Private and Continual Release of Statistics

T-H. Hubert Chan[1], Elaine Shi[2], and Dawn Song[3]

[1] The University of Hong Kong
[2] PARC
[3] UC Berkeley

Abstract. We ask the question – *how can websites and data aggregators continually release updated statistics, and meanwhile preserve each individual user's privacy?* Given a stream of 0's and 1's, we propose a differentially private continual counter that outputs at every time step the approximate number of 1's seen thus far. Our counter construction has error that is only poly-log in the number of time steps. We can extend the basic counter construction to allow websites to continually give top-k and hot items suggestions while preserving users' privacy.

1 Introduction

Websites such as online retailers, search engines and social networks commonly publish aggregate statistics about their users to realize valuable social and economic utilities. Moreover, the published statistics are continually updated over time as new data arrive. Such practices are ubiquitous and we name a few examples below. Sites such as Amazon, IMDB, Delicious and Flickr recommend popular items to users to enhance their browsing experience and engage their interests. Search engines such as Google and Yahoo help a user to auto-complete her search query by suggesting the most frequent search terms matching the prefix specified by the user. During political campaigns, websites survey the population and continually update the support rates for candidates.

Releasing aggregate information about users may seem harmless at first glance. However, previous work has shown that such statistical disclosures can expose sensitive information about an individual user [3, 11]. In particular, sites that continually update the published statistics over time can give even more leverage to the adversary and result in more severe privacy leakage [1].

In this paper, we ask the question – *how can we guarantee the users' privacy when a website must continually publish new statistics as new data arrive?* Independent from our work, Dwork *et.al.* also consider essentially the same problem, and they phrase the problem as *"differential privacy under continual observation"* [6, 9, 10].

The setting we consider is different from the traditional setting in which differential privacy was studied. The traditional setting assumes a static database, and a curator who must answer k interactive queries or publish some sanitized statistics of the database non-interactively. In our setting, the database is dynamic and evolves over time, and a mechanism must update the published statistics as new data items arrive. Therefore, traditional differentially private mechanisms either fail to apply directly to our setting, or result in an unsatisfactory loss in terms of utility or privacy if applied naively.

1.1 Contributions

Differentially private continual counter with poly-log error. We consider the *continual counting problem*. Assume that the input stream $\sigma \in \{0,1\}^{\mathbb{N}}$ is a sequence of bits. The bit $\sigma(t)$ at time $t \in \mathbb{N}$ may denote whether an event of interest occurred at time t, e.g., whether a user purchased an item at time t. At every time step $t \in \mathbb{N}$, the mechanism must output an approximate count of the number of 1's seen thus far.

We design an ϵ-differentially private continual counter with small error. Specifically, for each $t \in \mathbb{N}$, with probability at least $1-\delta$, we guarantee $O(\frac{1}{\epsilon} \cdot (\log t)^{1.5} \cdot \log \frac{1}{\delta})$ error[4]. In an independent work by Dwork *et.al.* [9], they also show a similar upper bound of $O(\frac{1}{\epsilon} \cdot (\log t)^{1.5})$ (omitting the δ term). The above upper bound is almost tight, since Dwork *et.al.* [9] show that any ϵ-differentially private mechanism will for some stream, with probability at least δ, make an error of at least $\Omega(\frac{1}{\epsilon}(\log T + \log \frac{1}{\delta}))$ at some time before T.

Our mechanism achieves *time unboundedness*, i.e., the mechanism does not require a priori knowledge of an upper bound on the time for which it will run, and provides guarantees even when it is run indefinitely. This represents an improvement over the work by Dwork *et.al.* [6] – their mechanism requires a priori knowledge of an upper bound on the number of time steps.

Pan privacy. Dwork *et.al.* first introduced the notion of pan privacy [6, 10]. A mechanism is pan privacy if it can preserve differential privacy even when an adversary can observe snapshots of the mechanism's internal states, e.g., in subpoenas. We show how to modify our mechanism to achieve pan privacy, without incurring any loss in the asymptotic guarantees (Section 5).

Applications. Our continual counter construction has immediate practical applications. As mentioned earlier, it is a common practice for websites to suggest to users the most popular movies, news items or photos. We show how websites can continually make such top-k or hot items suggestions in a differentially private manner. Due to limited space, we describe applications in the full version of the paper [18].

The counter is also an important primitive in numerous data streaming algorithms [2, 14, 16]. Our differentially private continual counter is an initial step towards designing a broad class of streaming algorithms that continually report outputs over time.

1.2 Related Work

Most closely related work. Independent from our work, Dwork *et.al.* show a similar result in a recent paper [9]. They also construct a differentially private continual counter with error $O(\frac{1}{\epsilon} \cdot (\log t)^{1.5})$ where t is the number of timesteps. Moreover, they show a lower bound of $\Omega(O(\frac{1}{\epsilon} \cdot (\log t))$, indicating that the upper bound is almost tight. A preliminary version of the result was revealed at the SODA'10 conference [6] in an

[4] For large values of t, we can actually get a better bound $O(\frac{1}{\epsilon} \cdot (\log t)^{1.5} \cdot \sqrt{\log \frac{1}{\delta}})$. To get a high probability statement, we can set $\delta := \frac{1}{\text{poly}(t)}$ and the corresponding error becomes $O((\log t)^2/\epsilon)$.

invited talk by Dwork. The preliminary result contains a slightly looser upper bound – a differentially private continual counter with error square root in the number of 1's seen thus far.

