\text{conv-quota}} = \epsilon_{\text{querier}}$ ; general workloads would set  $r > 0$ . *Step (2):* the distributions of  $N, M, n$  across devices could be computed exactly by running full conversion attribution, but a deployer can instead use cheap, conservative per-device statistics that upper-bound them: unique impression sites ( $\tilde{M} \geq M$ ), unique conversion sites ( $\tilde{N} \geq N$ ), and unique conversion sites per (device, impression-site) pair ( $\tilde{n} \geq n$ ); Tab. 3 reports these statistics at several percentiles. *Step (3):* the same table shows the resulting capacities; e.g., the p85 estimates  $(\tilde{N}, \tilde{M}, \tilde{n}) = (4, 2, 4)$  yield  $\epsilon_{\text{global}} = \max(\tilde{N}, \tilde{n}\tilde{M})(1 + r)\epsilon_{\text{querier}} = 8$  and  $\epsilon_{\text{imp-quota}} = \tilde{n}(1 + r)\epsilon_{\text{querier}} = 4$ . Only these values are Criteo-specific: the procedure itself is dataset-driven, and any browser vendor can run it on its own production sample.

The percentile choice governs the tradeoff between quota size (and thus utility) and the tightness of the global DP guarantee  $\epsilon_{\text{global}}$ . Supporting 100% of devices maximizes utility, since it expects no quota-induced errors, but yields very loose  $\epsilon_{\text{global}}$ . The 85th percentile offers a balanced configuration: tight enough for a single-digit  $\epsilon_{\text{global}} = 8$  while avoiding quota errors for the vast majority of devices. As in §5.3, setting  $\epsilon_{\text{imp-quota}}$  below the table value (e.g.,  $\epsilon_{\text{imp-quota}} = 2$  instead of 4) substantially improves resilience at a negligible utility cost: the resulting slack in  $\epsilon_{\text{global}}$  ensures enough budget is left for benign queries even under attack.

### 5.3 Utility and Resilience with Quota Configuration

We fix all default parameters and sweep the impression-site quota  $\epsilon_{\text{imp-quota}}$  and per-action domain cap  $\kappa_{\text{action}}$  to deter-

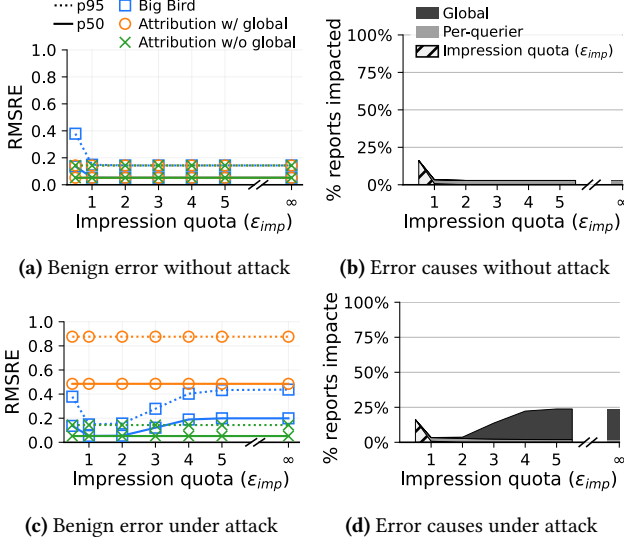


Fig. 5. Utility and resilience with quota configuration. (a), (b): Benign error and root causes *without* attack. (c), (d): Benign error and root causes *under* the random attacker.

mine the range of quota values that preserve benign utility while confining the attacker.

**Benign error without attack (Fig. 5a, Fig. 5b).** We first measure utility in a purely benign setting, varying the impression-site quota at  $\epsilon_{global} = 8$  (derived from the p85 workload in Tab. 3 with  $\epsilon_{querier} = 1$ ). Fig. 5a plots median and 95th-percentile  $RMSRE_\tau$  as  $\epsilon_{imp-quota}$  grows. Both Attribution baselines, which lack impression-site quotas, show constant and identical error across  $\epsilon_{imp-quota}$  (median 5.2%). Their agreement confirms that the global budget alone does not reduce utility without an attack, an important sanity check that isolates quota effects from global-budget effects. In contrast, Big Bird performs poorly at  $\epsilon_{imp-quota} = 0.5$ , where aggressive quota exhaustion produces null reports (median error 14.0%, a  $2.7\times$  increase). Once  $\epsilon_{imp-quota}$  reaches 2, quotas no longer affect accuracy: Big Bird matches the baselines at 5.2%. This holds through arbitrarily large quota values, including Big Bird with the impression-site quota disabled entirely. The p85 quotas from Tab. 3 therefore suffice for normal operation and appear conservative, suggesting that true workload parameters for most devices are well below the upper-bound estimates we use to size  $N$ ,  $M$ , and  $n$ .

Fig. 5b breaks down the bias component of Big Bird’s  $RMSRE_\tau$  by attributing each blocked report to the first budget in priority order that was exhausted. At  $\epsilon_{imp-quota} = 0.5$ , the impression-site quota blocks 16.2% of reports, explaining the elevated error in Fig. 5a. At  $\epsilon_{imp-quota} = 1$ , this drops to 0.8%. For  $\epsilon_{imp-quota} \geq 2$ , the impression-site quota blocks fewer than 0.002% of reports; the only residual blocking comes from per-querier budgets (2.8% of reports). This residual appears in all baselines, and is inherent to IDP systems when device contributions exhaust the querier’s budget. The takeaway is that, once quotas are sized to the p85 workload, Big Bird

introduces no bias beyond what the underlying Attribution accounting already imposes.

**Benign error under attack (Fig. 5c, Fig. 5d).** We next evaluate resilience when an attacker targets the global budget. Fig. 5c reports error for benign queries.

Attribution w/o global budget is unaffected (median 5.2%), as expected: without a global budget, there is nothing to deplete. But this immunity comes at the cost of no global privacy guarantee (§3). Attribution w/ global budget suffers severe degradation: its median  $RMSRE_\tau$  reaches 48.5%, a  $9.3\times$  increase, and its p95 reaches 87.6%. This confirms Big Bird’s premise: simply adding a global budget to fix per-querier unsoundness is trivially defeated by Sybil attacks.

In contrast, Big Bird achieves near-baseline utility *and* the privacy guarantee of a system with a global budget. With a well-sized quota, it nearly matches Attribution w/o global budget at a median of 5.5% and a p95 of 14.8%. Moreover, Big Bird is robust across a meaningful range of impression-site quota values  $\epsilon_{imp-quota} \in [1, 2]$ , for which the budget the attacker can consume remains well below  $\epsilon_{global}$ , leaving substantial slack in the global budget even after the attack.

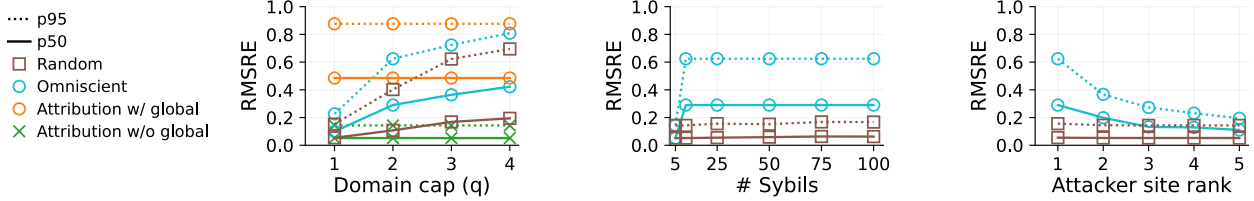
We also observe that, as in the benign case, quotas that are too small degrade utility. But under attack, quotas that are *too large* also degrade utility, for a fundamentally different reason: they allow the attacker to route enough traffic through Sybils to drain the global budget despite the quota mechanism. At  $\epsilon_{imp-quota} = 3$ , median error rises to 12.5% and by  $\epsilon_{imp-quota} = 5$  it plateaus near 20.1%. Fig. 5d corroborates this transition: for  $\epsilon_{imp-quota}$  up to 2, fewer than 1% of benign reports are blocked by global-budget depletion; at  $\epsilon_{imp-quota} = 3$  this rises to approximately 11%, and at  $\epsilon_{imp-quota} = 4$  to approximately 19%, dominating all other error sources.

#### Impact of per-action domain cap and attacker strategy.

We now fix  $\epsilon_{imp-quota} = 2$  (our default) and sweep  $\kappa_{action}$  from 1 to 4 to evaluate how the per-action domain cap impacts utility under attack. Additionally, we compare the omniscient attacker to the random attacker, revealing how adversary sophistication affects resilience. Fig. 6a shows that across both attackers,  $\kappa_{action} = 1$  keeps median benign  $RMSRE_\tau$  at or below 10.0%, and  $\kappa_{action} = 2$  (our default) at or below 28.6%. Both remain far below the unprotected Attribution API (48.3–48.4%).  $\kappa_{action} = 1$  minimizes worst-case error at the expense of flexibility in the benign case, allowing only one site per user action. Big Bird’s quota mechanism provides the structural foundation for resilience, bounding depletion even against the optimal attacker. Atomic deduction dramatically reduces damage against more realistic adversaries: the random attacker achieves only 5.4% at  $\kappa_{action} = 2$ , versus the omniscient attacker’s 28.6%.

#### 5.4 Impact of Attacker Strength

The preceding section fixed attacker strength and varied the defense configuration. We now fix the defense ( $\kappa_{action} =$



**Fig. 6. (a) Impact of per-action domain cap and (b), (c) attacker strength on resilience.** (b) Number of attacker-controlled Sybil domains capable of accessing the API and (c) popularity of the attacker-controlled sites in terms of number of user actions across all distinct users in the Criteo dataset.

2,  $\epsilon_{\text{imp-quota}} = 2$ ) and vary attacker resources to test the guarantee from §4.1: damage should scale with the number of Sybil domains (dimension (i)) and the volume of genuine user traffic the attacker attracts (dimensions (ii) and (iii)).

**Number of Sybil domains (Fig. 6b).** We vary the attacker’s Sybil pool from 5 to 100 domains, holding site popularity at maximum. The random attacker rises only from 5.2% at 5 Sybils to 6.4% at 100, as the tight  $\epsilon_{\text{imp-quota}} = 2$  quota sharply limits per-domain budget consumption regardless of pool size. The omniscient attacker saturates early:  $\text{RMSRE}_\tau$  reaches 29.4% at 10 Sybils and remains flat through 100. With perfect tracking, 10 domains already exhaust the available per-device quota headroom; additional Sybils provide no benefit once the global budget becomes the bottleneck. This matches Thm. 4.1(i): depletion is bounded by the product of domain count and quota capacity, so once  $M^{\text{adv}} \cdot \epsilon_{\text{imp-quota}}$  exceeds the global budget, extra Sybils add nothing.

**Site popularity (Fig. 6c).** We vary which of Criteo’s busiest sites the attacker controls, from the top 10 sites to ranks 41–50. Lower-ranked sites have fewer user actions and fewer distinct users, jointly exercising dimensions (ii) and (iii) of the threat model. The random attacker falls from 5.4% to 5.2%, remaining near baseline across all ranks — the tight source quota renders the random attacker essentially harmless regardless of popularity. The omniscient attacker falls from 29.0% to 10.9%, a  $2.7\times$  reduction. This confirms that damage scales with the volume of genuine user traffic the attacker attracts, as Thm. 4.1(ii–iii) bounds depletion as a function of  $U^{\text{adv}}$  and the number of affected devices. This dimension is critical: attracting genuine traffic requires costly real-world investment, unlike cheap domain registration.

## 6 Additional Related Work

§3 covers DP enforcement granularities. Other work follows.

**Ad-measurement APIs.** Our primary application is the W3C Attribution API. Cookie Monster [43] introduced device-epoch IDP accounting. Google’s ARA line [2, 10, 16], IPA [30], PAM [3], and Hybrid [31] explored alternative, albeit now obsolete, privacy architectures. In these proposals, per-querier enforcement is treated as a natural semantic, with collusion recognized as the primary limitation. We show that even *without* collusion, per-querier DP is unsound. Attribution’s measurement-only scope, with

targeting unsupported, is being debated [20]; the debate concerns what the standard should protect, not whether its guarantees hold. We argue that measurement is a sensible place to start, separable from targeting and already treated distinctly by some U.S. privacy laws [8].

**Privacy budget management.** This paper contributes to the broad challenge of *privacy budget management*, a crucial but understudied area in DP. Budget allocation *within* a query is well studied [1, 27], but managing budgets *across* queriers, especially untrusted ones, remains relatively unexplored. Pujol et al. [36, 37] balance utility across analysts sharing a global budget, but without per-analyst utility assurances. Relevant systems include those for global budget scheduling [23, 28, 42], but none of these works address a Sybil DoS threat. Finally, work on privacy filters [12, 14, 25, 29, 38] provides building blocks on which Big Bird implements budgets.

**DoS and quota mechanisms.** DoS defense is studied extensively in systems and networking (e.g., [4, 9, 39, 41]), and quotas are ubiquitous for conventional resources. But these approaches do not apply directly to privacy budgets, whose consumption reflects *data contribution*, not request volume. To our knowledge, no work treats privacy budgets as a *DoS target* and provides Sybil-resilient budget management that preserves DP semantics and utility. Big Bird’s novelty does not rest on the notion of quota, but on how the quota *architecture* is shaped by DP semantics and workload structure.

## 7 Conclusions

We study DP enforcement in the W3C Attribution draft standard for privacy-preserving advertising and find that its current reliance on per-querier device-epoch IDP is not a sound foundation: even without collusion, realistic multi-querier behaviors such as cross-querier adaptivity and shared limits can violate its guarantees. We therefore advocate global device-epoch IDP and present Big Bird, a budget manager that makes it deployable by tying privacy-budget consumption to the stock-and-flow structure of benign workloads via IDP-aligned quotas and per-action domain caps. Our results show that Big Bird achieves rigorous global privacy, resilience to depletion attacks, and strong utility under adversarial conditions. Both contributions are now part of the draft standard and have broader implications for DP systems in open, untrusted environments.

## Acknowledgments

We thank the anonymous reviewers and the PATCG/PATWG participants for their constructive feedback. Nikos Goutzoulis contributed to the implementation of an early prototype. This work was supported by NSF grant CNS-2143868, NSF Cooperative Agreement No. EEC-2133516, and Google, Meta, Microsoft, and Sloan Faculty Fellowships. Co-author Lécuyer acknowledges the Canada CIFAR AI Chair program and Amii. Co-author Geambasu was partially employed by Meta during this project, and subsequently by Google; the work was done exclusively in the context of Geambasu’s Columbia University research.

## References

- [1] John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineke, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. 2022. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review Special Issue 2* (6 2022). doi:10.1162/99608f92.529e3cb9
- [2] Hidayet Aksu, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Adam Sealfon, and Avinash V. Varadarajan. 2024. Summary Reports Optimization in the Privacy Sandbox Attribution Reporting API. *Proc. Priv. Enhancing Technol.* 2024, 4 (2024), 605–621. doi:10.56553/POPETS-2024-0132
- [3] Apple. 2023. Private Ad Measurement (PAM). <https://github.com/patcg-individual-drafts/private-ad-measurement>.
- [4] Nirav Atre, Hugo Sadok, Erica Chiang, Weina Wang, and Justine Sherry. 2022. SurgeProtector: Mitigating Temporal Algorithmic Complexity Attacks using Adversarial Scheduling. In *Proceedings of the 2022 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, New York, NY, USA, 723–738. doi:10.1145/3544216.3544250
- [5] Marcos Caceres. 2023. The User Activation API. <https://webkit.org/blog/13862/the-user-activation-api/>.
- [6] Rutchathon Chairattana-Apirom, Stefano Tessaro, and Nirvan Tyagi. 2025. Fraud Mitigation in Privacy-Preserving Attribution. *Cryptology ePrint Archive, Paper 2025/1891*. <https://eprint.iacr.org/2025/1891>
- [7] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Transactions on Information and System Security* 14, 3 (2011), 26:1–26:24. doi:10.1145/2043621.2043626
- [8] Colorado General Assembly. 2025. Colorado Privacy Act, Section 6-1-1303 (25)(b)(IV). <https://law.justia.com/codes/colorado/title-6/fair-trade-and-restraint-of-trade/article-1/part-13/section-6-1-1303/>.
- [9] Scott A. Crosby and Dan S. Wallach. 2003. Denial of Service via Algorithmic Complexity Attacks. In *12th USENIX Security Symposium (USENIX Security 03)*. USENIX Association, Washington, D.C. <https://www.usenix.org/conference/12th-usenix-security-symposium/denial-service-algorithmic-complexity-attacks>
- [10] Matthew Dawson, Badih Ghazi, Pritish Kamath, Kapil Kumar, Ravi Kumar, Bo Luan, Pasin Manurangsi, Nishanth Mundru, Harikesh Nair, Adam Sealfon, and Shengyu Zhu. 2023. Optimizing Hierarchical Queries for the Attribution Reporting API. In *Proceedings of the Workshop on Data Mining for Online Advertising (AdKDD 2023) Co-located with the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2023), Long Beach, CA, USA, August 7, 2023 (2023) (CEUR Workshop Proceedings, Vol. 3556)*, Abraham Bagherjeiran, Nemanja Djuric, Kuang-chih Lee, Linsey Pang, Vladan Radosavljevic, and Suju Rajan (Eds.). CEUR-WS.org. <https://ceur-ws.org/Vol-3556/adkdd23-dawson-optimizing-ceur-paper.pdf>
- [11] Sergey Denisov, H. Brendan McMahan, John Rush, Adam Smith, and Abhradeep Guha Thakurta. 2022. Improved Differential Privacy for SGD via Optimal Private Linear Operators on Adaptive Streams. *Advances in Neural Information Processing Systems* 35 (2022), 5910–5924. [https://papers.nips.cc/paper\\_files/paper/2022/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html](https://papers.nips.cc/paper_files/paper/2022/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html)
- [12] David Durfee and Ryan M Rogers. 2019. Practical Differentially Private Top-k Selection with Pay-what-you-get Composition. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 3527–3537.
- [13] Hamid Ebad, David Sands, and Gerardo Schneider. 2015. Differential Privacy: Now It’s Getting Personal. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Mumbai India, 2015-01-14). ACM, 69–81. doi:10.1145/2676726.2677005
- [14] Vitaly Feldman and Tijana Zrnic. 2021. Individual Privacy Accounting via a Rényi Filter. In *Advances in Neural Information Processing Systems* (2021), M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 28080–28091. [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/ec7f346604f518906d35ef0492709f78-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/ec7f346604f518906d35ef0492709f78-Paper.pdf)
- [15] Marco Gaboardi, James Honaker, Gary King, Jack Murtagh, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. 2018. PSI (Ψ): A Private Data Sharing Interface. (2018). arXiv:1609.04340 [cs, stat] <http://arxiv.org/abs/1609.04340>
- [16] Badih Ghazi, Charlie Harrison, Arpana Hosabettu, Pritish Kamath, Alexander Knop, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Mariana Raykova, Vikas Sahu, and Philipp Schoppmann. 2025. On the Differential Privacy and Interactivity of Privacy Sandbox Reports. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (2025), 382–397. doi:10.56553/popets-2025-0104
- [17] Google. 2022. *Attribution Reporting: Full System Overview*. Privacy Sandbox. <https://privacysandbox.google.com/private-advertising/attribution-reporting/system-overview>
- [18] Abhradeep Guha Thakurta and Adam Smith. 2013. (Nearly) Optimal Algorithms for Private Online Learning in Full-information and Bandit Settings. In *Advances in Neural Information Processing Systems* (2013), Vol. 26. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2013/hash/c850371fda6892fbfd1c5a5b457e5777-Abstract.html>
- [19] Charlie Harrison. 2025. Safety limits enable possible DOS attacks. <https://github.com/w3c/attribution/issues/69>.
- [20] Kyle Hogan, Alishah Chator, Gabriel Kaptchuk, Mayank Varia, and Srinivas Devadas. 2026. Making Sense of Private Advertising: A Principled Approach to a Complex Ecosystem. *Proc. Priv. Enhancing Technol.* 2026, 1 (2026), 450–469. doi:10.56553/POPETS-2026-0023
- [21] Palak Jain, Sofya Raskhodnikova, Satchit Sivakumar, and Adam Smith. 2023. The Price of Differential Privacy under Continual Observation. In *Proceedings of the 40th International Conference on Machine Learning* (2023-07-03). PMLR, 14654–14678. <https://proceedings.mlr.press/v202/jain23b.html>
- [22] Zach Jorgensen, Ting Yu, and Graham Cormode. 2015. Conservative or Liberal? Personalized Differential Privacy. In *2015 IEEE 31st International Conference on Data Engineering* (2015-04). 1023–1034. doi:10.1109/ICDE.2015.7113353
- [23] Nicolas Küchler, Emanuel Opel, Hidde Lycklama, Alexander Viand, and Anwar Hithnawi. 2024. Cohere: Managing Differential Privacy in Large Scale Systems. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 991–1008. doi:10.1109/SP54263.2024.00122
- [24] Nicolas Küchler, Alexander Viand, Hidde Lycklama, and Anwar Hithnawi. 2025. DPolicy: Managing Privacy Risks Across Multiple Releases with Differential Privacy. In *2025 IEEE Symposium on Security and Privacy (SP)* (2025-05). 3950–3968. doi:10.1109/SP61157.2025.00256
- [25] Mathias Lécuyer. 2021. Practical Privacy Filters and Odometers with Rényi Differential Privacy and Applications to Differentially Private