Dwork, Naor, Pitassi and Rothblum [10] recently propose the notion of pan privacy, i.e., how to achieve differential privacy even in the presence of intrusions, in which the adversary is allowed access to the mechanism's internal states. Dwork *et.al.* used the notion of pan privacy in the continual counter mechanism [6, 9], and showed how to make their counter mechanism resilient against a single unannounced intrusion. Inspired by their techniques, we also convert our mechanism to a pan private version that is immune to a single unannounced intrusion or multiple afterwards announced intrusions.

Differential privacy in the traditional setting. In the traditional setting, a trusted curator who holds a large data set must respond to queries *interactively* or publish sanitized statistics about the data *non-interactively*. The notion of differential privacy was first proposed and studied by Dwork *et.al.* [4, 8]. An extensive literature has since emerged, studying the different tradeoffs between utility and privacy. To better understand the motivation and state-of-the-art of this line of research, we recommend the readers to these excellent survey papers by Dwork [5, 7].

Researchers have also applied theoretical results in differential privacy to real-world applications. For example, McSherry and Mironov show how to build privacy into the Netflix database published for the Netflix contest [15]. Korolova *et.al.* show how to release search logs and click logs privately [13].

Attacks against privacy. A complementary line of research is attacks against privacy. Narayanan *et.al.* show how to de-anonymize the Netflix data set [17]. Jones *et.al.* show how to break the privacy of query log bundles [12]. More relevant to this work, Calandrino *et.al.* [1] recently demonstrate that by observing continual updates from websites such as Amazon over a period of time, an adversary can learn individual user behavior at a fine level of granularity. Our work is partly inspired by the problem they expose.

2 Preliminaries

2.1 Definitions

We consider streams of 0's and 1's. Formally, a stream $\sigma \in \{0,1\}^{\mathbb{N}}$ is a bit-string of countable length, where $\mathbb{N} := \{1, 2, 3, \ldots\}$ is the set of positive integers. Specifically, $\sigma(t) \in \{0,1\}$ denotes the bit at time $t \in \mathbb{N}$. We write $[T] := \{1, 2, 3, \ldots, T\}$ and $\sigma_T \in \{0,1\}^T$ is the length T prefix of the stream σ. We will use the term *item* to refer to a bit in the stream.

At every time t, we wish to output the number of 1's that have arrived up to time t.

Definition 1 (Continual Counting Query). *Given a stream $\sigma \in \{0,1\}^{\mathbb{N}}$, the count for the stream is a mapping $c_\sigma : \mathbb{N} \to \mathbb{Z}$ such that for each $t \in \mathbb{N}$, $c_\sigma(t) := \sum_{i=1}^{t} \sigma(i)$. We write c instead of c_σ when there is no risk of ambiguity on the stream σ in question.*

We now formally define the notion of a continual counting mechanism which continually outputs the number of 1's seen thus far.

Definition 2 (Counting Mechanism). *A counting mechanism $\mathcal{M}$ takes a stream $\sigma \in \{0,1\}^{\mathbb{N}}$ and produces a (possibly randomized) mapping $\mathcal{M}(\sigma) : \mathbb{N} \to \mathbb{R}$. Moreover, for all $t \in \mathbb{N}$, $\mathcal{M}(\sigma)(t)$ is independent of all $\sigma(i)$'s for $i > t$. We can also view $\mathcal{M}(\sigma)$ as a point in $\mathbb{R}^{\mathbb{N}}$. When there is no risk of ambiguity on the stream σ in question, we drop the dependence on σ and use $\mathcal{M}(t)$ to mean $\mathcal{M}(\sigma)(t)$.*

Definition 3 (Time-bounded Mechanism). *A counting mechanism $\mathcal{M}$ is unbounded, if it accepts streams of indefinite lengths, i.e., given any stream σ, $\mathcal{M}(\sigma) \in \mathbb{R}^{\mathbb{N}}$. Given $T \in \mathbb{N}$, a mechanism $\mathcal{M}$ is T-bounded if it only accepts streams of lengths at most T and returns $\mathcal{M}(\sigma) \in \mathbb{R}^T$. In other words, the mechanism needs to know the value T in advance and only looks at the length T prefix of any given stream.*

We would like the mechanism to be useful, that is, its output should well approximate the true count at any point of time. We formally define the notion of utility below.

Definition 4 (Utility). *A counting mechanism $\mathcal{M}$ is (λ, δ)-useful at time t, if for any stream σ, with probability at least $1 - \delta$, we have $|c_\sigma(t) - \mathcal{M}(\sigma)(t)| \leq \lambda$. Note that λ may be a function of δ and t.*

Intuitively, a mechanism is differentially private if it cannot be used to distinguish two streams that are almost the same. In other words, an adversary is unable to determine whether an event of interest took place or not by observing the output of the mechanism over time. For example, the adversary is unable to determine whether a user purchased an item at some time t.