- Deep Learning. arXiv:2103.01379 [stat.ML] <https://arxiv.org/abs/2103.01379>
- [26] Mathias Lécuyer, Riley Spahn, Kiran Vodrahalli, Roxana Geambasu, and Daniel Hsu. 2019. Privacy Accounting and Quality Control in the Sage Differentially Private Machine Learning Platform. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP)*. ACM, 181–195. doi:10.1145/3341301.3359639
- [27] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. 2010. Optimizing linear counting queries under differential privacy. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Indianapolis, Indiana, USA) (*PODS '10*). Association for Computing Machinery, New York, NY, USA, 123–134. doi:10.1145/1807085.1807104
- [28] Tao Luo, Mingen Pan, Pierre Tholoniati, Asaf Cidon, Roxana Geambasu, and Mathias Lécuyer. 2021. Privacy Budget Scheduling. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. USENIX Association, 55–74. <https://www.usenix.org/conference/osdi21/presentation/luo>
- [29] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. AIM: an adaptive and iterative mechanism for differentially private synthetic data. *Proc. VLDB Endow.* 15, 11 (July 2022), 2599–2612. doi:10.14778/3551793.3551817
- [30] Meta and Mozilla. 2022. Interoperable Private Attribution (IPA). <https://github.com/patcg-individual-drafts/ipa>.
- [31] Meta and Mozilla. 2024. Hybrid Proposal. <https://github.com/patcg-individual-drafts/hybrid-proposal>.
- [32] Mozilla. 2024. Experiment: Privacy-Preserving Attribution Measurement API. <https://github.com/mozilla/explainers/tree/main/archive/ppa-experiment>.
- [33] Andrew Paseltiner, Andy Leiserson, Benjamin Case, Benjamin Savage, Charlie Harrison, and Martin Thomson. 2026. Attribution Level 1. <https://w3c.github.io/attribution/>.
- [34] Andrew Paseltiner, Andy Leiserson, Benjamin Case, Benjamin Savage, Charlie Harrison, and Martin Thomson. 2026. Attribution Level 1, 4.2.1. Privacy Budget Store. <https://www.w3.org/TR/2026/WD-attribution-20260813/#s-privacy-budget-store>.
- [35] Andrew Paseltiner, Andy Leiserson, Benjamin Case, Benjamin Savage, Charlie Harrison, and Martin Thomson. 2026. Attribution Level 1, 7.1.1. Formal Analysis of Privacy Properties and Their Limitations. <https://www.w3.org/TR/2026/WD-attribution-20260813/#formal-analysis>.
- [36] David Pujol, Albert Sun, Brandon Fain, and Ashwin Machanavajjhala. 2022. Multi-Analyst Differential Privacy for Online Query Answering. *Proc. VLDB Endow.* 16, 4 (2022), 816–828. doi:10.14778/3574245.3574265
- [37] David Pujol, Yikai Wu, Brandon Fain, and Ashwin Machanavajjhala. 2021. Budget sharing for multi-analyst differential privacy. *Proc. VLDB Endow.* 14, 10 (June 2021), 1805–1817. doi:10.14778/3467861.3467870
- [38] Ryan Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. 2016. Privacy odometers and filters: pay-as-you-go composition. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) (*NIPS'16*). Curran Associates Inc., Red Hook, NY, USA, 1921–1929.
- [39] C.L. Schuba, I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram, and D. Zamboni. 1997. Analysis of a denial of service attack on TCP. In *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097)*. 208–223. doi:10.1109/SECPRI.1997.601338
- [40] Mehdi Sebbar, Corentin Odic, Mathieu Léchine, Aloïs Bissuel, Nicolas Chrysanthos, Anthony D'Amato, Alexandre Gilotte, Fabian Höring, Sarah Nogueira, and Maxime Vono. 2025. CriteoPrivateAds: A Real-World Bidding Dataset to Design Private Advertising Systems. arXiv:2502.12103 [cs.CR] <https://arxiv.org/abs/2502.12103>
- [41] Oliver Spatscheck and Larry L. Peterson. 1999. Defending Against Denial of Service Attacks in Scout. In *3rd Symposium on Operating Systems Design and Implementation (OSDI 99)*. USENIX Association, New Orleans, LA, 59–72. <https://www.usenix.org/conference/osdi-99/defending-against-denial-service-attacks-scout>
- [42] Pierre Tholoniati, Kelly Kostopoulou, Mosharaf Chowdhury, Asaf Cidon, Roxana Geambasu, Mathias Lécuyer, and Junfeng Yang. 2025. DPack: Efficiency-Oriented Privacy Budget Scheduling. In *Proceedings of the Twentieth European Conference on Computer Systems* (Rotterdam, Netherlands) (*EuroSys '25*). Association for Computing Machinery, New York, NY, USA, 1194–1209. doi:10.1145/3689031.3696096
- [43] Pierre Tholoniati, Kelly Kostopoulou, Peter McNeely, Prabhpreet Singh Sodhi, Anirudh Varanasi, Benjamin Case, Asaf Cidon, Roxana Geambasu, and Mathias Lécuyer. 2024. Cookie Monster: Efficient On-Device Budgeting for Differentially-Private Ad-Measurement Systems. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles* (New York, NY, USA, 2024-11-15) (*SOSP '24*). Association for Computing Machinery, 693–708. doi:10.1145/3694715.3695965
- [44] Shufan Zhang and Xi He. 2023. DProvDB: Differentially Private Query Processing with Multi-Analyst Provenance. *Proc. ACM Manag. Data* 1, 4, Article 267 (Dec. 2023), 27 pages. doi:10.1145/3626761

Note: This appendix has not been peer-reviewed.

## A Per-Querier DP Analysis (§3 Proofs)

### A.1 Data Adaptivity

We provide more details on the model introduced in §3.2. Our formalism with a data generation process follows existing work [26]. Other adaptive stream definitions have been proposed [18, 21]. Adaptive data generation definitions capture stronger adversaries than non-adaptive adjacency definitions for streams [7]. Note that when there is only one querier, a single adversary can be both the data generation process and the querier.

Without loss of generality, the opt-out record and the time at which it is inserted are known upfront. We could let  $\mathcal{G}$  adaptively decide when and what to insert, while enforcing that this happens at most once, but this decision would not actually depend on the challenge bit and would therefore follow a distribution known by  $\mathcal{G}$  upfront.

Thm. 3.1 shows that data adaptivity results in excess per-querier privacy loss growing linearly in the number of queriers. The proof exhibits queriers and a data generation process, where results from queriers  $Q^1, \dots, Q^n$  are first encoded into the dataset itself.  $Q^{n+1}$  later decodes this information to gain additional confidence about the value of  $\chi$  beyond what its own queries should allow according to the per-querier filter.

**Theorem 3.1.** Consider Alg. 1, where  $\mathcal{F}^k$  are instantiated with pure DP filters [38], each with capacity  $\epsilon$ . Denote by  $V^{(\chi)}$  the output of the algorithm on challenge bit  $\chi$ .

When  $K = 1$ , Alg. 1 reduces to single-querier fully adaptive composition on adaptively generated streams. We have:

$$\forall x_0 \in \mathcal{X}, \forall t \in [t_{\max}], \forall \mathcal{G}, \forall Q^1, \\ \forall S \subseteq \mathcal{R}^{t_{\max}}, \left| \ln \left( \frac{\Pr[V^{1,(0)} \in S]}{\Pr[V^{1,(1)} \in S]} \right) \right| \leq \epsilon.$$

However, for  $K = n + 1 > 1$ , there exists  $\epsilon' > 0$  such that:

$$\exists x_0 \in \mathcal{X}, \exists t_0 \in [t_{\max}], \exists \mathcal{G}, \exists (Q^1, \dots, Q^{n+1}), \\ \exists S \subseteq \mathcal{R}^{t_{\max}}, \left| \ln \left( \frac{\Pr[V^{n+1,(0)} \in S]}{\Pr[V^{n+1,(1)} \in S]} \right) \right| = \epsilon + n\epsilon' > \epsilon.$$

*Proof.* When  $K = 1$ , we are exactly in the setting of Lécuyer et al. [26]. Briefly, when data can change adaptively based on past results, inserting a single record can have cascading effects on future data and future queries. Still, after conditioning on a prefix  $V_1^1, \dots, V_t^1$ , the databases in both worlds differ only by at most one record.

Now, consider  $K = n + 1$  with  $n > 0$ . Take  $x_0 = 1$  and  $t_0 = 1$ . Without loss of generality, take  $t_{\max} = 2$  and  $\mathcal{X} = \{1\}$ . We use histogram notation for the database, but since  $\mathcal{X}$  is a singleton, we write  $D \in \mathbb{N}$  instead of  $D \in \mathbb{N}^{\mathcal{X}}$ . In multiset notation, we could take  $\sum_{x \in D} x$  to get an integer representation of the database. Define  $\mathcal{G}$  and  $Q^1, \dots, Q^n, Q^{n+1}$  as follows:

1. At  $t = 1$ :
  - a. The data generation process does not return any data:  $\mathcal{G}(\emptyset, \dots, \emptyset) = \emptyset$ . Thus  $D_1 = 1$  if  $\chi = 1$ , and  $D_1 = 0$  otherwise, that is:

$$D_1 = \chi \quad (1)$$

- b. For  $k \in \{1..n\}$ ,  $Q^k$  submits  $q_1^k : D \mapsto D + \text{Lap}(1/\epsilon)$ , which consumes  $\epsilon$  from the per-querier filter  $\mathcal{F}^k$ .
  - c.  $Q^K$  submits a query that consumes no budget from the per-querier filter:  $Q^K(\emptyset) = q_1^K : D \mapsto 0$ .

2. At  $t = 2$ :

- a. The data generation process inserts up to  $n$  new records, conditionally on outputs  $V_1^1, \dots, V_1^n$  from  $Q^1, \dots, Q^n$ :  $\mathcal{G}(V_1^1, \dots, V_1^K) = |\{k \in \{1..n\} : V_1^k > 1/2\}|$ . That is:

$$D_2 = \sum_{k=1}^n \mathbb{1}[D_1 + Z_k > 1/2] \quad (2)$$

where  $Z_k \sim \text{Lap}(1/\epsilon)$ . Since  $V_1^k$  is more likely to be greater than  $1/2$  if  $\chi = 1$ ,  $D_2$  is more likely to be high if  $\chi = 1$ . In other words,  $\mathcal{G}$  encodes what  $Q^1, \dots, Q^n$  have learned into the dataset itself.

- b. For  $k \in \{1..n\}$ ,  $Q^k$  submits a query that consumes no budget from the per-querier filter:  $Q^k(V_1^k) = q_2^k : D \mapsto 0$ .
  - c.  $Q^K$  submits a query that consumes  $\epsilon$  from the per-querier filter  $\mathcal{F}^K$ :  $Q^K(V_1^K) = q_2^K : D \mapsto D + \text{Lap}(1/\epsilon)$ , and therefore:

$$V_2^K = D_1 + D_2 + Z \quad (3)$$

where  $Z \sim \text{Lap}(1/\epsilon)$ . We will show that  $V_2^K$  is disproportionately more likely to be high if  $\chi = 1$ . In other words,  $Q^K$  retrieves the information encoded into the dataset, that originally came from  $Q^1, \dots, Q^n$ .

Recall that  $\Pr[Z_k > 1/2] = \frac{1}{2}e^{-\epsilon/2}$ . Also,  $\Pr[1 + Z_k > 1/2] = \Pr[-Z_k < 1/2] = 1 - \frac{1}{2}e^{-\epsilon/2}$ , since  $-Z_k \sim Z_k$ . Set  $p := \frac{1}{2}e^{-\epsilon/2}$ .  $D_2$  follows a binomial distribution:  $D_2^{(0)} \sim \mathcal{B}(n, p)$  and  $D_2^{(1)} \sim \mathcal{B}(n, 1 - p) \sim n - \mathcal{B}(n, p)$ . Hence, Eq. 3 gives:

$$\begin{cases} V_2^{K,(0)} \sim \mathcal{B}(n, p) + \text{Lap}(1/\epsilon) \\ V_2^{K,(1)} \sim 1 + n - \mathcal{B}(n, p) + \text{Lap}(1/\epsilon) \end{cases} \quad (4)$$

We already have  $\mathbb{E}[V_2^{K,(0)}] = np < n/2$  and  $\mathbb{E}[V_2^{K,(1)}] = 1 + n(1 - p) > 1 + n/2$ . We now exhibit an event for which  $V_2^{K,(0)} \ll V_2^{K,(1)}$ .

Take  $B \sim \mathcal{B}(n, p)$ . Take  $N \geq n + 1$ . Recall that since  $Z \sim \text{Lap}(\epsilon)$ , for all  $z \geq 0$  and  $t \in \mathbb{R}$  such that  $z + t \geq 0$ , we have:

$$\Pr[Z \geq z] = e^{\epsilon t} \Pr[Z \geq z + t] \quad (5)$$

On one hand, we have  $\Pr[V^0 \geq N] = \Pr[B + Z \geq N] =$

$\Pr[Z \geq N - B]$ . On the other hand,  $\Pr[V^1 \geq N] = \Pr[1 + n - B + Z \geq N] = \Pr[Z \geq N - 1 - n + B]$ . Since  $N - 1 - n + B \geq 0$ , we have:  $\Pr[Z \geq N - 1 - n + B] = e^\epsilon \Pr[Z \geq N - n + B]$ . Hence,

$$\frac{\Pr[V^1 \geq N]}{\Pr[V^0 \geq N]} = e^\epsilon \frac{\Pr[Z \geq N - n + B]}{\Pr[Z \geq N - B]} \quad (6)$$

For all  $k \in [0, n]$  we have  $N - n + k \geq 0$  and  $N - k \geq 0$ , so we can compute the exact probabilities:

$$\begin{aligned} \Pr[Z \geq N - n + B] &= \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} \Pr[Z \geq N - n + k] \\ &= \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} \frac{1}{2} e^{-(N-n+k)\epsilon} \\ &= \frac{1}{2} e^{-N\epsilon} \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} (e^\epsilon)^{n-k} \\ &= (p + (1-p)e^\epsilon)^n \end{aligned}$$

Similarly,

$$\begin{aligned} \Pr[Z \geq N - B] &= \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} \Pr[Z \geq N - k] \\ &= \frac{1}{2} e^{-N\epsilon} \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} (e^\epsilon)^k \\ &= (pe^\epsilon + 1 - p)^n \end{aligned}$$

Hence,

$$\frac{\Pr[V^1 \geq N]}{\Pr[V^0 \geq N]} = e^\epsilon \left( \frac{p + (1-p)e^\epsilon}{pe^\epsilon + 1 - p} \right)^n$$

Moreover, we have:

$$\begin{aligned} p + (1-p)e^\epsilon - (pe^\epsilon + 1 - p) &= (1-2p)e^\epsilon - (1-2p) \\ &= (1-2p)(e^\epsilon - 1) \\ &> 0 \end{aligned}$$

since  $p < 1/2$  and  $\epsilon > 0$ . Define  $\epsilon' := \ln \left( \frac{p+(1-p)e^\epsilon}{pe^\epsilon+1-p} \right) > 0$ . Then:

$$\ln \left( \frac{\Pr[V^1 \geq N]}{\Pr[V^0 \geq N]} \right) = \epsilon + n\epsilon'$$

□

## A.2 Shared Limits

Since we know that data adaptivity already threatens per-querier budget semantics, we consider a simpler setting where data is known upfront. Alg. 3 presents this variant of Alg. 1. Thm. 3.2 then shows that adding a shared filter results in excess per-querier privacy loss, also growing linearly in the number of queriers. The proof exhibits  $n$  queriers  $Q_1, \dots, Q_n$  that first ask queries. These queriers then adaptively encode what they have learned into the privacy parameters of their next queries. The shared privacy filter tracks privacy loss across queriers, so the state of that filter now depends on all queriers' past queries. Finally, a querier  $Q_{n+1}$  can decode the information stored into the shared filter by ob-

serving whether its own queries get rejected by the shared filter. This gives  $Q_{n+1}$  additional information about the dataset, beyond what its own per-querier filter allows. This happens without different queriers sharing their DP outputs, which is what people have referred to as collusion before. In fact, even if  $Q_1, \dots, Q_n$  do not follow a particular strategy, and  $Q_{n+1}$  doesn't know about them,  $Q_{n+1}$ 's view can still be impacted beyond what its per-querier filter allows.

---

### Alg. 3 Formal model for shared limits

---

```

1: Input
2: Database  $D$ 
3: Queriers  $Q^1, \dots, Q^K$ 
4: Per-querier filters  $\mathcal{F}^1, \dots, \mathcal{F}^K$ 
5: Shared limit  $\mathcal{L}$ 
6: Output
7: View  $V = (V^1, \dots, V^K) \in \mathcal{R}^{t_{\max} \times K}$  of all  $Q^k$ 
8: for  $t \in [t_{\max}]$  do
9:   for  $k \in [K]$  do
     // Query based on  $Q^k$ 's own previous results
10:   $q_t^k \leftarrow Q^k(V_{<t}^k)$ 
     // Execute if (1)  $Q^k$  has enough budget and (2) shared
     // limit accepts the query.
11:  if  $\mathcal{F}^k(q_1^k, \dots, q_t^k) \wedge \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^K)$  then
12:     $V_t^k \leftarrow q_t^k(D)$ 
13:  else
14:     $V_t^k \leftarrow \perp$ 

```

---

**Theorem 3.2.** Take  $K = n + 1 > 1$ , per-querier filters with capacity  $\epsilon$  and shared filter with capacity  $\epsilon_g < (n+1)\epsilon$  (allowing executions that trigger the shared limit).

For  $n$  large enough that  $\lceil \frac{n}{2} e^{-\epsilon/2} \rceil < n/2$ , there exists neighboring databases  $D, D'$  such that:

$$\begin{aligned} &\exists(Q^1, \dots, Q^{n+1}), \exists S \subseteq \mathcal{R}^{t_{\max}}, \\ &\left| \ln \left( \frac{\Pr[V^{n+1,(0)} \in S]}{\Pr[V^{n+1,(1)} \in S]} \right) \right| > \epsilon. \end{aligned}$$

Moreover, there exists  $\epsilon' > 0$  and neighboring databases  $D, D'$  such that for all  $n \geq \frac{4 \ln 2}{(1-e^{-\epsilon/2})^2}$ ,

$$\begin{aligned} &\exists(Q^1, \dots, Q^{n+1}), \exists S \subseteq \mathcal{R}^{t_{\max}}, \\ &\left| \ln \left( \frac{\Pr[V^{n+1,(0)} \in S]}{\Pr[V^{n+1,(1)} \in S]} \right) \right| \geq \epsilon + n\epsilon'. \end{aligned}$$

*Proof.* Consider  $D = 0, D' = 1$  from  $\mathcal{N}^X = \mathbb{N}$  with  $X = \{1\}$ . Define  $Q^1, \dots, Q^n$  as follows. Take  $\epsilon_0, \epsilon_1, \epsilon_2 > 0$  that we will calibrate later.

1. At  $t = 1$ :
  - a. For  $k \in \{1..n\}$ ,  $Q^k$  submits  $q_1^k : D \mapsto D + \text{Lap}(1/\epsilon_1)$ , which consumes  $\epsilon_1$ .
  - b. Then,  $Q^{n+1}$  submits a query that consumes all but  $\epsilon_0$  from its budget:  $Q^{n+1}(\emptyset) = q_1^{n+1} : D \mapsto D + \text{Lap}(1/(\epsilon - \epsilon_0))$ .
2. At  $t = 2$ :

- a. For  $k \in \{1..n\}$ ,  $Q^k$  submits a query that consumes no budget if  $V_1^k$  was small, or  $\epsilon_2$  if  $V_1^k$  was large:  $Q^k(V_1^k) = q_2^k : D \mapsto \mathbb{1}[V_1^k > 1/2](D + \text{Lap}(1/\epsilon_2))$ . In other words,  $Q^k$  encodes the information it learned at  $t = 1$  into the shared limit itself, through the adaptively chosen budget associated with its request at time  $t = 2$ .
- b.  $Q^{n+1}$  submits a query that attempts to consume  $\epsilon_0$ :  $Q^{n+1}(V_1^{n+1}) = q_2^{n+1} : D \mapsto D + \text{Lap}(1/\epsilon_0)$ , and therefore  $V_2^{n+1} \sim D + \text{Lap}(1/\epsilon_0)$  if the shared limit and the per-querier filter both accept the query, and  $V_2^{n+1} = \perp$  otherwise. In other words,  $Q^{n+1}$  retrieves the information encoded into the shared limit that originally came from  $Q^1, \dots, Q^n$ 's own requests.

Now, we configure  $\epsilon_1$  and  $\epsilon_2$  to ensure that no query prior to  $V_2^{n+1}$  gets filtered. That is,  $\epsilon - \epsilon_0 + n(\epsilon_1 + \epsilon_2) \leq \epsilon_g = n_0\epsilon$  and  $\epsilon_1 + \epsilon_2 \leq \epsilon$ . To saturate the global filter when the first  $n$  queriers all ask for  $\epsilon_2$ , we want:

$$\epsilon - \epsilon_0 + n(\epsilon_1 + \epsilon_2) = \epsilon_g \quad (7)$$

This means that the state  $f_2^{n,(\chi)}$  of the shared filter just before query  $t = 2, k = n + 1$ , on input  $D^{(\chi)}$ , is:

$$f_2^n = n\epsilon_1 + \epsilon - \epsilon_0 + \sum_{k=1}^n \Pr[V_1^k > 1/2]\epsilon_0 \quad (8)$$

Set  $p = \frac{1}{2}e^{-\epsilon/2}$ . We have:

$$\begin{cases} f_2^{n,(0)} \sim n\epsilon_1 + \epsilon - \epsilon_0 + \mathcal{B}(n, p)\epsilon_2 \\ f_2^{n,(1)} \sim n\epsilon_1 + \epsilon - \epsilon_0 + (n - \mathcal{B}(n, p))\epsilon_2 \end{cases} \quad (9)$$

In particular,  $\mathbb{E}[f_2^{n,(0)}] = \epsilon - \epsilon_0 + n(\epsilon_1 + p\epsilon_2)$  and  $\mathbb{E}[f_2^{n,(1)}] = \epsilon - \epsilon_0 + n(\epsilon_1 + (1 - p)\epsilon_2)$ . We take  $\epsilon_2$  so that  $Q^{n+1}$  is more likely to trigger the global filter on  $D^{(1)}$  and more likely to succeed on  $D^{(0)}$ :

$$n\epsilon_1 + \epsilon - \epsilon_0 + n\epsilon_2/2 + \epsilon_0 = \epsilon_g \quad (10)$$

Subtracting Eq. 7 from Eq. 10, we get  $\epsilon_0 - n\epsilon_2/2 = 0$ , i.e.,  $\epsilon_2 = 2\epsilon_0/n$ .

Define  $B^{(\chi)} \sim \chi(n - B) + (1 - \chi)B$ , the number of queriers that ask for  $\epsilon_2$  at  $t = 2$ , with  $B \sim \mathcal{B}(n, p)$  as in the proof of Thm. 3.1. By construction,  $\epsilon_0$  passes  $Q^{n+1}$ 's per-querier filter, so we can only return  $\perp$  if  $\epsilon_0$  hits the shared filter's capacity:

$$V_2^{n+1,(\chi)} = \perp \iff f_2^{n,(0)} + \epsilon_0 > \epsilon_g \quad (11)$$

$$\iff n\epsilon_1 + \epsilon - \epsilon_0 + B^{(\chi)}\epsilon_2 + \epsilon_0 > \epsilon - \epsilon_0 + n(\epsilon_1 + \epsilon_2) \quad (12)$$

$$\iff B^{(\chi)}\epsilon_2 + \epsilon_0 > n\epsilon_2 \quad (13)$$

$$\iff 2B^{(\chi)}/n + 1 > 2 \quad (14)$$

$$\iff B^{(\chi)} > n/2 \quad (15)$$

We now upper-bound  $\Pr[B^{(0)} > n/2]$  and lower-bound

$\Pr[B^{(1)} > n/2]$ . By Hoeffding's inequality, we have:

$$\Pr[B^{(0)} > n/2] = \Pr[B > n/2] \quad (16)$$

$$\leq \Pr[B \geq n/2] \quad (17)$$

$$= \Pr[B - np \geq n(1/2 - p)] \quad (18)$$

$$\leq \exp(-2(1/2 - p)^2 n) \quad (19)$$

$$(20)$$

Similarly:

$$\Pr[B^{(1)} > n/2] = \Pr[n - B > n/2] \quad (21)$$

$$= 1 - \Pr[B \geq n/2] \quad (22)$$

$$\geq 1 - \exp(-2(1/2 - p)^2 n) \quad (23)$$

Finally, set  $S = \{(v, \perp), v > 1\}$ . Since  $\Pr[V_1^{n+1,(1)} > 1] = e^{\epsilon - \epsilon_0} \Pr[V_1^{n+1,(0)} > 1]$  and  $V_1^{n+1,(\chi)}$  and  $V_2^{n+1,(\chi)}$  are independent, we get:

$$\frac{\Pr[V^{n+1,(1)} \in S]}{\Pr[V^{n+1,(0)} \in S]} = e^{\epsilon - \epsilon_0} \frac{\Pr[V_2^{n+1,(1)} = \perp]}{\Pr[V_2^{n+1,(0)} = \perp]} \quad (24)$$

$$= e^{\epsilon - \epsilon_0} \frac{\Pr[B^{(1)} > n/2]}{\Pr[B^{(0)} > n/2]} \quad (25)$$

$$\geq e^{\epsilon - \epsilon_0} (\exp(2(1/2 - p)^2 n) - 1) \quad (26)$$

Now, since the median of  $B$  is  $\lfloor np \rfloor$  or  $\lceil np \rceil$ , we already have  $\Pr[B^{(0)} > n/2] < 1/2 < \Pr[B^{(1)} > n/2]$  when  $\lceil np \rceil < n/2$ , which happens for  $n$  large enough or  $p$  small enough (i.e.,  $\epsilon$  large enough).

We can get a more informative bound asymptotically. Set  $\alpha_n := \exp((1/2 - p)^2 n)$ . We have  $\frac{\Pr[V^{n+1,(1)} \in S]}{\Pr[V^{n+1,(0)} \in S]} \geq \alpha_n^2 - 1$ . For  $\alpha_n$  large enough, for instance any  $\alpha_n \geq 2$ , we have  $\alpha_n^2 - 1 \geq \alpha_n$ . We have  $\alpha_n \geq 2 \iff n \geq \frac{\ln 2}{(1/2 - p)^2} = \frac{4 \ln 2}{(1 - e^{-\epsilon/2})^2}$ , so for such  $n$  we get:

$$\frac{\Pr[V^{n+1,(1)} \in S]}{\Pr[V^{n+1,(0)} \in S]} \quad (27)$$

$$\geq e^{\epsilon - \epsilon_0} \alpha_n \quad (28)$$

$$= e^{\epsilon - \epsilon_0} (\exp((1/2 - p)^2 n))^n \quad (29)$$

That is:

$$\left| \ln \left( \frac{\Pr[V^{n+1,(0)} \in S]}{\Pr[V^{n+1,(1)} \in S]} \right) \right| \geq \epsilon + n\epsilon' > \epsilon \quad (30)$$

with  $\epsilon' := (1/2 - p)^2 > 0$ .  $\square$

**Remark about atomicity.** We clarify notation for Alg. 1, Line 14, where a query is executed only if both the per-querier filter and the shared limit permit it. The joint condition

$$\mathcal{F}^k(q_1^k, \dots, q_t^k) \wedge \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^K)$$

means that each filter takes its decision independently. In particular, if  $\mathcal{F}^k$  accepts a query  $q_t^k$  while  $\mathcal{L}$  rejects it,  $q_t^k$  will not be executed, but  $\mathcal{F}^k$  will not know that fact when it is time to make a decision about  $q_{t+1}^k$ . That means  $\mathcal{F}^k$  can be overly conservative, by deducting budget for queries that are in fact not executed. While this preserves privacy, it can hurt utility. We propose a solution to this conundrum in §4, thanks to a new form of atomic filter that is designed to avoid spending per-querier budget needlessly. This can be modeled by a new object  $(\mathcal{F}^k \circ \mathcal{L})(q_1^1, \dots, q_1^K, \dots, q_t^k)$  that has access to all the previous queries. The shared state in this object means that per-querier budgets act as heuristics (as opposed to formal per-querier DP guarantees) unless a dedicated siloing assumption holds.

### A.3 Extension to Individual DP filters

The model from Alg. 1 considers standard filters [38], where queriers can directly observe whether a query gets rejected. However, systems such as Cookie Monster [43] or ARA [16] operate under individual DP filters [14], which behave differently. Because an IDP budget depends on the data itself, IDP filters cannot return  $\perp$  explicitly when they are out of budget. Instead, they silently drop the out-of-budget individuals from the dataset and return a noisy answer.

This raises the question: since filter outputs are not visible under IDP, do shared limits still leak extra information? We give a positive answer. First, we adapt the formalism to IDP filters in Alg. 4, with changes from Alg. 3 in blue. Next, Thm. A.1 shows a variant of Thm. 3.2 for IDP filters. Finally, Thm. 3.1 immediately applies since the counterexample from the proof does not trigger any filter, and therefore works identically under IDP filters.

**Theorem A.1.** *Take  $K = n + 1 > 1$ , per-querier individual filters with capacity  $\epsilon$  and shared individual filter with capacity  $\epsilon_g < (n+1)\epsilon$  (allowing executions that trigger the shared limit). There exists  $\epsilon' > 0$  and neighboring databases  $D, D'$  such that*

$$\exists(Q^1, \dots, Q^{n+1}), \exists S \subseteq \mathcal{R}^{t_{\max}},$$

$$\left| \ln \left( \frac{\Pr[V^{n+1,(0)} \in S]}{\Pr[V^{n+1,(1)} \in S]} \right) \right| \geq \epsilon + n\epsilon'.$$

*Proof.* We modify the execution from the proof for Thm. 3.2 to match IDP semantics, where records get dropped conditionally on past results. This emulates a form of data adaptivity, except that datasets  $S_2^{n+1}$  now differ because of shared filters for various records, instead of through a data generation process. We then reuse the calculation from the proof of Thm. 3.2.

The changes in execution are as follows:

- $D'$  contains  $n+1$  distinct records  $\{x_0, x_1, \dots, x_n\}$  and  $D$  only contains  $\{x_1, \dots, x_n\}$ . We no longer have  $D \in \mathbb{N}$ , instead  $D \in \mathbb{N}^X$  with  $X = \{x_0, \dots, x_n\}$ .
- At  $t = 1$ , queriers submit  $q_1^k : D \mapsto |\{x \in D : x =$

### Alg. 4 Formal model for shared limits with IDP filters

```

1: Input
2: Database  $D$ 
3: Queriers  $Q^1, \dots, Q^K$ 
4: Per-querier individual filters  $(\mathcal{F}_x^1, \dots, \mathcal{F}_x^K)_{x \in D}$ 
5: Shared limit  $(\mathcal{L}_x)_{x \in D}$ 
6: Output
7: View  $V = (V^1, \dots, V^K) \in \mathcal{R}^{t_{\max} \times K}$  of all  $Q^k$ 
8: for  $t \in [t_{\max}]$  do
9:   for  $k \in [K]$  do
      // Query based on  $Q^k$ 's own previous results
10:   $q_t^k \leftarrow Q^k(V_{<t}^k)$ 
      // Execute if (1)  $Q^k$  has enough budget and (2) shared
      // limit accepts the query.
11:   $S_t^k \leftarrow D$ 
12:  for  $x \in D$  do
13:    if  $\mathcal{F}_x^k(q_1^k, \dots, q_t^k) \wedge \mathcal{L}_x(q_1^1, \dots, q_1^K, \dots, q_t^k)$  then
      // Drop out-of-budget records.
14:     $S_t^k \leftarrow S_t^k \setminus \{x\}$ 
15:   $V_t^k \leftarrow q_t^k(S_t^k)$ 

```

$x_0\} + \text{Lap}(1/\epsilon)$ . This consumes  $x_0$ 's full individual budget.

- At  $t = 2$ :
  - For  $k \in [n]$ , querier  $k$  requests  $x_k$  with full budget, conditionally on past results. That is:
 
$$Q^k(V_1^k) = q_2^k : D \mapsto \mathbb{1}[V_1^k \leq 1/2] \cdot (|\{x \in D : x = x_k\}| + \text{Lap}(1/\epsilon))$$
  - Finally, querier  $n+1$  requests all the records  $x_1, \dots, x_n$ :
 
$$Q^{n+1}(V_1^{n+1}) = q_2^{n+1} : D \mapsto |\{x \in D : x \in \{x_1, \dots, x_n\}\}| + \text{Lap}(1/\epsilon)$$

Since  $\{x_1, \dots, x_n\}$  are present in both worlds,  $V^{n+1,(1)}$  and  $V^{n+1,(0)}$  differ only by the points that get dropped at Line 14. Point  $x_k$  gets dropped only if  $V_1^k \leq 1/2$ , which is more likely to happen in world 0 (on input  $D$ ). More precisely, we have:

$$V^{n+1,(0)} \sim B + Z \text{ and } V^{n+1,(1)} \sim n - B + Z \quad (31)$$

with  $B \sim \mathcal{B}(n, p)$  with  $p = \frac{1}{2} \exp(-\epsilon/2)$  and  $Z \sim \text{Lap}(1/\epsilon)$ .

This is exactly the same situation as Eq. 6, where the  $e^\epsilon$  factor comes from the privacy loss at  $t = 1$ . We conclude the same way.  $\square$

### A.4 Querier-Siloing Assumption

**Lemma A.2.** *Consider Alg. 1 with  $K \in \mathbb{N}$ , per-querier filters with capacity  $\epsilon > 0$  and shared limit  $\mathcal{L}$ . Suppose that the*

following assumption holds:

$$\forall k \in [K], \forall u, w \in \mathcal{R}^t : u^k = w^k, \quad (32)$$

$$\begin{aligned} & \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(u_{<t}^1), \dots, q_t^k(u_{<t}^k)) \cdot q_t^k(D_{\leq t}(u_{<t})) \\ &= \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(w_{<t}^1), \dots, q_t^k(w_{<t}^k)) \cdot q_t^k(D_{\leq t}(w_{<t})) \end{aligned}$$

Then, for all  $k \in [K]$ :

$$\begin{aligned} & \forall x_0 \in \mathcal{X}, \forall t_0 \in [t_{\max}], \forall \mathcal{G}, \forall (Q^1, \dots, Q^{n+1}), \\ & \forall S \subseteq \mathcal{R}^{t_{\max}}, \left| \ln \left( \frac{\Pr[V^{k,(0)} \in S]}{\Pr[V^{k,(1)} \in S]} \right) \right| \leq \epsilon. \end{aligned}$$

In other words, if querier  $k$ 's results don't depend on other queriers, either through data or shared limits, then Alg. 1 is per-querier  $\epsilon$ -DP for querier  $k$ .

*Proof.* Take  $x \in \mathcal{X}$ ,  $\mathcal{G}$ ,  $Q^1, \dots, Q^K$ ,  $k \in [K]$ , and  $v^k \in \mathcal{R}_{\max}^t$ . We have:

$$\Pr[V^k = v^k] = \prod_{t=1}^{t_{\max}} \Pr[V_t^k = v_t^k | V_1^k = v_1^k, \dots, V_{t-1}^k = v_{t-1}^k]$$

Take  $t \in [t_{\max}]$ . We now express  $V_t^k$  in terms of queries and filter decisions, since the filter properties are necessary for the privacy proof. Without loss of generality, we treat filters as functions that return 0 for reject and 1 for accept at Line 14, and we output 0 instead of  $\perp$  at Line 17. Hence, we can express  $V_t^k$  as:

$$V_t^k = \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^k) \cdot q_t^k(D_{\leq t}) \quad (33)$$

The terms in this expression depend on various variables:

- Once we condition on  $v_1^k, \dots, v_{t-1}^k$ , we know the queries asked by  $Q^k$ :

$$q_1^k = Q^k(), q_2^k = Q^k(v_1^k), \dots, q_t^k = Q^k(v_1^k, \dots, v_{t-1}^k). \quad (34)$$

- As a consequence,  $\mathcal{F}^k(q_1^k, \dots, q_t^k)$  is also fully determined by  $v_1^k, \dots, v_{t-1}^k$ .
- However,  $\mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^k)$  can depend on other queriers' past results  $V_t^i$  for  $i \neq k$ , since these outputs are used to select  $q_t^i$ .
- Similarly,  $D_{\leq t}$  can also depend on other queriers' past results.

Pose  $V^{\neq k} := (V_i^j)_{i \in [t], j \in [K] \setminus \{k\}}$ . By the law of total probability, we have:

$$\begin{aligned} & \Pr[V_t^k = v_t^k | V_{<t}^k = v_{<t}^k] \\ &= \int_{v^{\neq k} \in \mathcal{R}^{t \times (K-1)}} \Pr[V_t^k = v_t^k | V_{<t}^k = v_{<t}^k, V^{\neq k} = v^{\neq k}] \\ & \quad \cdot \Pr[V^{\neq k} = v^{\neq k}] dv^{\neq k} \end{aligned} \quad (35)$$

Take any  $v^{\neq k} \in \mathcal{R}^{t \times (K-1)}$ . Define  $u \in \mathcal{R}^{t \times K}$  by concatenating  $v^k$  and  $v^{\neq k}$  such that  $u^k = v^k$  and  $u^{\neq k} = v^{\neq k}$ . Thanks

to Eq. 33 with explicit conditioning on all past results, we have:

$$\begin{aligned} & \Pr[V_t^k = v_t^k | V_1^k = v_1^k, \dots, V_{t-1}^k = v_{t-1}^k, V^{\neq k} = v^{\neq k}] \\ &= \mathcal{F}^k(q_1^k, \dots, q_t^k(u_{<t}^k)) \\ & \quad \cdot \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(u_{<t}^1), \dots, q_t^k(u_{<t}^k)) \\ & \quad \cdot q_t^k(D_{\leq t}(u_{<t})) \end{aligned} \quad (36)$$

Define  $\tilde{v} \in \mathcal{R}^{t \times K}$  by concatenating  $v^k$  and  $(\perp, \dots, \perp) \in \mathcal{R}^{t \times (K-1)}$ . We have  $\tilde{v}^k = u^k$ , so by Eq. 32, Eq. 36 becomes:

$$\Pr[V_t^k = v_t^k | V_1^k = v_1^k, \dots, V_{t-1}^k = v_{t-1}^k, V^{\neq k} = v^{\neq k}] \quad (37)$$

$$= \mathcal{F}^k(q_1^k, \dots, q_t^k(\tilde{v}_{<t}^k)) \quad (38)$$

$$\cdot \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(\tilde{v}_{<t}^1), \dots, q_t^k(\tilde{v}_{<t}^k)) \quad (39)$$

$$\cdot q_t^k(D_{\leq t}(\tilde{v}_{<t})) \quad (40)$$

This term is independent of  $v^{\neq k}$ , so we can express it as a function of  $v_{<t}^k$  and take it out of the integral at Eq. 35, which gives:

$$\Pr[V_t^k = v_t^k | V_1^k = v_1^k, \dots, V_{t-1}^k = v_{t-1}^k] \quad (41)$$

$$= \left( \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_t^k) \cdot q_t^k(D_{\leq t}) \right) \left( v_{<t}^k \right) \quad (42)$$

Now we can bound the per-querier privacy loss:

$$\begin{aligned} & \left| \ln \left( \frac{\Pr[V^k = v^k]}{\Pr[V^k = v^k]} \right) \right| \\ &= \left| \ln \left( \prod_{t=1}^{t_{\max}} \frac{\Pr[V_t^{k,(0)} = v_t^k | V_1^{k,(0)} = v_1^k, \dots, V_{t-1}^{k,(0)} = v_{t-1}^k]}{\Pr[V_t^{k,(1)} = v_t^k | V_1^{k,(1)} = v_1^k, \dots, V_{t-1}^{k,(1)} = v_{t-1}^k]} \right) \right| \\ &\leq \sum_{t=1}^{t_{\max}} \left| \ln \left( \frac{\Pr \left[ \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(0)}) \{v_{<t}^k\} = v_t^k \right]}{\Pr \left[ \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(1)}) \{v_{<t}^k\} = v_t^k \right]} \right) \right| \\ &\leq \sum_{t=1}^{t_{\max}} \left| \ln \left( \frac{\Pr[\mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(0)}) \{v_{<t}^k\} = v_t^k]}{\Pr[\mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(1)}) \{v_{<t}^k\} = v_t^k]} \right) \right| \end{aligned} \quad (43)$$

The last inequality comes from what follows.  $\mathcal{L}(q_1^1, \dots, q_t^k)$  is entirely determined by  $v_{<t}^k$  and thus identical in the numerator and denominator. For  $t$  where  $\mathcal{L}(q_1^1, \dots, q_t^k) = 1$ , the  $\mathcal{L}$  factor is one and can be omitted from the product. For

$t$  where  $\mathcal{L}(q_1^1, \dots, q_t^k) = 0$ , we have:

$$\begin{aligned} & \left| \ln \left( \frac{\Pr \left[ \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_t^k) \right]}{\Pr \left[ \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \mathcal{L}(q_1^1, \dots, q_t^k) \right]} \right) \right| \\ &= 0 \leq \left| \ln \left( \frac{\Pr[\mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(0)})\{v_{< t}^k\} = v_t^k]}{\Pr[\mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot q_t^k(D_{\leq t}^{(1)})\{v_{< t}^k\} = v_t^k]} \right) \right| \end{aligned}$$

The rest of the proof is identical to standard filter guarantees on adaptively generated data. Suppose the filter uses an upper bound  $\epsilon_i$  on the privacy loss for each query at step  $i \in [t_{\max}]$ :

$$\left| \ln \left( \frac{\Pr[q_i^k(D_{\leq i}^{(0)})\{v_{< i}^k\} = v_i^k]}{\Pr[q_i^k(D_{\leq i}^{(1)})\{v_{< i}^k\} = v_i^k]} \right) \right| \leq \epsilon_i(v_{< i}^k) \quad (44)$$

Then, by definition of pure DP filters, we get the following recursive property for  $t \in [t_{\max}]$ :  $\mathcal{F}^k(q_1^k, \dots, q_t^k) = 1$  iff  $\sum_{i < t} \mathcal{F}^k(q_1^k, \dots, q_t^k) \cdot \epsilon_i(v_{< i}^k) + \epsilon_t(v_{< t}^k) \leq \epsilon$ .

We conclude by applying this property at  $t = t_{\max}$  in Eq. 43:

$$\left| \ln \left( \frac{\Pr[V^k = v^k]}{\Pr[V^k = v^k]} \right) \right| \leq \epsilon$$

□

**Theorem A.3.** Suppose that the following two assumptions hold:

1. Querier  $k$ 's queries operate on a restricted data stream that only depends on its own past results. That is,  $D_t \leftarrow \mathcal{G}(V_{< t}^1, \dots, V_{< t}^K)$  is of the following form:

$$D_t^1 \leftarrow \mathcal{G}(V_{< t}^1), \dots, D_t^K \leftarrow \mathcal{G}(V_{< t}^K)$$

with  $q_t^k(D_{\leq t}) = q_t^k(D_{\leq t}^k)$ .

2. Shared limits do not filter  $k$ 's reports depending on other queriers' results:

$$\begin{aligned} & \forall k \in [K], \forall u, w \in \mathcal{R}^t : u^k = w^k, \\ & \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(u_{< t}^1), \dots, q_t^k(u_{< t}^k)) \\ &= \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(w_{< t}^1), \dots, q_t^k(w_{< t}^k)) \end{aligned}$$

Sufficient conditions for this property include queriers being non-adaptive, never reaching the shared limit, or operating on disjoint sets of devices (for on-device accounting with IDP filters).

Then condition Eq. 32 holds, and thus per-querier guarantees hold.

*Proof.* The first point implies:

$$q_t^k(v_{< t}^k)(D_{\leq t}(v_{< t})) = q_t^k(v_{< t}^k)(D_{\leq t}(v_{< t}^k))$$

Combining this with the second point, we directly get the assumption:

$$\forall k \in [K], \forall u, w \in \mathcal{R}^t : u^k = w^k, \quad (45)$$

$$\begin{aligned} & \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(u_{< t}^1), \dots, q_t^k(u_{< t}^k)) \cdot q_t^k(D_{\leq t}(u_{< t})) \\ &= \mathcal{L}(q_1^1, \dots, q_1^K, \dots, q_t^1(w_{< t}^1), \dots, q_t^k(w_{< t}^k)) \cdot q_t^k(D_{\leq t}(w_{< t})) \end{aligned}$$

□

## B Global IDP with Big Bird (§4 Proofs)

### B.1 Data and query model for Big Bird

This section sets up the formalism that underpins the main theorems, proved in subsequent sections and evoked in the body of the paper. We begin by defining Big Bird's data and query model, broadly adapted from Cookie Monster's formalism. We then provide useful properties that will be needed in subsequent sections. Indeed, while Cookie Monster analyzes global and individual sensitivities of queries at the device-epoch level, Big Bird additionally needs such analyses at the device-epoch-site level.

**Definition B.1** (User actions, sites and events). Let  $\mathcal{U}$  be the set of user-action identifiers,  $\mathcal{S}$  the set of site identifiers,  $\mathcal{I}$  the set of impression payloads and  $\mathcal{C}$  the set of conversion payloads.

An *event* is either an impression event  $(u, i, \text{imp})$  or a conversion event  $(u, c, \text{b}, \text{conv})$ :

- $u \in \mathcal{U}$  is the user action that caused the impression or conversion. The same user action can cause multiple events, and is denoted  $\text{uaCtx}$  in the body of the paper. On a fixed device, all accepted impression events carrying the same user-action identifier occur in a single epoch.
- $i \in \mathcal{S}$  (resp.  $c \in \mathcal{S}$ ) is the impression (resp. conversion) site where the event occurred.
- Conversions contain the set of queriers  $\mathbf{b} \subset \mathcal{S}$  that will receive the conversion report. ( $\mathbf{b}$  is a singleton unless we use the optimization from §D.)
- The payload  $\text{imp} \in \mathcal{I}$  or  $\text{conv} \in \mathcal{C}$  can be used to define concrete attribution functions.

We write sets of all impression events and all conversion events as follows:

$$\mathbb{F}^{\text{imp}} := \mathcal{U} \times \mathcal{S} \times \mathcal{I}, \quad \mathbb{F}^{\text{conv}} := \mathcal{U} \times \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{C},$$

and we write  $\mathbb{F} := \mathbb{F}^{\text{imp}} \cup \mathbb{F}^{\text{conv}}$  for the set of all events.

**Definition B.2** (Database of device-epoch records). Let  $\mathcal{D}$  be the set of device identifiers and  $\mathcal{E}$  the set of epoch identifiers. A database  $D$  is a set of device-epoch records where each record  $x = (d, e, F) \in \mathcal{X} = \mathcal{D} \times \mathcal{E} \times \mathcal{P}(\mathbb{F})$  contains a device  $d$ , an epoch  $e$  and a set of impression and conversion events  $F \subset \mathbb{F}$ . We write  $\mathbb{D}$  for the set of all databases where each  $(d, e)$  pair appears at most once.

**Definition B.3** (Conversion-site and impression-site event sets from events). For any event set  $F$ , define:

$$\begin{aligned}\text{ImpSites}(F) &:= \{i \in \mathcal{S} : \exists u \in \mathcal{U}, \text{imp} \in \mathcal{I}, (u, i, \text{imp}) \in F\}, \\ \text{ConvSites}(F) &:= \{c \in \mathcal{S} : \exists u \in \mathcal{U}, \mathbf{b} \subset \mathcal{S}, \text{conv} \in \mathcal{C}, \\ &\quad (u, c, \mathbf{b}, \text{conv}) \in F\}.\end{aligned}$$

Furthermore, for any user action  $u \in \mathcal{U}$ , define the user-action-scoped variants of these sets:

$$\begin{aligned}\text{ImpSites}_u(F) &:= \{i \in \mathcal{S} : \exists \text{imp} \in \mathcal{I}, (u, i, \text{imp}) \in F\}, \\ \text{ConvSites}_u(F) &:= \{c \in \mathcal{S} : \exists \mathbf{b} \subset \mathcal{S}, \text{conv} \in \mathcal{C}, \\ &\quad (u, c, \mathbf{b}, \text{conv}) \in F\},\end{aligned}$$

and let  $\text{AccessedSites}_u(F) := \text{ImpSites}_u(F) \cup \text{ConvSites}_u(F)$ . We also write  $\text{ImpActions}(F) := \{u \in \mathcal{U} : \exists i \in \mathcal{S}, \text{imp} \in \mathcal{I}, (u, i, \text{imp}) \in F\}$  for the user actions that saved the impressions of  $F$ .

We now define queries and reports, with a slight adaptation of Cookie Monster's definitions to our per-site semantics. While Cookie Monster comes with an arbitrary set of public events  $P$  for each querier  $b$ , here for simplicity we assume that all the conversions for a querier are public. Using Cookie Monster's terminology, that means we set  $P := \{(u, c, \mathbf{b}, \text{conv}) \in \mathbb{F}^{\text{conv}} : b \in \mathbf{b}\}$ . Also, while Cookie Monster defines a set of *relevant events*, potentially including conversions, in Def. B.5 we only consider *relevant impressions* for simplicity. In particular, this hardcodes "Case 1" from [43, Thm. 1].

**Definition B.4** (Neighboring databases).  $D \sim_x D'$  iff database  $D'$  is obtained from  $D$  by removing all impression events of a single device-epoch  $x$ . All conversion events are identical for  $D$  and  $D'$ .

**Definition B.5** (Attribution function, adapted from Cookie Monster). Fix a set of *relevant impressions*  $F_A \subset \mathbb{F}^{\text{imp}}$  such that there exists  $G_A \subset \mathcal{S} \times \mathcal{I}$  with  $F_A = \mathcal{U} \times G_A$  (i.e., user actions play no role in defining relevant impressions). Fix  $k, m \in \mathbb{N}^*$  where  $k$  is a number of epochs. An *attribution function* is a function  $A : \mathcal{P}(\mathbb{F})^k \rightarrow \mathbb{R}^m$  that takes  $k$  event sets  $F_1, \dots, F_k$  from  $k$  epochs and outputs an  $m$ -dimensional vector  $A(F_1, \dots, F_k)$ , such that only *relevant events* contribute to  $A$ . That is, for all  $(F_1, \dots, F_k) \in \mathcal{P}(\mathbb{F})^k$ , we have:

$$A(F_1, \dots, F_k) = A(F_1 \cap F_A, \dots, F_k \cap F_A). \quad (46)$$

In particular, conversion events never contribute to  $A$ .

**Definition B.6** (Report identifier and attribution report, same as Cookie Monster). Fix a domain of report identifiers  $\mathbb{Z}$ . Consider a mapping  $d(\cdot) : \mathbb{Z} \rightarrow \mathcal{D}$  from report identifiers to devices that gives the device  $d_r$  that generated a report  $r$ .

Given an attribution function  $A$ , a set of epochs  $E$  and a report identifier  $r \in \mathbb{Z}$ , the *attribution report*  $\rho_{r,A,E}$ , or  $\rho_r$  for short, is a function over the whole database  $D$  defined by:

$$\rho_r : D \in \mathbb{D} \mapsto A(D_{d_r}^E), \quad (47)$$

When several reports are in play, we write  $A_r$ ,  $E_r$  and  $F_{A_r}$  for the attribution function, the set of epochs and the set of relevant impressions attached to report  $r$ , so that  $\rho_r : D \mapsto A_r(D_{d_r}^{E_r})$ . The mechanism also stores, with each identifier, the conversion site  $c_r$  and the opaque action context  $u_r$  of the `measureConversion` request that created  $r$  (Alg. 7). Neither changes the function  $\rho_r$ ;  $u_r$  is only used for the per-action domain cap.

**Definition B.7** (Query, same as Cookie Monster). Consider a set of report identifiers  $R \subset \mathbb{Z}$ , and a set of attribution reports  $(\rho_r)_{r \in R}$  each with output in  $\mathbb{R}^m$ . The *query* for  $(\rho_r)_{r \in R}$  is the function  $Q : \mathbb{D} \rightarrow \mathbb{R}^m$  defined as  $Q(D) := \sum_{r \in R} \rho_r(D)$  for  $D \in \mathbb{D}$ .

**Definition B.8** (Device-epoch sensitivity, same as Cookie Monster). Fix a report  $\rho : \mathbb{D} \rightarrow \mathbb{R}^m$  for some  $m$ . We define the *individual  $L_1$  sensitivity of  $\rho$  for a device-epoch  $x \in \mathcal{X}$*  as follows:

$$\Delta_x(\rho) := \max_{D, D' \in \mathbb{D} : D' \sim_{x,D}} \|\rho(D) - \rho(D')\|_1. \quad (48)$$

We also define the *device-epoch global  $L_1$  sensitivity of  $\rho$*  as follows:

$$\Delta(\rho) := \max_{x \in \mathcal{X}} \Delta_x(\rho) \quad (49)$$

**Definition B.9** (Device-epoch-impression-site sensitivity). Fix a report  $\rho : \mathbb{D} \rightarrow \mathbb{R}^m$  for some  $m$ , and an impression site  $i \in \mathcal{S}$ . We define the *individual  $L_1$  sensitivity of  $\rho$  for a device-epoch-site  $x \in \mathcal{X}, i \in \mathcal{S}$*  as follows:

$$\Delta_{x,i}(\rho) := \max_{D, D' \in \mathbb{D} : D' \sim_{x,i,D}} \|\rho(D) - \rho(D')\|_1 \quad (50)$$

where  $D \sim_{x,i} D'$  iff database  $D'$  is obtained from  $D$  by removing all impression events on a single site  $i$  of a single device-epoch  $x$ . Equivalently, there exists  $D_0 \in \mathbb{D}$ , a record  $x \in \mathcal{X}$  such that  $\{D, D'\} = \{D_0 + x, D_0 + x^{i \rightarrow \emptyset}\}$ , where  $x^{i \rightarrow \emptyset} := (d, e, F^{i \rightarrow \emptyset})$  is the record obtained by removing all the impressions on  $i$  from  $x$ .

We also define the *device-epoch-site global  $L_1$  sensitivity of  $\rho$*  as follows:

$$\Delta_i(\rho) := \max_{x \in \mathcal{X}} \Delta_{x,i}(\rho) \quad (51)$$

To simplify subsequent results, we define some notation:

**Definition B.10** (Zeroing-out). Fix a vector of impression sets  $\mathbf{F} = (F_1, \dots, F_k) \in \mathcal{P}(\mathbb{F}^{\text{imp}})^k$  for any  $k > 0$ . For  $i \in \mathcal{S}$  and  $j \in [k]$  we define:

- $\mathbf{F}^{j \rightarrow \emptyset} := (F_1, \dots, F_{j-1}, \emptyset, F_{j+1}, \dots, F_k)$ , i.e., we zero-out the  $j$ th epoch.

- $\mathbf{F}^{j,i \rightarrow 0} := (F_1, \dots, F_{j-1}, F_j \setminus \mathcal{I}_i, F_{j+1}, \dots, F_k)$ , i.e., we zero-out all the impressions  $\mathcal{I}_i := \{(u, i, \text{imp}) : u \in \mathcal{U}, \text{imp} \in \mathcal{I}\}$  belonging to site  $i$  from the  $j$ th epoch.

Plugging Def. B.5 into Def. B.8 and Def. B.9 immediately gives:

**Lemma B.11** (Global sensitivity of reports). *Fix a device  $d$ , a set of  $k$  epochs  $E$ , an attribution function  $A$  and the corresponding report  $\rho : D \mapsto A(D_d^E)$ . We have:*

$$\Delta(\rho) = \max_{\mathbf{F} \in \mathcal{P}(\mathbb{F}^{\text{imp}})^k, j \in [k]} \|A(\mathbf{F}) - A(\mathbf{F}^{j \rightarrow 0})\|_1. \quad (52)$$

Moreover, for any impression site  $i \in \mathcal{S}$  we have:

$$\Delta_i(\rho) = \max_{\mathbf{F} \in \mathcal{P}(\mathbb{F}^{\text{imp}})^k, j \in [k]} \|A(\mathbf{F}) - A(\mathbf{F}^{j,i \rightarrow 0})\|_1. \quad (53)$$

**Theorem B.12** (Individual sensitivity of reports per epoch-site). *Fix a report identifier  $r$ , with device  $d_r$ , epochs  $E_r = \{e_1^{(r)}, \dots, e_k^{(r)}\}$ , attribution function  $A_r$  and relevant impressions  $F_{A_r}$ , and let  $\rho_r$  be the corresponding report from Eq. 47. Fix an impression site  $i \in \mathcal{S}$  and a device-epoch record  $x = (d, e, F) \in \mathcal{X}$ , and write  $F_i := F \cap \mathcal{I}_i$  for the impressions of  $x$  on site  $i$ . Finally, let*

$$I_r(x) := \text{ImpSites}(F \cap F_{A_r}) \quad (54)$$

*be the impression sites of  $x$  that can contribute to  $\rho_r$ . We can upper bound the individual sensitivity of  $\rho_r$  per epoch-site as follows:*

$$\Delta_{x,i}(\rho_r) \leq \begin{cases} 0 & \text{if } d \neq d_r, e \notin E_r, \\ & \text{or } i \notin I_r(x), \\ \|A_r(F_i) - A_r(\emptyset)\|_1 & \text{if } d = d_r, E_r = \{e\} \\ & \text{and } I_r(x) = \{i\}, \\ \Delta_i(\rho_r) & \text{if } d = d_r, e \in E_r, \\ & i \in I_r(x), \text{ and} \\ & |E_r| \geq 2 \text{ or } |I_r(x)| \geq 2. \end{cases} \quad (55)$$

*Proof.* By Def. B.9, each pair  $(D, D')$  in the maximum defining  $\Delta_{x,i}(\rho_r)$  satisfies  $\{D, D'\} = \{D_0 + x, D_0 + x^{i \rightarrow 0}\}$  for some  $D_0 \in \mathbb{D}$ , where  $x^{i \rightarrow 0} = (d, e, F \setminus \mathcal{I}_i)$ . Since  $D_0 + x \in \mathbb{D}$ , the database  $D_0$  has no record at  $(d, e)$ . W.l.o.g we assume  $D = D_0 + x$  and  $D' = D_0 + x^{i \rightarrow 0}$ . We now take the three cases in turn.

- *Case 1:  $d \neq d_r$ , or  $e \notin E_r$ , or  $i \notin I_r(x)$ .* If  $d \neq d_r$ , then  $D$  and  $D'$  only differ in a record of device  $d$ , so  $D_{d_r}^{E_r} = (D_0)_{d_r}^{E_r} = (D')_{d_r}^{E_r}$ . If  $e \notin E_r$ , then  $D$  and  $D'$  only differ in an epoch that  $\rho_r$  does not read, and again  $D_{d_r}^{E_r} = (D')_{d_r}^{E_r}$ . Finally, suppose  $i \notin I_r(x)$ , and assume  $d = d_r$  and  $e \in E_r$ . The records at  $(d, e)$  do differ, but not in any relevant impression:  $F \cap F_{A_r}$  contains no impression on site  $i$ , so

$$(F \setminus \mathcal{I}_i) \cap F_{A_r} = (F \cap F_{A_r}) \setminus \mathcal{I}_i = F \cap F_{A_r}. \quad (56)$$

Then Eq. 46 gives  $A_r(D_{d_r}^{E_r}) = A_r((D')_{d_r}^{E_r})$ .

- *Case 2:  $d = d_r$ ,  $E_r = \{e\}$  and  $I_r(x) = \{i\}$ .* Here  $k = 1$  and  $\rho_r(D) = A_r(D_{d_r}^e)$ . As  $D_0$  has no record at  $(d_r, e)$ , we get  $D_{d_r}^e = F$  and  $(D')_{d_r}^e = F \setminus \mathcal{I}_i$ . Since site  $i$  has all the relevant impressions of  $x$ , we have  $F \cap F_{A_r} \subseteq \mathcal{I}_i$ , and therefore

$$F \cap F_{A_r} = F_i \cap F_{A_r}, \quad (F \setminus \mathcal{I}_i) \cap F_{A_r} = \emptyset. \quad (57)$$

Eq. 46 then gives  $A_r(F) = A_r(F_i)$  and  $A_r(F \setminus \mathcal{I}_i) = A_r(\emptyset)$ , so every pair  $(D, D')$  yields the same value  $\|A_r(F_i) - A_r(\emptyset)\|_1$ .

- *Case 3:  $d = d_r$ ,  $e \in E_r$ ,  $i \in I_r(x)$ , and  $|E_r| \geq 2$  or  $|I_r(x)| \geq 2$ .* Let  $j \in [k]$  be the index of  $e$  in  $E_r$  and  $\mathbf{F} := D_{d_r}^{E_r}$ , so that  $(D')_{d_r}^{E_r} = \mathbf{F}^{j,i \rightarrow 0}$  and

$$\|\rho_r(D) - \rho_r(D')\|_1 = \|A_r(\mathbf{F}) - A_r(\mathbf{F}^{j,i \rightarrow 0})\|_1 \leq \Delta_i(\rho_r), \quad (58)$$

by Eq. 53. Unlike in the first two cases, this quantity depends on  $D_0$  (through the events that the other epochs of  $E_r$  contribute to  $\mathbf{F}$ ). Taking the maximum over  $D_0$  can only give back the global bound  $\Delta_i(\rho_r)$  in the worst case.  $\square$

Next, we show that  $\Delta_i(\rho) \leq 2\Delta(\rho)$  for any report. For reports using certain attribution functions, we can have a tighter  $\Delta_i(\rho) = \Delta(\rho)$  bound, but it does not hold in general.

**Lemma B.13** (Relation between Device-Epoch-Site and Device-Epoch Sensitivities). *For any report  $\rho$  with attribution function  $A$ , let  $F$  denote the full dataset,  $F_j$  denote all data in  $F$  pertaining to epoch  $j$ , and  $F_{j,i}$  denote all data in  $F_j$  pertaining to site  $i$ . The following inequality holds:*

$$\Delta_i(\rho) \leq 2\Delta(\rho) \quad (59)$$

*Proof.* Recall the definition of device-epoch-site global sensitivity:

$$\Delta_i(\rho) = \max_{\mathbf{F} \in \mathcal{P}(\mathbb{F}^{\text{imp}})^k, j \in [k]} \|A(\mathbf{F}^{j,i \rightarrow 0}) - A(\mathbf{F})\|_1. \quad (60)$$

We can decompose this expression using the triangle inequality:

$$\|A(\mathbf{F}^{j,i \rightarrow 0}) - A(\mathbf{F})\|_1 \quad (61)$$

$$= \|A(\mathbf{F}^{j,i \rightarrow 0}) - A(\mathbf{F}^{j \rightarrow 0}) + A(\mathbf{F}^{j \rightarrow 0}) - A(\mathbf{F})\|_1 \quad (62)$$

$$\leq \|A(\mathbf{F}^{j,i \rightarrow 0}) - A(\mathbf{F}^{j \rightarrow 0})\|_1 + \|A(\mathbf{F}^{j \rightarrow 0}) - A(\mathbf{F})\|_1 \quad (63)$$

For the first term, note that  $A(\mathbf{F}^{j,i \rightarrow 0})$  uses events  $F_{j,\neg i}$  from epoch  $j$  where  $F_{j,\neg i} = F_j \setminus F_{j,i}$ , while  $A(\mathbf{F}^{j \rightarrow 0})$  uses no events from epoch  $j$ . Since  $F_{j,\neg i} \subseteq F_j$  and all events from epoch  $j$  for a given device come from a single device-epoch record, we can view this as the change from adding a single record containing events  $F_{j,\neg i}$ . This is bounded by the definition of  $\Delta(\rho)$ :

$$\|A(\mathbf{F}^{j,i \rightarrow 0}) - A(\mathbf{F}^{j \rightarrow 0})\|_1 \leq \Delta(\rho). \quad (64)$$

The second term represents the sensitivity to removing

events from the entire epoch  $j$ . In the individual DP setting with device-epoch records, all events from epoch  $j$  for a given device come from a single device-epoch record. Removing this entire record corresponds exactly to one of the cases considered in the definition of  $\Delta(\rho)$ . Therefore:

$$\|A(F^{j \rightarrow \emptyset}) - A(F)\|_1 \leq \Delta(\rho). \quad (65)$$

Therefore, substituting the two upper bounds into Eq. (63):

$$\Delta_i(\rho) \leq \Delta(\rho) + \Delta(\rho) = 2\Delta(\rho). \quad (66)$$

□

## B.2 Comprehensive Big Bird Model

**Overview.** Big Bird manages per-site and global privacy filters using the quota mechanisms described in §4.3. Alg. 5 depicts the functionality triggered on receiving a report request (i.e., `measureConversion()` and `getReport()`). Big Bird checks and consumes budget from the relevant filters and prunes the resulting report based on filter status. First, Big Bird ensures all filters and quotas are initialized. Second, it computes the privacy losses incurred both at the epoch level [43, §C] and at the epoch-impression-site level (Def. B.14). Third, it checks whether all filters have sufficient budget and attempts to consume it. To avoid wasting budget from some filters (or quotas) when other filters are out of budget, Big Bird uses a transactional protocol to deduct privacy losses from multiple filters atomically. Alg. 6 formalizes this: if any filter cannot afford its share of the privacy loss, the report is zeroed out, and no budget is consumed from any filter for that report.

**Enforcing the per-action domain cap.** In addition to the filter-based budget checks, Big Bird enforces a per-action domain cap  $\kappa_{\text{action}}$  on the number of distinct sites that can access the Attribution API within a single user action context (Def. B.1). This mechanism limits adversarial creation of quota filters. As formalized in Alg. 5, the per-action domain cap check occurs *before* the atomic filter check: if the check fails, the epoch’s data is zeroed out ( $F_e \leftarrow \emptyset$ , using the zeroing-out notation from Appendix B.1), and no filters are accessed for that epoch.

**Subroutines.** Def. B.15, B.14, B.16 and Alg. 6 define subroutines used in Alg. 5. The first two definitions rely on the sensitivity bounds from Thm. B.12 (for the impression-site quota) and [43] (for the filters and the other quotas).

**Definition B.14** (EpochImpSiteBudget). Let  $x = (d, e, F) \in \mathcal{X}$  be a device-epoch record,  $i \in \mathcal{S}$  an impression site,  $\rho$  an attribution report, and  $\lambda > 0$  the Laplace noise scale applied to  $\rho$ . Given the upper bound  $\tilde{\Delta}_{x,i}(\rho) \geq \Delta_{x,i}(\rho)$  on the per-epoch-site individual sensitivity given by Thm. B.12, we can upper bound the epoch-site privacy loss consumed by  $\rho$  at  $(x, i)$  by

$$\text{EpochImpSiteBudget}(x, i, \rho, \lambda) := \frac{\tilde{\Delta}_{x,i}(\rho)}{\lambda}. \quad (67)$$

**Definition B.15** (EpochBudget, from [43]). Fix a device-epoch record  $x \in \mathcal{X}$ ,  $\rho$  an attribution report, and  $\lambda > 0$  the Laplace noise scale applied to  $\rho$ . Given the upper bound  $\tilde{\Delta}_x(\rho) \geq \Delta_x(\rho)$  on the per-epoch individual sensitivity given by [43], the individual privacy loss for device-epoch record  $x$  is:

$$\text{EpochBudget}(x, \rho, \lambda) := \frac{\tilde{\Delta}_x(\rho)}{\lambda} \quad (68)$$

**Definition B.16** (Privacy filter  $\mathcal{F}$ , adapted from [38]). Privacy filters track privacy loss of adaptively chosen mechanisms against a predetermined budget. We adapt the standard definition for pure DP privacy filters and define a filter  $\mathcal{F}$  as a stateful object storing its remaining capacity, initialized to  $\epsilon_{\text{initial}}$  (the filter capacity).

We then recursively define  $\mathcal{F}.\text{canConsume}$ ,  $\mathcal{F}.\text{tryConsume}$  and  $\text{pass}_{\mathcal{F},j}$ , the indicator that the  $j$ th budget was granted, for a sequence of adaptively chosen privacy budgets  $\epsilon^1, \epsilon^2, \dots$ , as follows:

- $\mathcal{F}.\text{canConsume}(\epsilon^j)$ : Returns TRUE if the filter can accommodate additional privacy loss  $\epsilon^j$ , i.e.,  $\epsilon^j \leq \epsilon_{\text{initial}} - \sum_{j' < j} \epsilon^{j'} \cdot \text{pass}_{\mathcal{F},j'}$ , where  $\epsilon_{\text{initial}}$  is the filter capacity.
- $\mathcal{F}.\text{tryConsume}(\epsilon^j)$ : Calls  $\text{canConsume}(\epsilon^j)$ . If successful, sets  $\text{pass}_{\mathcal{F},j} = 1$  to deduct  $\epsilon^j$  from the filter’s remaining capacity; otherwise, sets  $\text{pass}_{\mathcal{F},j} = 0$ .

By definition of  $\text{canConsume}$  we obtain the following *capacity invariant*:

$$\sum_j \epsilon^j \text{pass}_{\mathcal{F},j} \leq \epsilon_{\text{initial}}. \quad (69)$$

**Definition B.17** (Per-action domain cap  $\mathcal{K}$ ). We define a *domain cap* as a stateful object  $\mathcal{K} \subseteq \mathcal{S}$ , initialized to  $\emptyset$  with a cap  $\kappa_{\text{action}}$  and two methods, for a domain  $s \in \mathcal{S}$ :

- $\mathcal{K}.\text{canAdmit}(s) \iff |\mathcal{K} \cup \{s\}| \leq \kappa_{\text{action}}$ ,
- $\mathcal{K}.\text{tryAdmit}(s)$ : if  $\mathcal{K}.\text{canAdmit}(s)$ , set  $\mathcal{K} \leftarrow \mathcal{K} \cup \{s\}$ ; otherwise leave  $\mathcal{K}$  unchanged.

Note that only the first admission for a given domain can be rejected, and the invariant  $|\mathcal{K}| \leq \kappa_{\text{action}}$  bounds the number of distinct domains.

**Definition B.18** (Per-device-epoch state  $\mathcal{F}_{(d,e)}$ ). For each device-epoch record  $x = (d, e, F)$ , we maintain four families of pure DP filters (global, per querier  $b$ , per conversion site  $c$ , per impression site  $i$ ) and domain caps for each user action  $u$ . We denote the resulting state  $\mathcal{F}_{d,e}$ :

$$\mathcal{F}_{d,e} := \left( \mathcal{F}_{d,e}^{\text{global}}, (\mathcal{F}_{d,e}^{\text{querier}[b]})_{b \in \mathcal{S}}, (\mathcal{F}_{d,e}^{\text{conv-quota}[c]})_{c \in \mathcal{S}}, \right. \\ \left. (\mathcal{F}_{d,e}^{\text{imp-quota}[i]})_{i \in \mathcal{S}}, (\mathcal{K}_{d,e}[u])_{u \in \mathcal{U}} \right), \quad (70)$$

We lazily initialize these filters/cap instantiated with respective capacities  $\epsilon_{\text{global}}, \epsilon_{\text{querier}}, \epsilon_{\text{imp-quota}}, \epsilon_{\text{conv-quota}}, \kappa_{\text{action}}$ .

We make two remarks. First, the state  $\mathcal{F}_{d,e}$  can change depending on the adversary’s queries, even when the under-

**Alg. 5** Big Bird algorithm on a single device

---

```

1: Input
2: Device  $d$  with growing on-device database  $D_d \leftarrow \emptyset$ 
3: State  $(\mathcal{F}_{d,e})_{e \in E}$  lazily initialized with
   caps  $\epsilon_{\text{global}}, \epsilon_{\text{querier}}, \epsilon_{\text{imp-quota}}, \epsilon_{\text{conv-quota}}, \kappa_{\text{action}}$ 

4: def SaveImpression( $i, \text{imp}, u$ )
5:    $e \leftarrow \text{CurrentEpoch}()$ 
6:   if  $\mathcal{F}_{d,e}.\mathcal{K}[u].\text{canAdmit}(i) = \text{FALSE}$  then
7:     return // Silently reject impression if cap exceeded
8:    $\mathcal{F}_{d,e}.\mathcal{K}[u].\text{tryAdmit}(i)$ 
9:   // Store impression event
10:   $D_d^e \leftarrow D_d^e \cup \{(u, i, \text{imp})\}$ 
11:  return

// Generate report and update on-device budget
12: def GenerateReport( $D, \rho, \lambda, u, b$ )
13:  Read  $\rho$  to get device  $d$ , conversion site  $c$ , target epochs  $E$ , attribution
   function  $A$  with relevant impressions  $F_A$ .
14:  for  $e \in E$  do
15:     $x \leftarrow (d, e, D_d^e)$ 
16:    if  $\mathcal{F}_{d,e}.\mathcal{K}[u].\text{canAdmit}(c) = \text{FALSE}$  then // Def. B.18
17:       $F_e \leftarrow \emptyset$  // Zero out epoch if per-action domain cap exceeded
18:      continue // Skip to next epoch
19:     $\mathcal{F}_{d,e}.\mathcal{K}[u].\text{tryAdmit}(c)$ 
20:     $F_e \leftarrow D_d^e \cap F_A$  // Effective relevant impressions
21:     $i_e \leftarrow \text{ImpSites}(F_e)$  // Effective impression sites
22:     $\epsilon_x \leftarrow \text{EpochBudget}(x, \rho, \lambda)$  // Def. B.15
23:     $\epsilon_x^i \leftarrow \{\}$ 
24:    for  $i \in i_e$  do
25:       $\epsilon_x^i \leftarrow \text{EpochImpSiteBudget}(x, i, \rho, \lambda)$  // Def. B.14
26:       $\epsilon_x^i[i] \leftarrow \epsilon_x^i$ 

// Check and update filters/quotas atomically (Alg. 6)
27:  if  $\text{AtomicFilterCheckAndConsume}(\mathcal{F}_x, b, c, i_e, \epsilon_x, \epsilon_x^i) = \text{FALSE}$ 
   then
28:     $F_e \leftarrow \emptyset$  // Empty the epoch if any check fails
29:     $\hat{\rho} \leftarrow A((F_e)_{e \in E})$  // Clipped attribution report
30:  return  $\hat{\rho}$ 
31:

```

---

lying data is fixed (e.g., when the data is generated in chunks epoch by epoch). Second, since the pair  $(d, e)$  appears only once in valid databases, we can index by  $x = (d, e, F)$  to obtain the same  $\mathcal{F}_x$  even as  $F$  changes, e.g., when new events are added to the same device.

**B.3 Global Privacy Guarantees**

We now state and prove the privacy guarantees, stated for a formal privacy game modeling Alg. 5 execution over the whole system.

**Execution model.** Alg. 7 presents an abstract model of Big Bird’s operation, capturing how it answers queries sequentially for all queriers. At each time step  $t$ , querier  $k$  can formulate a query  $Q_t^k$ , which is a batch of reports as defined in Def. B.7. When a querier has nothing to ask, it can send an empty query with zero sensitivity. `AnswerQuery` then processes each individual report  $\rho_r$  (for  $r \in R$ , where  $R$  is the set of report identifiers in  $Q_t^k$ ). For each report, `GenerateReport` creates an individual clipped attribution report. Finally, re-

**Alg. 6** AtomicFilterCheckAndConsume

---

```

1: Input
2: Budgets  $\mathcal{F}_x$ , querier  $b$ , conversion site  $c$ , impression
   sites  $i$ 
3:  $\epsilon_x$ : epoch-level privacy loss
4:  $\epsilon_x^i$ : epoch-site-level privacy loss
5: // Phase 1: check if all filters can consume (Def. B.16)
6: if  $\mathcal{F}_x^{\text{querier}[b]}. \text{canConsume}(\epsilon_x) = \text{FALSE}$  then
7:   return FALSE
8: if  $\mathcal{F}_x^{\text{global}}. \text{canConsume}(\epsilon_x) = \text{FALSE}$  then
9:   return FALSE
10: if  $\mathcal{F}_x^{\text{conv-quota}[c]}. \text{canConsume}(\epsilon_x) = \text{FALSE}$  then
11:   return FALSE
12: for  $i \in i$  do
13:   if  $\mathcal{F}_x^{\text{imp-quota}[i]}. \text{canConsume}(\epsilon_x^i[i]) = \text{FALSE}$  then
14:     return FALSE
15: // Phase 2: consume from all filters (Def. B.16)
16:  $\mathcal{F}_x^{\text{querier}[b]}. \text{tryConsume}(\epsilon_x)$ 
17:  $\mathcal{F}_x^{\text{global}}. \text{tryConsume}(\epsilon_x)$ 
18:  $\mathcal{F}_x^{\text{conv-quota}[c]}. \text{tryConsume}(\epsilon_x)$ 
19: for  $i \in i$  do
20:    $\mathcal{F}_x^{\text{imp-quota}[i]}. \text{tryConsume}(\epsilon_x^i[i])$ 
21: return TRUE

```

---

ports are summed and noised before returning the query result.

At Line 13, world  $\chi = 0$  skips all the impressions on  $(d_0, e_0)$ , following the neighboring relation from Def. B.4, and every other call is executed identically in both worlds.

**Theorem 4.2.** Consider  $x \in \mathcal{X}$  with global filter capacity  $\epsilon_{\text{global}}$ . Then, Big Bird satisfies individual device-epoch  $\epsilon_{\text{global}}$ -DP for  $x$  with respect to the neighboring relation in Definition B.4. That is, for all data generation processes  $\mathcal{G}$ , for all queriers  $Q^1, \dots, Q^K$ , the output  $V^{(\chi)}$  of Alg. 7 under challenge bit  $\chi$  verifies:

$$\forall v \in \mathcal{R}^{\epsilon_{\text{max}} \times (t_{\text{max}}+1) \times K}, \left| \ln \left( \frac{\Pr[V^{(0)} = v]}{\Pr[V^{(1)} = v]} \right) \right| \leq \epsilon_{\text{global}}. \quad (71)$$

*Proof.* Take the in-or-out device-epoch  $x = (d_0, e_0, F) \in \mathcal{X}$ . Denote by  $x_C = (d_0, e_0, F \cap \mathbb{F}^{\text{conv}})$  the device-epoch obtained by keeping only public events from  $x$ , where public events are the set of all conversions. Take  $v \in \mathcal{R}^{\epsilon_{\text{max}} \times (t_{\text{max}}+1) \times K}$  and  $\chi \in \{0, 1\}$ .  $v$  is the vector of values returned at different points in Alg. 7. We split it into  $v = (v_{1,0}, v_{1,1}, \dots, v_{1,t_{\text{max}}}, \dots, v_{e,0}, \dots, v_{e,t})$  where  $v_{e,0} = (v_{e,0}^1, \dots, v_{e,0}^K)$  is a value for the output from Line 22 of Alg. 7 (public events from epoch  $e$  for each querier  $k \in [K]$ ), and  $v_{e,t} = (v_{e,t}^1, \dots, v_{e,t}^K)$ , where  $v_{e,t}^k$  is the output for the query  $Q_t^k$  for querier  $k$  at step  $t$  from epoch  $e$  (Line 27). By conditioning over past outputs at each time step  $(e, t, k) \in$

---

**Alg. 7** Formalism for DP analysis
 

---

```

1: Input
2: Challenge bit  $\chi \in \{0, 1\}$ 
3: In-or-out device-epoch  $(d_0, e_0)$ 
4: Data generation process  $\mathcal{G}$ 
5: Queriers  $Q^1, \dots, Q^K$ 
6: Output
7: View  $V = (V^1, \dots, V^K) \in \mathcal{R}^{e_{\max} \times (t_{\max} + 1) \times K}$  of all  $Q^k$ 
8: for  $e \in [e_{\max}]$  do
  // Devices and events for epoch  $e$ , generated from all previous results
9:  $G \leftarrow \mathcal{G}(V_{\leq e}^1, \dots, V_{\leq e}^K)$ 
  // View of each querier, initialized empty for conversions
10:  $(V_{e,0}^k)_{k \in [K]} = (\emptyset)_{k \in [K]}$ 
11: for  $(d, f) \in G$  do
12:   if  $f = (u, i, \text{imp}) \in \mathbb{F}^{\text{imp}}$  then
13:     if  $\chi = 0$  and  $d = d_0, e = e_0$  then
14:       skip // Keep impression events out in World 0
15:     else
16:       SaveImpression( $i, \text{imp}, u$ ) on  $d$ 
17:   else
  // Each querier receives its conversions and corresponding report identifiers
18:    $D_d^e \leftarrow D_d^e + f$ , with  $f = (u, c, \mathbf{b}, \text{conv})$ 
19:   Generate report identifier  $r \xleftarrow{\$} U(\mathcal{Z})$ 
  // Save the device and action context that generated  $r$ 
20:    $(d_r, u_r) \leftarrow (d, u)$ 
21:   for  $k \in \mathbf{b}$  do
22:      $V_{e,0}^k \leftarrow V_{e,0}^k \cup \{(r, f)\}$  // Output conversion and report identifier
23: // Queriers ask queries, round robin
24: for  $t \in [t_{\max}]$  do
25:   for  $k \in [K]$  do
    // Querier  $k$  chooses based on its own previous results
26:    $Q_t^k, \lambda_t^k \leftarrow Q^k(V_{\leq e, < t}^k)$ 
27:    $V_{e,t}^k \leftarrow \text{AnswerQuery}(D^{\leq e}, Q_t^k, \lambda_t^k, k)$ 
28: // Collect, aggregate and noise reports to answer  $Q$ 
29: def AnswerQuery( $D, Q, \lambda, \mathbf{b}$ )
30:    $(\rho_r)_{r \in R} \leftarrow Q$  // Get report identifiers from  $Q$ 
31:   for  $r \in R$  do
32:      $\hat{\rho}_r \leftarrow \text{GenerateReport}(D, \rho_r, \lambda, u_r, \mathbf{b})$ 
33:   Sample  $X \sim \text{Lap}(\lambda)$ 
34:   return  $\sum_{r \in R} \hat{\rho}_r + X$ 

```

---

$[e_{\max}] \times \{0, \dots, t_{\max}\} \times [K]$  we get:

$$\Pr[V^{(\chi)} = v] \quad (72)$$

$$= \prod_{e=1}^{e_{\max}} \prod_{t=0}^{t_{\max}} \prod_{k=1}^K \Pr[V_{e,t}^{k,(\chi)} = v_{e,t}^k | v_{\leq e, < t}^k]. \quad (73)$$

Take  $e \in [e_{\max}]$ ,  $t \in [t_{\max}]$  and  $k \in [K]$ . By Algorithm 7, we have:

$$\Pr[V_{e,t}^{k,(\chi)} = v_{e,t}^k | v_{\leq e, < t}^k] \quad (74)$$

$$= \Pr[\text{AnswerQuery}(D^{(\chi)}, Q_t^k, \lambda_t^k, k; \mathcal{F}_{e,t}^{k,(\chi)}) = v_{e,t}^k], \quad (75)$$

where the database  $D^{(\chi)}$ , the query  $Q_t^k$  with associated noise  $\lambda_t^k$  and the state  $\mathcal{F}_{e,t}^{k,(\chi)}$  are functions of past results  $v_{\leq e, < t}^k$ . The generated data and the queries are chosen by  $\mathcal{G}$  and  $Q^k$  conditionally on past results but independently of  $\chi$ . The database can depend on  $\chi$ , but only at  $(d_0, e_0)$ , since world 0 skips the impression saves on that device-epoch at Line 13.

The state can also depend on  $\chi$ , albeit in a limited fashion, since we have:

$$\forall x' \in D^{(0)} \cap D^{(1)}, (\mathcal{F}_{e,t}^{k,(0)})_{x'} = (\mathcal{F}_{e,t}^{k,(1)})_{x'} \quad (76)$$

Indeed, Alg. 5 shows that the budget spent for  $x'$  at each step only depends on past queries, which are identical in both worlds once we condition on  $v_{\leq e, < t}^k$ , and on the record  $x'$  itself, which is identical in both worlds. This is thanks to device-epoch level accounting in Def. B.15, B.14. The same holds for domain caps  $\mathcal{K}_{d',e'}[u]$ : even though GenerateReport iterates over the whole window  $E$ , each iteration reads and writes only  $\mathcal{K}_{d',e'}[u]$ , decides from that entry, and skips only that epoch if it hits the cap.

Let  $\rho_r(D; \mathcal{F}) = \text{GenerateReport}(D, \rho_r, \lambda; \mathcal{F})$  denote the filtered report returned by Alg. 5, which maintains the state  $\mathcal{F}$ . Then:

$$\begin{aligned} \Pr[V_{e,t}^{k,(\chi)} = v_{e,t}^k | v_{\leq e, < t}^k] \\ = \Pr \left[ \sum_{r \in R_{e,t}^k} \rho_r(D^{(\chi)}; \mathcal{F}_{e,t,r}^{(\chi)}) + X_{e,t} = v_{e,t}^k \right], \end{aligned} \quad (77)$$

where  $X_{e,t}$  is the Laplace noise added at time  $e, t$ .

We omit the  $e, t, k$  indices when they are clear from context. We denote by  $D$  the database generated conditionally on  $v_{\leq e, < t}^k$  before adding the in-or-out record, giving:  $D^{(0)} = D + x_C$  and  $D^{(1)} = D + x$  for  $e \geq e_0$  (and  $D^{(0)} = D^{(1)} = D$  for  $e < e_0$ ). By Def. B.5, since  $F_A \subseteq \mathbb{F}^{\text{imp}}$  and  $F \cap \mathbb{F}^{\text{conv}} \cap \mathbb{F}^{\text{imp}} = \emptyset$ , we have:

$$\begin{aligned} \rho_r(D^{(0)}; \mathcal{F}_r^{(0)}) &= \rho_r(D + (d_0, e_0, F \cap \mathbb{F}^{\text{conv}} \cap F_A)) \\ &= \rho_r(D; \mathcal{F}_r^{(0)}) \end{aligned} \quad (78)$$

By Alg. 5, remark that the state  $(\mathcal{F}_r^{(0)})_{x'}$  for  $x' \in \{x, x_C\}$  has no impact on the output: replacing the corresponding data by  $\emptyset$  at Line 17 (when hitting a per-action domain cap) or Line 28 (when running out of budget or quota) is a no-op. Combining this observation with Eq. 76 gives:

$$\rho_r(D^{(0)}; \mathcal{F}_r^{(0)}) = \rho_r(D; \mathcal{F}_r^{(1)}) \quad (79)$$

We can now bound the privacy loss at step  $e, t, k$ :

$$\left| \ln \left( \frac{\Pr[V_{e,t}^{k,(0)} = v_{e,t}^k | v_{\leq e, < t}^k]}{\Pr[V_{e,t}^{k,(1)} = v_{e,t}^k | v_{\leq e, < t}^k]} \right) \right| \quad (79)$$

$$\leq \left| \ln \left( \frac{\Pr \left[ \sum_{r \in R_{e,t}^k} \rho_r(D; \mathcal{F}_r^{(1)}) + X_{e,t} = v_{e,t}^k \right]}{\Pr \left[ \sum_{r \in R_{e,t}^k} \rho_r(D + x; \mathcal{F}_r^{(1)}) + X_{e,t} = v_{e,t}^k \right]} \right) \right| \quad (80)$$

We now omit  $\mathcal{F}_r^{(1)}$  since it is present on both sides. Take  $r \in R_t$ . We define the boolean  $\text{pass}_r$  by  $\text{pass}_r = 0$  if  $F_e$  is ze-

roed out (either by the domain cap at Line 17 or by the atomic check at Line 28), in the world where  $x$  is present. If  $\text{pass}_r = 0$ , we have  $\rho_r(D+x) = \rho_r(D)$  because of Alg. 5, Line 28. Otherwise, if  $\text{pass}_r = 1$ , we have  $\|\rho_r(D+x) - \rho_r(D)\|_1 \leq \Delta_x(\rho_r)$ .

Hence,

$$\|\rho_r(D) - \rho_r(D+x)\| \leq \Delta_x(\rho_r)\text{pass}_r \quad (81)$$

Thus by triangle inequality followed by Def. B.15 we have:

$$\left\| \sum_{r \in R_t} \rho_r(D) - \sum_{r \in R_t} \rho_r(D+x) \right\| \leq \sum_{r \in R_t} \Delta_x(\rho_r)\text{pass}_r \quad (82)$$

$$\leq \sum_{r \in R_t} \lambda_t^k \cdot \epsilon_r \text{pass}_r \quad (83)$$

And since  $X_t \sim \text{Lap}(\lambda_t^k)$ , by property of the Laplace distribution Eq. 80 becomes:

$$\left| \ln \left( \frac{\Pr[V_{e,t}^{k,(0)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]}{\Pr[V_{e,t}^{k,(1)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]} \right) \right| \leq \sum_{r \in R_t} \epsilon_r \text{pass}_r \quad (84)$$

We can now bound the total privacy loss. First, we use Eq. 72 and the fact that the outputs  $v_{e,0}^k$  at Line 22 are identical across both worlds by definition of  $x_C$  to drop indices where  $t = 0$ :

$$\left| \ln \left( \frac{\Pr[V^{(0)} = v]}{\Pr[V^{(1)} = v]} \right) \right| \quad (85)$$

$$= \left| \ln \left( \prod_{e=1}^{e_{\max}} \prod_{t=0}^{t_{\max}} \prod_{k=1}^K \frac{\Pr[V_{e,t}^{k,(0)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]}{\Pr[V_{e,t}^{k,(1)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]} \right) \right| \quad (86)$$

$$= \left| \sum_{e=1}^{e_{\max}} \sum_{t=1}^{t_{\max}} \sum_{k=1}^K \ln \left( \frac{\Pr[V_{e,t}^{k,(0)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]}{\Pr[V_{e,t}^{k,(1)} = v_{e,t}^k | v_{\leq e, < t}^{< k}]} \right) \right| \quad (87)$$

By triangle inequality and Eq. 84, we get:

$$\left| \ln \left( \frac{\Pr[V^{(0)} = v]}{\Pr[V^{(1)} = v]} \right) \right| \quad (88)$$

$$\leq \sum_{e=1}^{e_{\max}} \sum_{t=1}^{t_{\max}} \sum_{k=1}^K \sum_{r \in R_t} \epsilon_r \text{pass}_r \quad (89)$$

Finally, each  $\epsilon_r \text{pass}_r$  is an amount granted by  $\mathcal{F}_{d_0, e_0}^{\text{global}}$ , so its capacity invariant Eq. 69 gives:

$$\sum_{e=1}^{e_{\max}} \sum_{t=1}^{t_{\max}} \sum_{k=1}^K \sum_{r \in R_t} \epsilon_r \text{pass}_r \leq \epsilon_{\text{global}} \quad (90)$$

Hence we have shown Eq. 71.  $\square$

## B.4 DoS Resilience

This section proves our main resilience result for Big Bird's quota-based online algorithm: Thm. 4.1. First, Lem. B.19 shows that the 2-PC check (Alg. 6) ensures (1) atomic consumption across all filters relevant to a query, and (2) when all filters have sufficient budget, each consumes an amount proportional to its level-specific sensitivity either at epoch or at epoch-site level. Next, Lem. B.21 and B.23 bound the

total privacy budget the adversary can consume from the global budget at any device-epoch, using the atomic consumption guarantees of Lem. B.19. These first two lemmas give Thm. 4.1(i). Lem. B.24 then connects user actions to obtain Thm. 4.1(ii), and Lem. B.25 lifts the bound to the query level for Thm. 4.1(iii).

### B.4.1 Preliminaries.

**Notation.** Everything in this section (§B.4) is stated inside one fixed execution of Alg. 7. Let  $s_{\max} := e_{\max} t_{\max} K$ . We index the queries from Alg. 7 by  $s \in [s_{\max}]$  instead of triples  $(e, t, k)$ , so each execution of AnswerQuery is associated with a unique  $s$ , with corresponding report identifiers  $R_s$  and querier  $b_s$ .

For a device-epoch  $x = (d, e, F)$ , a query  $s$  and  $r \in R_s$  we write  $c_r$  for the conversion site of  $r$ ,  $\mathbf{i}_{r,e} = \text{ImpSites}(F \cap F_{A_r})$  for its effective impression sites (i.e.,  $I_r(x)$  from Thm. B.12),  $\epsilon_{r,s}$  and  $\epsilon_{r,s}^i$  (for  $i \in \mathbf{i}_{r,e}$ ) for the losses computed at Line 22 and Line 25, and  $\text{pass}_{r,s} \in \{0, 1\}$  for the indicator that GenerateReport zeroes  $F_e$  at neither Line 17 nor Line 28. Each of  $\epsilon_{r,s}$  and  $\epsilon_{r,s}^i$  is 0 when Alg. 5 does not compute it at  $x$ . When the report index is clear from context, we drop  $r$  and write  $c_s$ ,  $\mathbf{i}_s$ ,  $\epsilon_s^i$  and  $\text{pass}_s$ .

**Lemma B.19** (Atomic filter check guarantees). *Fix a device-epoch record  $x = (d, e, F)$ , a query  $s$  and  $r \in R_s$ , with querier  $b = b_s$ . Using the definition of  $\text{pass}_{r,s}$  and Alg. 6, we have:*

- (a)  $\mathcal{F}_{d,e}^{\text{querier}[b]}$ ,  $\mathcal{F}_{d,e}^{\text{global}}$  and  $\mathcal{F}_{d,e}^{\text{conv-quota}[c_r]}$  each grant exactly  $\epsilon_{r,s} \text{pass}_{r,s}$ ;
- (b) for each  $i \in \mathbf{i}_{r,e}$ ,  $\mathcal{F}_{d,e}^{\text{imp-quota}[i]}$  grants exactly  $\epsilon_{r,s}^i \text{pass}_{r,s}$ .

*Proof.* We prove both properties together. Fix  $r \in R_s$  with  $\text{pass}_{r,s} = 1$ . By Alg. 6, the atomic check returns TRUE exactly when all Phase 1 checks pass and all Phase 2 operations execute. For this report, with conversion site  $c_r$ , querier  $b_s$ , and impression sites  $\mathbf{i}_{r,e}$ , Phase 2 calls:

- $\mathcal{F}_x^{\text{querier}[b_s]}$ . tryConsume( $\epsilon_{r,s}$ )
- $\mathcal{F}_x^{\text{global}}$ . tryConsume( $\epsilon_{r,s}$ )
- $\mathcal{F}_x^{\text{conv-quota}[c_r]}$ . tryConsume( $\epsilon_{r,s}$ )
- For each  $i \in \mathbf{i}_{r,e}$ :  $\mathcal{F}_x^{\text{imp-quota}[i]}$ . tryConsume( $\epsilon_{r,s}^i$ )

Note that  $\epsilon_{r,s}$  is the device-epoch-level individual privacy loss computed for this specific report  $r$  (e.g., via EpochBudget in Line 22 of Alg. 5). Similarly, each  $\epsilon_{r,s}^i$  is the device-epoch-site-level individual privacy loss for impression site  $i$  relevant to this report  $r$  (via EpochImpSiteBudget in Alg. 5). Therefore, when  $\text{pass}_{r,s} = 1$ , the conversion-site quota filter (for  $c_r$ ), the per-site filter (for  $b_s$ ), and the global budget all consume exactly the same amount  $\epsilon_{r,s}$ . Concurrently, each relevant impression-site quota filter (for  $i \in \mathbf{i}_{r,e}$ ) consumes its specific amount  $\epsilon_{r,s}^i$ , which is proportional to its sensitivity at the epoch-impression-site level for this report  $r$ .  $\square$

**B.4.2 Number of sites.** With these atomic guarantees we can bound how much of the global budget can be depleted

at any device-epoch  $x \in \mathcal{X}$ . The main isolation result is in Thm. 4.1, which uses Lem. B.21, B.22 and B.23.

**Definition B.20** (Budget consumed and quotas charged). Fix a device-epoch  $x = (d, e, F)$ , and any set of reports  $\mathcal{R}_s \subseteq R_s$  for each query  $s \in [s_{\max}]$ .

Let  $\mathcal{R}^+ := \{(s, r) : s \in [s_{\max}], r \in \mathcal{R}_s, \text{pass}_{r,s} = 1\}$  the reports that pass the filters. Then, we define:

$$\epsilon_{\text{global}}(\mathcal{R}) := \sum_{(s,r) \in \mathcal{R}^+} \epsilon_{r,s}, \quad (91)$$

$$C(\mathcal{R}) := \bigcup_{(s,r) \in \mathcal{R}^+} \{c_r\},$$

$$N(\mathcal{R}) := |C(\mathcal{R})|,$$

$$I(\mathcal{R}) := \bigcup_{(s,r) \in \mathcal{R}^+} \mathbf{i}_{r,e},$$

$$M(\mathcal{R}) := |I(\mathcal{R})|. \quad (92)$$

Def. B.20 defines what a group of report consumes. When reports are associated with an attacker, this defines the impact of the attacker, but the definition itself does not involved any notion of attacker. By Lem. B.19, reports  $\mathcal{R}$  consume  $\epsilon_{\text{global}}(\mathcal{R})$  from  $\mathcal{F}_{d,e}^{\text{global}}$ , and they charge the domains  $C(\mathcal{R})$  and  $I(\mathcal{R})$ . In the lemmas below, we take  $\mathcal{R}$  to be the reports an adversary elicits and note:  $\mathcal{R}_{\text{adv}} = (R_s^{\text{adv}})_{s \in [s_{\max}]}$  with  $R_s^{\text{adv}} \subseteq R_s$ , writing  $\epsilon_{\text{global}}^{\text{adv}} := \epsilon_{\text{global}}(\mathcal{R}_{\text{adv}})$  and likewise  $\mathcal{R}_{\text{adv}}^+$ ,  $C_{\text{adv}}$ ,  $N_{\text{adv}}$ ,  $I_{\text{adv}}$ ,  $M_{\text{adv}}$ . Note that  $I_{\text{adv}}$  contains every impression site charged by an attacker-elicited report, even honest impression sites that happen to allow malicious conversion sites to query their impressions. This corresponds to the threat model from §4.1, where  $I_{\text{adv}} \cup C_{\text{adv}}$  is the set of domains directly controlled by, or transitively partnering with the attacker.

**Lemma B.21** (Conversion-site quotas bound). Fix a device-epoch  $x = (d, e, F)$  and an adversary that elicits the reports  $\mathcal{R}_{\text{adv}}$ . By Def. B.20, these charge  $N_{\text{adv}}$  distinct conv-quota filters and consume  $\epsilon_{\text{global}}^{\text{adv}}$  from  $\mathcal{F}_{d,e}^{\text{global}}$ .

Then, we have:

$$\epsilon_{\text{global}}^{\text{adv}} \leq N_{\text{adv}} \epsilon_{\text{conv-quota}}. \quad (93)$$

*Proof.* By definition of  $\epsilon_{\text{global}}^{\text{adv}}$  we have:

$$\begin{aligned} \epsilon_{\text{global}}^{\text{adv}} &= \sum_{s \in [s_{\max}]} \sum_{r \in \mathcal{R}_s : \text{pass}_{r,s}=1} \epsilon_{r,s} \\ &= \sum_{s \in [s_{\max}]} \sum_{r \in \mathcal{R}_s} \epsilon_{r,s} \text{pass}_{r,s} \end{aligned}$$

Then we use the definition of  $C_{\text{adv}}$  to sum over all reports  $R_s$  instead of  $\mathcal{R}_s \subset R_s$ :

$$\begin{aligned} \epsilon_{\text{global}}^{\text{adv}} &\leq \sum_{c \in C_{\text{adv}}} \sum_{s \in [s_{\max}]} \sum_{\substack{r \in R_s \\ c_r = c}} \epsilon_{r,s} \text{pass}_{r,s} \\ &\leq \sum_{c \in C_{\text{adv}}} \epsilon_{\text{conv-quota}} \end{aligned}$$

where the last step comes from Lem. B.19(a): each  $r$  grants  $\epsilon_{r,s} \text{pass}_{r,s}$  to  $\mathcal{F}_{d,e}^{\text{global}}$  and the same amount to  $\mathcal{F}_{d,e}^{\text{conv-quota}[c_r]}$ , followed by the capacity invariant Eq. 69.

The definition of  $N_{\text{adv}}$  finally gives  $\epsilon_{\text{global}}^{\text{adv}} \leq N_{\text{adv}} \epsilon_{\text{conv-quota}}$ .  $\square$

**Lemma B.22** (Impression-site quota allocation consistency). Fix a device-epoch  $x = (d, e, F)$ , a query  $s$  and a report  $r \in R_s$ , and take  $\tilde{\Delta}_{x,i}(\rho_r)$  as in Eq. 55 with  $2\Delta(\rho_r)$  as an upper bound for  $\Delta_i(\rho_r)$  (Lem. B.13). We have:

$$\epsilon_{r,s} \leq \sum_{i \in \mathbf{i}_{r,e}} \epsilon_{r,s}^i \quad (94)$$

*Proof.* Write  $\lambda$  for the noise scale of query  $s$ , so that  $\epsilon_{r,s} = \tilde{\Delta}_x(\rho_r)/\lambda$  and  $\epsilon_{r,s}^i = \tilde{\Delta}_{x,i}(\rho_r)/\lambda$  (Def. B.15, B.14). We go through the different cases for the upper bound on  $\Delta_x(\rho_r)$ :

- If  $d \neq d_r$ ,  $e \notin E_r$ , or  $F \cap F_{A_r} = \emptyset$ , then  $\epsilon_{r,s} = 0$ . In that case, by Thm. B.12 we also have  $\epsilon_{r,s}^i = 0$  for all  $i$ . This covers in particular  $\mathbf{i}_{r,e} = \emptyset$ , since  $\mathbf{i}_{r,e} = \text{ImpSites}(F \cap F_{A_r})$ .
  - Otherwise  $\mathbf{i}_{r,e} \neq \emptyset$ , and we fix any  $i \in \mathbf{i}_{r,e}$ :
    - If  $d = d_r$  and  $E_r = \{e\}$ :
      - \* If  $\mathbf{i}_{r,e} = \{i\}$ , we have  $\epsilon_{r,s} = \epsilon_{r,s}^i = \|A_r(F) - A_r(\emptyset)\|_1/\lambda$ .
      - \* Else, we have  $\epsilon_{r,s}^i = 2\Delta(\rho_r)/\lambda \geq \|A_r(F) - A_r(\emptyset)\|_1/\lambda = \epsilon_{r,s}$ .
    - Else, we have  $\epsilon_{r,s}^i = 2\Delta(\rho_r)/\lambda \geq \Delta(\rho_r)/\lambda \geq \epsilon_{r,s}$ .
- In each case  $\epsilon_{r,s} \leq \epsilon_{r,s}^i$ , and the remaining terms of the sum are non-negative.  $\square$

Note that the upper bound in Lem. B.22 can be quite loose. Since the impression-site quota has no privacy meaning and is only used through Lem. B.22 to obtain isolation guarantees, instead of using per-epoch-site privacy loss we could use any heuristic that also satisfies Lem. B.22's  $\epsilon_{r,s} \leq \sum_{i \in \mathbf{i}_{r,e}} \epsilon_{r,s}^i$ . For instance, we could divide  $\epsilon_{r,s}$  uniformly across  $\mathbf{i}_{r,e}$ .

**Lemma B.23** (Impression-site quotas bound). Fix a device-epoch  $x = (d, e, F)$  and an adversary that elicits the reports  $\mathcal{R}_{\text{adv}}$ . By Def. B.20, these charge  $M_{\text{adv}}$  distinct imp-quota filters and consume  $\epsilon_{\text{global}}^{\text{adv}}$  from  $\mathcal{F}_{d,e}^{\text{global}}$ .

Then, we have:

$$\epsilon_{\text{global}}^{\text{adv}} \leq M_{\text{adv}} \epsilon_{\text{imp-quota}}. \quad (95)$$

*Proof.* By definition of  $\epsilon_{\text{global}}^{\text{adv}}$ , and then by Lem. B.22 applied to each report, we have:

$$\begin{aligned}\epsilon_{\text{global}}^{\text{adv}} &= \sum_{s \in [s_{\max}]} \sum_{r \in \mathcal{R}_s} \epsilon_{r,s} \text{pass}_{r,s} \\ &\leq \sum_{s \in [s_{\max}]} \sum_{r \in \mathcal{R}_s} \sum_{i \in \mathbf{i}_{r,e}} \epsilon_{r,s}^i \text{pass}_{r,s}\end{aligned}$$

Then we use the definition of  $I_{\text{adv}}$ , which contains  $\mathbf{i}_{r,e}$  for every report with  $\text{pass}_{r,s} = 1$ , to sum over all reports  $\mathcal{R}_s$  instead of  $\mathcal{R}_s \subset \mathcal{R}_s$ :

$$\begin{aligned}\epsilon_{\text{global}}^{\text{adv}} &\leq \sum_{i \in I_{\text{adv}}} \sum_{s \in [s_{\max}]} \sum_{r \in \mathcal{R}_s} \epsilon_{r,s}^i \text{pass}_{r,s} \\ &\leq \sum_{i \in I_{\text{adv}}} \epsilon_{\text{imp-quota}}\end{aligned}$$

where the last step comes from Lem. B.19(b): each  $r$  grants  $\epsilon_{r,s}^i \text{pass}_{r,s}$  to  $\mathcal{F}_{d,e}^{\text{imp-quota}[i]}$  for each  $i \in \mathbf{i}_{r,e}$ , followed by the capacity invariant Eq. 69.

The definition of  $M^{\text{adv}}$  finally gives  $\epsilon_{\text{global}}^{\text{adv}} \leq M^{\text{adv}} \epsilon_{\text{imp-quota}}$ .  $\square$

We can now combine Lem. B.21 and B.23 to obtain our main isolation theorem:

**Theorem 4.1(i) (Resilience to DoS depletion).** *Fix a device-epoch  $x = (d, e, F)$  and an adversary that elicits the reports  $\mathcal{R}_{\text{adv}}$ . By Def. B.20, these charge  $M^{\text{adv}}$  distinct imp-quota filters and  $N^{\text{adv}}$  distinct conv-quota filters.*

*Then, the adversary's consumption  $\epsilon_{\text{global}}^{\text{adv}}$  from  $\mathcal{F}_{d,e}^{\text{global}}$  satisfies:*

$$\epsilon_{\text{global}}^{\text{adv}} \leq \min(M^{\text{adv}} \epsilon_{\text{imp-quota}}, N^{\text{adv}} \epsilon_{\text{conv-quota}}). \quad (96)$$

*Proof.* The conversion-side bound is Lem. B.21 and the impression-side bound is Lem. B.23, and we take their minimum.  $\square$

**B.4.3 Number of user actions.** We now bound the number of sites by the user actions that resulted in quota deductions.

**Lemma B.24** (Quotas per user action). *Fix a device-epoch  $x = (d, e, F)$  and an adversary that elicits the reports  $\mathcal{R}_{\text{adv}}$ . Separate the actions that carry their conversions from those that saved the impressions they read at  $x$ :*

$$\begin{aligned}\mathcal{U}_{\text{conv}}^{\text{adv}} &:= \{u_r : (s, r) \in \mathcal{R}_{\text{adv}}^+\}, \\ \mathcal{U}_{\text{imp}}^{\text{adv}} &:= \bigcup_{(s,r) \in \mathcal{R}_{\text{adv}}^+} \text{ImpActions}(F \cap F_{A_r}).\end{aligned}$$

Write  $U_{\text{conv}}^{\text{adv}} := |\mathcal{U}_{\text{conv}}^{\text{adv}}|$ ,  $U_{\text{imp}}^{\text{adv}} := |\mathcal{U}_{\text{imp}}^{\text{adv}}|$ , and  $U^{\text{adv}} := |\mathcal{U}_{\text{conv}}^{\text{adv}} \cup \mathcal{U}_{\text{imp}}^{\text{adv}}|$ . Then,

$$N^{\text{adv}} \leq \kappa_{\text{action}} U_{\text{conv}}^{\text{adv}}, \quad (97)$$

$$M^{\text{adv}} \leq \kappa_{\text{action}} U_{\text{imp}}^{\text{adv}}, \quad (98)$$

$$N^{\text{adv}}, M^{\text{adv}} \leq \kappa_{\text{action}} U^{\text{adv}}. \quad (99)$$

*Proof.* For any  $c \in C_{\text{adv}}$ , some  $(s, r) \in \mathcal{R}_{\text{adv}}^+$  has  $c_r = c$ , so the `GenerateReport` path for that `measureConversion` request passed  $\mathcal{F}_{d,e} \cdot \mathcal{K}[u_r].\text{canAdmit}(c)$ . Hence  $c \in \mathcal{K}_{d,e}[u_r]$  with  $u_r \in \mathcal{U}_{\text{conv}}^{\text{adv}}$ .

Similarly, for any  $i \in I_{\text{adv}}$ , there is  $r$  such that  $i \in \mathbf{i}_{r,e} = \text{ImpSites}(F \cap F_{A_r})$ . Hence  $F \cap F_{A_r}$  contains an impression  $(u, i, \text{imp})$  with  $u \in \text{ImpActions}(F \cap F_{A_r}) \subseteq \mathcal{U}_{\text{imp}}^{\text{adv}}$ . That event was stored by an accepted `SaveImpression`, which passed  $\mathcal{F}_{d,e} \cdot \mathcal{K}[u].\text{canAdmit}(i)$ , so  $i \in \mathcal{K}_{d,e}[u]$ .

Finally, each cap contains at most  $\kappa_{\text{action}}$  domains (Def. B.17). Hence the first bound (Eq. 97) comes from  $C_{\text{adv}}$  with the caps indexed by  $\mathcal{U}_{\text{conv}}^{\text{adv}}$ , and the second bound (Eq. 98) comes from  $I_{\text{adv}}$  with the caps indexed by  $\mathcal{U}_{\text{imp}}^{\text{adv}}$ .

The third bound (Eq. 99) takes the union.  $\square$

We now prove the main graceful degradation theorem, which bounds the adversary's budget consumption as a function of the number of user actions they can elicit.

**Theorem 4.1(ii) (Graceful degradation).** *Consider an attacker collecting  $U^{\text{adv}}$  user actions across their domains for device  $d$ . Under the configuration of Table 2, the budget  $\epsilon_{\text{global}}^{\text{adv}}$  that this attacker can consume from the global budget is upper-bounded by:*

$$\epsilon_{\text{global}}^{\text{adv}} \leq (1 + r) \epsilon_{\text{querier}} \times \kappa_{\text{action}} \times U^{\text{adv}}. \quad (100)$$

*Proof.* By Lem. B.24, and using the configuration from Table 2, where  $\epsilon_{\text{imp-quota}} = n(1 + r) \epsilon_{\text{querier}}$  and  $\epsilon_{\text{conv-quota}} = (1 + r) \epsilon_{\text{querier}}$ ,

$$M^{\text{adv}} \epsilon_{\text{imp-quota}} \leq n(1 + r) \epsilon_{\text{querier}} \times \kappa_{\text{action}} \times U^{\text{adv}}$$

$$N^{\text{adv}} \epsilon_{\text{conv-quota}} \leq (1 + r) \epsilon_{\text{querier}} \times \kappa_{\text{action}} \times U^{\text{adv}}.$$

The limiting factor is  $N^{\text{adv}} \epsilon_{\text{conv-quota}}$ , and applying part (i) of Thm. 4.1 yields the result.  $\square$

Note that under the additional assumption that no domain both saves an impression and measures a conversion under the same user action, each charged quota filter would occupy a distinct domain cap slot, giving  $M^{\text{adv}} + N^{\text{adv}} \leq \kappa_{\text{action}} U^{\text{adv}}$ . Part (i) of Thm. 4.1 would then give  $\epsilon_{\text{global}}^{\text{adv}} \leq (1 + r) \epsilon_{\text{querier}} \min(nM^{\text{adv}}, N^{\text{adv}}) \leq \frac{n}{n+1} (1 + r) \epsilon_{\text{querier}} \kappa_{\text{action}} U^{\text{adv}}$ . We do not make this assumption, and Thm. 4.1(ii) uses the weaker bound above.

**B.4.4 Number of devices.**

**Lemma B.25** (Query error from depleted device-epochs). *Take a database  $D$  with filter state  $\mathcal{F}$ , and let  $\tilde{\mathcal{F}}$  be obtained from  $\mathcal{F}$  by depleting  $\epsilon_{\text{global}}$  at  $k$  distinct device-epochs. Let  $Q = \sum_{r \in R} \rho_r$  be a query in which  $\Delta(\rho_r) \leq \Delta$  for every  $r$ , and at most  $m$  reports belong to any one device. Then*

$$\|Q(D; \mathcal{F}) - Q(D; \tilde{\mathcal{F}})\|_1 \leq m \cdot k \cdot \Delta.$$

*Proof.* Take a query  $Q : D \mapsto \sum_{r \in R} \rho_r(D)$  as defined in Def. B.7. Take a database  $D$  and filter states  $\mathcal{F}, \tilde{\mathcal{F}}$  where  $\tilde{\mathcal{F}}$

is obtained from  $\mathcal{F}$  by depleting  $\epsilon_{\text{global}}$  for  $k$  distinct device-epochs. For a report  $\rho_r$ , we denote by  $\rho_r(D; \mathcal{F})$  the filtered report returned by `GenerateReport` in Alg. 5. Similarly, we denote by  $Q(D; \mathcal{F}) := \sum_{r \in R} \rho_r(D; \mathcal{F})$  the filtered query before adding noise.

Consider a single report identifier  $r \in R$ , with attribution function  $A_r$  over  $n_r := |E_r|$  epochs. We have  $\rho_r(D; \mathcal{F}) = A_r(F_1, \dots, F_{n_r})$  where  $F_1, \dots, F_{n_r}$  are the relevant events from  $D$  after passing the filter check based on  $\mathcal{F}$  in Alg. 5. Now, without loss of generality, suppose that the first  $k_r$  epochs for that report are depleted in  $\tilde{\mathcal{F}}$ . This causes the filter check to fail for these epochs, returning:

$$\rho_r(D; \tilde{\mathcal{F}}) = A_r(\emptyset, \dots, \emptyset, F_{k_r+1}, \dots, F_{n_r}).$$

For  $j \in \{0, \dots, k_r\}$ , write  $\mathbf{F}^{(j)} := (\emptyset, \dots, \emptyset, F_{j+1}, \dots, F_{n_r})$  for the input with the first  $j$  epochs zeroed out, so that  $\rho_r(D; \mathcal{F}) = A_r(\mathbf{F}^{(0)})$  and  $\rho_r(D; \tilde{\mathcal{F}}) = A_r(\mathbf{F}^{(k_r)})$ . Consecutive inputs  $\mathbf{F}^{(j-1)}$  and  $\mathbf{F}^{(j)}$  differ in a single epoch, so by definition of global sensitivity (Def. B.8) each step is bounded by  $\Delta(\rho_r) \leq \Delta$ . After telescoping over the  $k_r$  depleted epochs and applying the triangle inequality we get:

$$\begin{aligned} \|\rho_r(D; \mathcal{F}) - \rho_r(D; \tilde{\mathcal{F}})\|_1 &= \left\| \sum_{j=1}^{k_r} A_r(\mathbf{F}^{(j-1)}) - A_r(\mathbf{F}^{(j)}) \right\|_1 \\ &\leq \sum_{j=1}^{k_r} \|A_r(\mathbf{F}^{(j-1)}) - A_r(\mathbf{F}^{(j)})\|_1 \\ &\leq k_r \cdot \Delta \end{aligned}$$

Putting this together, we get:

$$\begin{aligned} \|Q(D; \mathcal{F}) - Q(D; \tilde{\mathcal{F}})\|_1 &= \left\| \sum_{r \in R} \rho_r(D; \mathcal{F}) - \rho_r(D; \tilde{\mathcal{F}}) \right\|_1 \\ &\leq \sum_{r \in R} \|\rho_r(D; \mathcal{F}) - \rho_r(D; \tilde{\mathcal{F}})\|_1 \\ &\leq \sum_{r \in R} k_r \cdot \Delta \\ &\leq m \cdot k \cdot \Delta \end{aligned}$$

where  $\sum_{r \in R} k_r \leq m \cdot k$ : each report reads depleted epochs only on its own device, so each of the  $k$  depleted device-epochs is counted by at most the  $m$  reports belonging to that device.  $\square$

**Theorem 4.1(iii) (Bounded query impact).** *Consider the setting of Lemma B.25 with an attacker collecting  $U^{\text{adv}}$  user actions across their domains for device  $d$ , whose own reports deplete the global budget at each of the  $k$  device-epochs. Under the configuration of Table 2, the error induced by the attacker is bounded by:*

$$\|Q(D; \mathcal{F}) - Q(D; \tilde{\mathcal{F}})\|_1 \leq m \cdot k \cdot \Delta,$$

where

$$k \leq U^{\text{adv}} / \left\lceil \frac{\epsilon_{\text{global}}}{n(1+r)\epsilon_{\text{querier}} \cdot \kappa_{\text{action}}} \right\rceil.$$

*Proof.* Lemma B.25 covers the first statement.

Denote by  $X_{\text{dep}}$  the  $k$  depleted device-epochs, and take  $x \in X_{\text{dep}}$ , writing  $M_x^{\text{adv}}$  and  $U_{\text{imp}}^{\text{adv}}(x)$  for the corresponding quantities at  $x$ . The attacker's own reports consume the whole capacity of  $\mathcal{F}_x^{\text{global}}$ , so  $\epsilon_{\text{global}}^{\text{adv}} = \epsilon_{\text{global}}$  at  $x$ . By Lem. B.23 and Eq. 98, and using the configuration from Table 2 where  $\epsilon_{\text{imp-quota}} = n(1+r)\epsilon_{\text{querier}}$ :

$$\epsilon_{\text{global}} \leq M_x^{\text{adv}} \epsilon_{\text{imp-quota}} \leq n(1+r)\epsilon_{\text{querier}} \times \kappa_{\text{action}} \times U_{\text{imp}}^{\text{adv}}(x).$$

Rearranging gives  $U_{\text{imp}}^{\text{adv}}(x) \geq \frac{\epsilon_{\text{global}}}{n(1+r)\epsilon_{\text{querier}} \cdot \kappa_{\text{action}}}$ , and  $U_{\text{imp}}^{\text{adv}}(x)$  is an integer, so we can round the right-hand side up. By Def. B.1, all the impressions of one action on  $d$  belong to a single device-epoch. The device-epochs of  $X_{\text{dep}}$  therefore draw disjoint sets of actions from the  $U^{\text{adv}}$  actions collected on  $d$ :

$$U^{\text{adv}} \geq \sum_{x \in X_{\text{dep}}} U_{\text{imp}}^{\text{adv}}(x) \geq k \left\lceil \frac{\epsilon_{\text{global}}}{n(1+r)\epsilon_{\text{querier}} \cdot \kappa_{\text{action}}} \right\rceil,$$

which gives the bound on  $k$ .  $\square$

## B.5 Simplified model (for educational purposes)

Alg. 8 shows a simplified version of the full Big Bird system model that we analyze formally, but one in line with the generic model from §3.1. The purpose of including this simplified version is educational: it provides easy-to-read understanding of how one can represent key aspects of system behavior in modeling.

First, the budgets' IDP semantics are visible in the  $x \in \mathcal{X}$  subscript at Line 7, where  $x$  corresponds to a device-epoch. This indicates that the filters are each maintained on each device and for each epoch. Second, privacy loss is computed at a finer granularity (Line 21) for  $\text{imp-quota}$  than for the other budgets. Third, Big Bird uses a shared limit ( $\mathcal{L}$  in §3) consisting of a collection of privacy budgets and quota budgets, denoted  $\mathcal{F}$ . At Line 22,  $(\mathcal{F}_x^{\text{global}} \circ \dots \circ \mathcal{F}_x^{\text{imp-quota}}[i])(\epsilon_x, \epsilon_x^i)$  denotes the atomic check and consumption of budgets performed for each record (device-epoch). For comparison,  $\mathcal{F}_x^{\text{global}}(\epsilon_x) \wedge \dots \wedge \mathcal{F}_x^{\text{imp-quota}}[i](\epsilon_x^i)$  would check budgets independently, potentially spending from one but not the other, which harms utility. We show that this atomicity, never previously studied, preserves global IDP.

## C Examples of Cap Interactions with Device-EPOCH IDP

To illustrate how easy it is for seemingly reasonable DoS caps to violate device-epoch IDP, we present a natural but incorrect instantiation of “capping global-budget consumption per user action,” followed by its corrected form.

---

**Alg. 8** Simplified version of Big Bird formal model.

---

```
1: Input
2: Challenge bit  $\chi \in \{0, 1\}$ 
3: In-or-out record  $x_0$  to be inserted at time  $t_0$ 
4: Public information  $C$ 
5: Data generation process  $\mathcal{G}$ 
6: Queriers  $Q^1, \dots, Q^K$ 
7: IDP budgets for each  $x \in \mathcal{X}$ :  $\mathcal{F}_x^{\text{global}}, (\mathcal{F}_x^{\text{querier}}[k])_{k \in [K]},$   

 $(\mathcal{F}_x^{\text{conv-quota}}[c])_{c \in [S]}, (\mathcal{F}_x^{\text{imp-quota}}[i])_{i \in [S]}$ 
8: for  $t \in [t_{\max}]$  do
  // Data generation model:
9:  $D_t \leftarrow \mathcal{G}(V_{<t}^1, \dots, V_{<t}^K)$  // new data depends on all past results
10: if  $t = t_0$  then
  // On challenge, insert in-or-out record...
11: if  $\chi = 0$  then
12:    $D_t \leftarrow D_t + x_0 \cap C$  // ...with public events only
13: else
14:    $D_t \leftarrow D_t + x_0$  // ...with public and private events
  // Query execution model:
15: for  $k \in [K]$  do
16:    $q_t^k \leftarrow Q^k(V_{<t}^k)$  // Abstract query over whole database
17:    $S_t^k \leftarrow D$ 
18:   for  $x \in D$  do
    // Compute device-epoch loss and device-epoch-impSite loss (for
    imp-quota)
19:    $\epsilon_x \leftarrow \text{EpochBudget}(x, q_t^k)$ 
20:   for  $i \in \mathbf{i}$  do
21:    $\epsilon_x^i \leftarrow \text{EpochImpSiteBudget}(x, i, q_t^k)$ 
    // Atomic transaction across all budgets
22:   if  $(\mathcal{F}_x^{\text{global}} \circ \mathcal{F}_x^{\text{querier}}[k] \circ \mathcal{F}_x^{\text{conv-quota}}[c] \circ$   

 $\mathcal{F}_x^{\text{imp-quota}}[i])(\epsilon_x, \epsilon_x^i)$  then
23:    $S_t^k \leftarrow S_t^k \setminus \{x\}$  // Drop out-of-budget record
24:    $V_t^k \leftarrow q_t^k(S_t^k)$ 
25: return  $V^1, \dots, V^K$ 
```

---

**Faulty instantiation.** Alg. 9 tries to cap the total privacy consumption generated by all reports within a single user-action context. A natural approach would be to cap the *cumulative* privacy loss *across all epochs* accessed by reports following that user action. The algorithm does exactly this by maintaining a single accumulator, `consumedPrivacy[uaCtx]`, that aggregates privacy loss over every epoch touched by any report in that `uaCtx`. If, during the execution of a report, this cumulative loss would exceed the cap  $\epsilon_{\text{action}}$ , the algorithm halts and returns a null report, `AttributionFn( $\emptyset, \dots, \emptyset$ )`.

While intuitive, this design inadvertently couples epochs: whether data from epoch  $e$  is included in the report now depends on privacy consumption in *other* epochs accessed earlier in the same `uaCtx`. This violates device-epoch IDP, which requires that privacy loss in epoch  $e$  depend only on that epoch's data. In effect, a querier's interactions in epoch

---

**Alg. 9** Non-IDP capping of budget consumption per user action

---

```
def onUserAction()
  uaCtx = createUserContext()
  consumedPrivacy[uaCtx]  $\leftarrow$  0
  return uaCtx

def measureConversion(uaCtx, impSites, querier, convSite, params)
for  $e \in \text{params.attributionWindow}$  do
   $I_e \leftarrow \text{MatchingFn}(\text{ImpressionStore}[e], \text{params})$ 
  // Compute IDP loss w/ Cookie Monster algo.
   $\epsilon_e \leftarrow \text{computeEpochLoss}(I_e, \text{params})$ 
  // Ensure that total privacy consumed in this user-action context across
  // all epochs stays within the cap. If it exceeds, stop execution and return
  // a null report.
  if consumedPrivacy[uaCtx] +  $\epsilon_e$  >  $\epsilon_{\text{action}}$  then
    return AttributionFn( $\emptyset, \dots, \emptyset$ ) // cap exceeded, return null report
  consumedPrivacy[uaCtx] +  $\epsilon_e$ 
  // Then deduct from per-epoch global budgets and per-querier budgets.
  tx = beginTransaction()
  if querierBudget[e][querier].tryConsume( $\epsilon_e$ , tx) and
    globalBudget[e].tryConsume( $\epsilon_e$ , tx) then
    tx.commit()
  else
    tx.abort()
     $I_e \leftarrow \emptyset$ 
  // Generate attribution histogram from remaining impressions
return AttributionFn( $I_{e_1}, \dots, I_{e_w}$ ) //  $e_i \in \text{params.attributionWindow}$ 
```

---

$e_1$  can suppress contributions from epoch  $e_2$ , which means that privacy loss accounted for  $e_2$  now depends on data in  $e_1$ , violating IDP.

**Correct instantiation.** Alg. 10 restores IDP soundness for this candidate approach by isolating caps *per epoch*. Each user-action context maintains a separate tracker for every epoch:

`consumedPrivacy[uaCtx][e].`

When epoch  $e$ 's loss in this `uaCtx` exceeds  $\epsilon_{\text{action}}$ , only that epoch's contribution is nullified; all other epochs proceed independently. This preserves the required per-epoch separability: the decision to include or exclude epoch  $e$  depends solely on that epoch's data and budgets, not on privacy loss incurred elsewhere. This same principle underlies Big Bird's per-action domain cap design (Alg. 2), where checks are carefully partitioned per epoch and user-action contexts never span epoch boundaries. Together, these mechanisms ensure that resilience mechanisms do not inadvertently break device-epoch IDP.

## D Cross-report Privacy Loss Optimization

In §B, we presented a general algorithm where queriers pay for each report separately. It is common for conversion sites to embed multiple intermediaries and to perform attribution once across impressions displayed by all intermediaries. When multiple reports for the same conversion involve disjoint impression sets (*e.g.*, across intermediaries),

---

**Alg. 10 Correct, IDP-respecting capping of budget consumption per user action**


---

```

def onUserAction()
  uaCtx = createUserContext()
  consumedPrivacy[uaCtx][e] ← 0 for each epoch e
  return uaCtx

def measureConversion(uaCtx, impSites, querier, convSite, params)
  for e ∈ params.attributionWindow do
    Ie ← MatchingFn(ImpressionStore[e], params)
    // Compute IDP loss w/ Cookie Monster algo.
    εe ← computeEpochLoss(Ie, params)
    // Ensure that privacy consumed per epoch in this user-action context
    // stays within the cap. If it exceeds, eliminate the epoch's data from
    // consideration for this report.
    if consumedPrivacy[uaCtx][e] + εe > εaction then
      Ie ← ∅ // nullify contribution of this epoch to the report
    consumedPrivacy[uaCtx][e] += εe
    // Then deduct from per-epoch global budgets and per-querier budgets.
    tx = beginTransaction()
    if querierBudget[e][querier].tryConsume(εe, tx) and
      globalBudget[e].tryConsume(εe, tx) then
      tx.commit()
    else
      tx.abort()
      Ie ← ∅
    // Generate attribution histogram from remaining impressions
  return AttributionFn(Ie1, ..., Iew) // ei ∈ params.attributionWindow

def onEpochChange(uaCtx)
  destroyUserContext(uaCtx)

```

---

we prove that one need not double-count against shared budgets (global, conversion-site, and impression-site quotas). Indeed, the combination of reports from different intermediaries reveals no more than a single attribution across intermediaries. We present the intuition first with an example and then with detailed formalism and proofs.

### D.1 Example

We modify the example in Fig. 4 and §4.3 as follows. Instead of working with a single intermediary `adtech.ex`, `shoes.ex` works with both `adtech.ex` and `adtech2.ex`. `adtech.ex` displays impressions on `news.ex`, and `adtech2.ex` on `blog.ex`. Upon conversion, attribution is still computed across all relevant impressions, spanning both intermediaries. For `shoes.ex`'s own measurement queries, accounting is unchanged since intermediaries operate on `shoes.ex`'s behalf and are not queriers themselves. As before, `adtech.ex` can request an additional report (j) measuring its own impressions for placement optimization purposes. Here, `adtech.ex` receives a report containing a histogram with attributed value 30 (for the impression on `news.ex`). Additionally, `adtech2.ex` can measure its own impressions and receive a report (k) containing a histogram with attributed value 30. Naively, one would expect a cumulative deduction of 0.9 from the global budget (three reports). Yet, one can observe that the two intermediary reports are shards of a single report with attrib-

uted value 60, because `adtech.ex` and `adtech2.ex` receive disjoint bins of the histogram. Leveraging this observation, we show that it is possible to deduct only 0.6 from the global budget. The same optimization applies to shared quotas.

### D.2 Problem Statement

The example in §D.1 admits a cross-report optimization, which we formalize here. We define variations of Alg. 7 and Alg. 5 in §D.3. We focus on histogram reports, defined in §D.4, and show that paying only once for a sequence of properly correlated histogram reports still satisfies global IDP guarantees in §D.5. We leave the generalization to other queries for future work.

### D.3 Algorithm

**Overview.** Alg. 11 updates the formalism from Alg. 7. Instead of generating a report immediately upon conversion, the first querier to request a report attached to a report identifier  $r$  calls `MeasureConversion` to create a stateful attribution object  $\alpha_r$ , which pays upfront for any valid sequence of histogram reports over non-overlapping impressions.

`MeasureConversion` performs attribution using the pre-specified attribution function  $a$  and the noise scale  $\lambda$ , which give an upper bound on the total leakage of any sequence of future reports applying  $a$  on disjoint sets of impressions. All shared privacy loss is charged upfront and atomically at `MeasureConversion` time, before any querier-specific interaction occurs. We run a modified two-phase commit protocol to deduct budget from all shared filters (global and quota filters), but not from querier-local filters, which are charged separately.<sup>2</sup> Finally,  $\alpha_r$  stores the attribution function  $a_F$  with corresponding impressions  $F$ , the privacy parameters  $\lambda$ , and the support  $S$  of impressions requested so far.

`GetReport` allows a querier to receive a report from  $r$ , once the attribution object  $\alpha_r$  has been created.  $\alpha_r$  is only created once, and is reused every time a querier requests a report from  $r$ . `GetReport` checks that privacy parameters match what was committed in the attribution object, and that impressions are not queried twice. If these checks pass, `GetReport` spends querier budget and computes the report using the predefined attribution function  $a$  stored in  $\alpha_r$ . We update the support of impressions  $S \leftarrow S \sqcup S_\rho$  in  $\alpha_r$  each time a new report  $\rho$  requests impressions in  $S_\rho \subset I$ .

**Subroutine.** We define `AtomicFilterCheckAndConsume2` as in Alg. 6, except that querier filters are not part of the atomic commit. That is, we do not take  $b$  as an input, and we delete Line 6 and Line 16 from Alg. 6.

---

<sup>2</sup>The querier budget is left out of the 2PC because we don't know ahead of time which queriers will request a report, and we don't want to block some queriers if other queriers are out of budget.

**Alg. 11** Updated formalism for cross-report optimization (diff with Alg. 7)

---

```

1: // Collect, aggregate and noise reports to answer Q
2: def AnswerQuery(D, Q, λ, b)
3:   (ρrb)r∈R ← Q // Get report identifiers from Q
4:   for r ∈ R do
5:     if αr is not defined then
6:       // Initialize attribution object
7:       αr ← MeasureConversion(D, ρrb, λ)
8:       ρrb(D), αr ← GetReport(αr, ρrb, λ, b)
9:   Sample X ~ L(λ)
10:  return ∑r∈R ρrb(D) + X

```

---

#### D.4 Histogram reports definition and properties

The following definitions (Def. D.1 and D.2) are adapted from [43, Thm. 18]. Histogram reports distribute a positive value across impressions, map each impression to a bucket, and then sum up the attributed value in each bucket. Lem. D.3 gives the global sensitivity of such histogram reports, which is bounded by the maximum attributable value.

**Definition D.1** (Scalar attribution function). Fix a number of epochs  $k > 0$ . A *scalar attribution function* is a function  $a : \mathcal{P}(\mathcal{I})^k \times \mathcal{I} \rightarrow \mathbb{R}_+$  that attributes a positive value  $a_F(f)$  to each impression  $f \in \mathcal{I}$ , depending on all the impressions in  $k$  epochs  $F \in \mathcal{P}(\mathcal{I})^k$ .

For a scalar attribution function  $a$ , we define its *maximum attributable value*  $a^{\max}$  as follows:

$$a^{\max} := \max_{F \in \mathcal{P}(\mathcal{I})^k} \sum_{j=1}^k \sum_{f \in F_j} a_F(f) \quad (101)$$

**Definition D.2** (Histogram report). Consider a scalar attribution function  $a : \mathcal{P}(\mathcal{I})^k \times \mathcal{I} \rightarrow \mathbb{R}_+$ , a support set of impressions  $S \subset \mathcal{I}$ , an output dimension  $m > 0$ , and a one-hot encoding function  $H$  that maps each event  $f$  to one of  $m$  buckets. That is,  $H : \mathcal{I} \rightarrow \{0, 1\}^m$  such that  $\forall f \in \mathcal{I}, \|H(f)\|_1 = 1$ .

First, we define  $A_{a,S,H} : \mathcal{P}(\mathcal{I})^k \rightarrow \mathbb{R}^m$  as follows:

$$A(F) = \sum_{j=1}^k \sum_{f \in F_j} \mathbb{1}[f \in S] a_F(f) \cdot H(f) \quad (102)$$

$A_{a,S,H}$  is a well-defined attribution function (in the sense of Def. B.5).

Second, for a device  $d$  and a set of epochs  $E$ , we define the *histogram report* associated with  $A_{a,S,H}$  as in Def. B.6:

$$\rho : D \mapsto A_{a,S,H}(D_d^E) \quad (103)$$

Next, Lem. D.3 and D.4 give two preliminary properties of histogram reports that will be used in Thm. D.5.

**Lemma D.3** (Histogram sensitivity). *Consider a histogram*

**Alg. 12** Big Bird algorithm (on-device) with cross-report optimization for histogram reports

---

```

1: Input
2: Filter and quota capacities εglobal, εquerier, εimp-quota,
   εconv-quota
3: def MeasureConversion(D, ρ, λ, b)
4:   Read ρ to get device d, epoch E, conversion site c, im-
     pression sites i, histogram attribution function Aaρ,Sρ,Hρ
     with scalar attribution function a, support impressions
     S and histogram bin mapping H.
5:   for e ∈ E do
6:     x ← (d, e, Dde)
7:     if Fx is not defined then
8:       Fx ← InitializeFilters(εglobal, εquerier,
         εimp-quota, εconv-quota)
9:       εx ← 2amax/λ
10:      εxi ← {i : 2amax/λ, i ∈ i}
11:      if AtomicFilterCheckAndConsume2(Fx, c, i, εx, εxi) =
        FALSE then
12:        Fe ← ∅ // Empty the epoch if any filter check fails
13:      return α = (aF, F, λ, ∅) // Start with Sα = ∅
14:
15: def GetReport(α, ρ, λ)
16:   Read α to get attribution function aα, impressions Fα,
     noise scale λα.
17:   Read ρ to get device d, querier b, target epochs E, his-
     togram attribution function Aa,S,H with scalar attribu-
     tion function a, support impressions S and histogram
     bin mapping H.
18:   if S ∩ Sα ≠ ∅ ∨ λ ≠ λα ∨ aρ ≠ aα then
19:     ρ ← A(∅, ..., ∅) // Null report if inconsistent with α
20:     return ρ
21:   for e ∈ E do
22:     x ← (d, e, Fαe)
23:     εx ← EpochBudget(x, ρ, λ) // Querier budget only
24:     if Fxquerier[b].tryConsume(εxi) = FALSE then
25:       Fe ← ∅
26:     else
27:       Fe ← Fαe
28:   α ← (aα, F, λ, Sα ∪ Sρ) // Update impression support
29:   ρ ← Aa,S,H((Fe)e∈E) // Clipped attribution report
30:   return ρ, α

```

---

report  $\rho$  with associated attribution function  $A_{a,S,H}$ . We have:

$$\Delta(\rho) \leq 2a^{\max} \quad (104)$$

*Proof.* Take a report  $\rho$  with scalar attribution function  $a$ , device  $d$  and epochs  $E$ . Consider two neighboring databases

$D, D'$  and denote  $\mathbf{F} := D_d^E$  and  $\mathbf{F}' := D_d'^E$ . We have:

$$\|\rho(D) - \rho(D')\|_1 = \|A_{a,S,H}(\mathbf{F}) - A_{a,S,H}(\mathbf{F}')\|_1 \quad (105)$$

$$= \left\| \sum_{j=1}^k \sum_{f \in \mathbf{F}_j} a_{\mathbf{F}}(f) \cdot H(f) - \sum_{j=1}^k \sum_{f \in \mathbf{F}'_j} a_{\mathbf{F}'}(f) \cdot H(f) \right\|_1 \quad (106)$$

$$\leq \sum_{j=1}^k \sum_{f \in \mathbf{F}_j} a_{\mathbf{F}}(f) \|H(f)\| + \sum_{j=1}^k \sum_{f \in \mathbf{F}'_j} a_{\mathbf{F}'}(f) \|H(f)\| \quad (107)$$

$$\leq 2a^{\max} \quad (108)$$

Even though this bound holds even for non-neighboring databases, [43, Thm. 18] provides mild conditions under which the bound is tight.  $\square$

## D.5 Privacy proof

**Lemma D.4** (Correlated histogram sensitivity). *Fix a device-epoch  $x = (d, e, F)$  and a database  $D$ . Fix a report identifier  $r \in \mathbb{Z}$  corresponding to a histogram attribution object  $\alpha_r$ . Fix a sequence of reports  $\rho_1, \dots, \rho_n$  that request a report from  $r$ , ordered lexicographically by time and querier  $(t, b)$ . In particular, MeasureConversion is called for  $\rho_1$  and then reused for subsequent reports.  $\sum_{i=1}^n \|\rho_i(D) - \rho_i(D+x)\|/\lambda_i$  represents the total contribution over reports computed from  $\alpha_r$ , each with its own requested noise scale  $\lambda_i$ . Denote by  $\text{pass}_1$  the output of the 2PC for  $x$  in Alg. 12.*

We have:

$$\sum_{i=1}^n \|\rho_i(D) - \rho_i(D+x)\|/\lambda_i \leq \text{pass}_1 \cdot 2a^{\max}/\lambda_1 \quad (109)$$

*Proof.* If  $\text{pass}_1 = 0$ , then  $\rho_i(D) = \rho_i(D+x)$  because in both cases the data for  $x$  is zeroed out ( $F_e = \emptyset$  at Line 12), which completes the proof in this case.

Now, suppose that  $\text{pass}_1 = 1$ . Take a report  $\rho_i$  with scalar attribution function  $a_i$  and support impressions  $S_i$ . These only depend on past results  $v_{<t_i, b_i}$ . If  $S_i \cap (S_1 \cup \dots \cup S_{i-1}) \neq \emptyset$ ,  $\lambda_i \neq \lambda_1$  or  $a_i \neq a_1$ , then  $\rho_i(D) = A(\emptyset, \dots, \emptyset) = \rho_i(D+x)$ . Also, if  $x = (d, e, F)$  is not queried by  $\rho_i$  ( $d \neq d_i$  or  $e \notin E_i$ ), then  $\rho_i(D) = \rho_i(D+x)$ .

Denote by  $I$  the set of remaining reports satisfying  $S_i \cap (S_1 \cup \dots \cup S_{i-1}) = \emptyset$ ,  $\lambda_i = \lambda_1$ ,  $a_i = a_1$ ,  $d_i = d$ , and  $e \in E_i$ . We have:

$$\sum_{i=1}^n \|\rho_i(D) - \rho_i(D+x)\|/\lambda_i \quad (110)$$

$$= \sum_{i \in I} \|\rho_i(D) - \rho_i(D+x)\|/\lambda_1 \quad (111)$$

$$= \sum_{i \in I} \left\| \sum_{j=1}^k \sum_{f \in \mathbf{F}_j} \mathbb{1}[f \in S_i] a_{\mathbf{F}}(f) H_i(f) - \sum_{j=1}^k \sum_{f \in \mathbf{F}'_j} \mathbb{1}[f \in S_i] a_{\mathbf{F}'}(f) H_i(f) \right\|/\lambda_1 \quad (112)$$

$$\sum_{j=1}^k \sum_{f \in \mathbf{F}'_j} \mathbb{1}[f \in S_i] a_{\mathbf{F}'}(f) H_i(f) \|\lambda_1 \quad (113)$$

$$\leq \sum_{i \in I} \sum_{j=1}^k \sum_{f \in \mathbf{F}_j} \mathbb{1}[f \in S_i] (a_{\mathbf{F}}(f) + a_{\mathbf{F}'}(f))/\lambda_1 \quad (114)$$

$$\leq \sum_{j=1}^k \sum_{f \in \mathbf{F}_j} \mathbb{1}[f \in \sqcup_{i \in I} S_i] (a_{\mathbf{F}}(f) + a_{\mathbf{F}'}(f))/\lambda_1 \quad (115)$$

$$\leq 2a^{\max}/\lambda_1 \quad (116)$$

by Def. D.1 and using the fact that the  $S_i$  are disjoint so each impression is counted at most once.  $\square$

**Theorem D.5.** *Consider  $x \in \mathcal{X}$  with global budget capacity  $\epsilon_{\text{global}}$ . For simplicity, because we are considering a global guarantee, we consider a non-adaptive data generation process in Alg. 11, which defines a mechanism  $q$ . Then,  $q$  satisfies individual device-epoch  $\epsilon_{\text{global}}$ -DP for  $x$  under public information  $C$ .*

*Proof.* Denote by  $x_C = (d, e, F \cap C)$  the device-epoch obtained by keeping only public events  $C$  from  $x$ , where public events are the set of all conversions.

Take  $v \in \text{Range}(q)$ . As in Thm. 4.2, we want to show that:

$$\left| \ln \left( \frac{\Pr[q(D+x_C) = v]}{\Pr[q(D+x) = v]} \right) \right| \leq \epsilon_{\text{global}}. \quad (117)$$

Using Bayes' rule and  $\rho(D+x_C) = \rho(D)$ , as in Thm. 4.2, we have:

$$\left| \ln \left( \frac{\Pr[q(D+x_C) = v]}{\Pr[q(D+x) = v]} \right) \right| \quad (118)$$

$$= \left| \ln \left( \frac{\prod_{t=1}^{t_{\max}} \prod_{b \in S} \Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D) + X_t^b = v_t^b]}{\prod_{t=1}^{t_{\max}} \prod_{b \in S} \Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D+x) + X_t^b = v_t^b]} \right) \right| \quad (119)$$

$$\leq \sum_{t=1}^{t_{\max}} \sum_{b \in S} \left| \ln \left( \frac{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D) + X_t^b = v_t^b]}{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D+x) + X_t^b = v_t^b]} \right) \right| \quad (120)$$

where each query  $Q_t^b = \{\rho_{r,t}^b, r \in R_t^b\}$  is chosen adaptively, potentially based on previous results  $v_1^b, \dots, v_{t-1}^b$ . Since the filters and attribution functions are identical for  $D$  and  $D+x$  when we condition on past results, we write  $\rho_{r,t}^b(D)$  for

simplicity instead of  $\rho_{r,t}^b(D; \mathcal{F}_{v_{<t}}, \alpha_{v_{<t}})$ .

Fix  $t \in [t_{\max}]$  and  $b \in \mathcal{S}$ . Without the optimization, as in Eq. 84, the device would pay for each report sent to any querier, which would give:

$$\left| \ln \left( \frac{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D) + X_t^b = v_t^b]}{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D+x) + X_t^b = v_t^b]} \right) \right| \quad (121)$$

$$\leq \sum_{r \in R_t^b} \Delta_x(\rho_{r,t}^b) \text{pass}_r^b / \lambda_t^b \quad (122)$$

Instead of upper-bounding the difference  $\|\rho_{r,t}^b(D) - \rho_{r,t}^b(D+x)\|$  for each report by  $\Delta_x(\rho_r)$  separately, which takes a maximum over all  $D$  right away, we keep information about  $D$  a bit longer. This will allow us to leverage the fact that reports  $\rho_{r,t}^b$  across different time steps and queriers tied to the same identifier  $r$  are correlated:

$$\left| \ln \left( \frac{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D) + X_t^b = v_t^b]}{\Pr[\sum_{r \in R_t^b} \rho_{r,t}^b(D+x) + X_t^b = v_t^b]} \right) \right| \quad (123)$$

$$\leq \sum_{r \in R_t^b} \|\rho_r^b(D) - \rho_r^b(D+x)\| / \lambda_t^b \quad (124)$$

For a report identifier  $r \in \mathbb{Z}$ , we now define  $\mathcal{T}_r$ , which keeps track of all queriers that requested a report from  $r$  and the time steps at which they requested it:

$$\mathcal{T}_r := \{(t, b) \in [t_{\max}] \times \mathcal{S} : r \in R_t^b\} \quad (125)$$

This notation allows us to swap the sums, after putting Eq. 124 into Eq. 120:

$$\left| \ln \left( \frac{\Pr[q(D+x_C) = v]}{\Pr[q(D+x) = v]} \right) \right| \quad (126)$$

$$\leq \sum_{t=1}^{t_{\max}} \sum_{b \in \mathcal{S}} \sum_{r \in R_t^b} \|\rho_r^b(D) - \rho_r^b(D+x)\| / \lambda_t^b \quad (127)$$

$$= \sum_{r \in \mathbb{Z}} \sum_{(t,b) \in \mathcal{T}_r} \|\rho_r^b(D) - \rho_r^b(D+x)\| / \lambda_t^b \quad (128)$$

Fix a report identifier  $r \in \mathbb{Z}$  corresponding to a histogram attribution object  $\alpha_r$ . Denote by  $(t_0, b_0)$  the first time-step/querier pair that requests a report from  $r$ , thereby calling MeasureConversion. By Lem. D.4, we have:

$$\sum_{(t,b) \in \mathcal{T}_r} \|\rho_r^b(D) - \rho_r^b(D+x)\| / \lambda_t^b \leq \text{pass}_{r,t_0}^{b_0} \cdot \Delta_x(\alpha_r) / \lambda_{t_0}^{b_0} \quad (129)$$

Finally, since  $\text{pass}_{r,t_0}^{b_0}$  implies that  $\epsilon_{r,t_0}^{b_0} = \Delta_x(\alpha_r) / \lambda_{t_0}^{b_0}$  passes the global budget, by the definition of global budget, Eq. 128 becomes:

$$\left| \ln \left( \frac{\Pr[q(D+x_C) = v]}{\Pr[q(D+x) = v]} \right) \right| \quad (130)$$

$$\leq \sum_{r \in \mathbb{Z}} \text{pass}_{r,t_0}^{b_0} \epsilon_{r,t_0}^{b_0} \quad (131)$$

$$\leq \epsilon_{\text{global}} \quad (132)$$

which concludes the proof.  $\square$

**Remark.** Lem. D.3 and D.4 show that for histogram reports with  $k > 1$  epochs, Alg. 12 spends up to  $|\mathbf{b}|$  times less budget than Alg. 5 when  $|\mathbf{b}|$  queriers request reports from the same conversion with a single report identifier  $r$ . This is because the privacy loss in these cases is proportional to  $\Delta(\rho) = 2a^{\max}$ . For histogram reports with a single epoch, we can use the individual sensitivity, which renders this optimization unnecessary.