Definition 5 (Differential Privacy). *Two streams σ and σ' are adjacent if they differ at exactly one time t. A counting mechanism $\mathcal{M}$ is ϵ-differentially private (or preserves ϵ-differential privacy) if for any adjacent streams σ and σ', and any measurable subset $S \subseteq \mathbb{R}^{\mathbb{N}}$ (or $S \subseteq \mathbb{R}^T$ for T-bounded mechanisms),* $\Pr[\mathcal{M}(\sigma) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(\sigma') \in S]$.

2.2 Tools

In the design of differentially private mechanisms, the Laplace distribution is often used to introduce random noise [4, 8]. We use $\mathsf{Lap}(b)$ to denote the Laplace distribution with mean 0 and variance $2b^2$. Its probability density function is $x \mapsto \frac{1}{2b} \exp(-\frac{|x|}{b})$.

Dwork *et.al.* showed that if we mask the true answer of a query with Laplacian noise proportional to the sensitivity of the query function, such a mechanism preserves differential privacy for static databases [4, 8]. This is stated formally in the Fact 1.

Fact 1 (Laplace Distribution Maintains Differential Privacy.) *Let $a, b \in \mathbb{R}$ and $|a - b| \leq \Delta$. Let $\gamma \sim \mathsf{Lap}(\frac{\Delta}{\epsilon})$ be a random variable having Laplace distribution. Then, for any measurable subset $S \subseteq \mathbb{R}$,* $\Pr[a + \gamma \in S] \leq \exp(\epsilon) \cdot \Pr[b + \gamma \in S]$.

In the constructions that we propose, the noise may not come from a single Laplace distribution, but rather is the sum of multiple independent Laplace distributions. We now derive a property of the sum of independent Laplace distributions.

Lemma 1 (Sum of Independent Laplace Distributions). *Suppose γ_i's are independent random variables, where each γ_i has Laplace distribution $\mathsf{Lap}(b_i)$. Suppose $Y := \sum_i \gamma_i$, and $b_M := \max_i b_i$. Let $\nu \geq \sqrt{\sum_i b_i^2}$ and $0 < \lambda < \frac{2\nu^2}{b_M}$. Then, $\Pr[Y > \lambda] \leq \exp(-\frac{\lambda^2}{8\nu^2})$.*

The proof to Lemma 1 is a Chernoff-like argument using moment generating functions. We provide the detailed proof in the full version [18].

Corollary 1 (Measure Concentration). *Let Y, ν, $\{b_i\}_i$ and b_M be defined as in Lemma 1. Suppose $0 < \delta < 1$ and $\nu > \max\{\sqrt{\sum_i b_i^2}, b_M\sqrt{2\ln\frac{2}{\delta}}\}$. Then, $\Pr[|Y| > \nu\sqrt{8\ln\frac{2}{\delta}}] \leq \delta$.*

To simplify our presentation and improve readability, we choose $\nu := \sqrt{\sum_i b_i^2} \cdot \sqrt{2\ln\frac{2}{\delta}}$ and use the following slightly weaker result: with probability at least $1-\delta$, the quantity $|Y|$ is at most $O(\sqrt{\sum_i b_i^2}\log\frac{1}{\delta})$.

3 Time-Bounded Counting Mechanisms

In this section, we describe mechanisms that require a priori knowledge of an upper bound on time. In Section 4, we show how to remove this requirement, and achieve unbounded counting mechanisms.

3.1 Simple Counting Mechanisms

To aid the understanding of our contributions and techniques, we first explain two simple constructions.

Simple Counting Mechanism I. The mechanism is given a stream $\sigma \in \{0,1\}^{\mathbb{N}}$, a differential privacy parameter $\epsilon > 0$, and an upper bound T on time. At each time step t, the mechanism samples a fresh independent random variable $\gamma_t \sim \mathsf{Lap}(\frac{1}{\epsilon})$, and releases $\alpha_t = c(t) + \gamma_t$, where $c(t)$ is the true count at time step t. It is not hard to see that the above mechanism is $O(T\epsilon)$-differentially private, and at each time step, the error is $O(\frac{1}{\epsilon})$ with high probability. Alternatively, one can substitute $\epsilon' = \epsilon/T$, and add much bigger noise $\sim \mathsf{Lap}(\frac{1}{\epsilon'})$ at every time step. In this way, we get ϵ differential privacy; however, now the error at each time step is $O(\frac{T}{\epsilon})$.

Simple mechanism I is a straightforward extension of the Laplace mechanism proposed by Dwork *et.al.* [4, 8]. Basically, at every time step, the mechanism answers a new query, and randomizes the answer with fresh independent noise. The down side of this approach is that the privacy loss grows linearly with respect to the number of queries, which is t in our setting.

Simple Counting Mechanism II. In essence, Simple Counting Mechanism II produces a "sanitized" stream by adding independent Laplacian noise to each item in the stream. Suppose the mechanism is given a stream $\sigma \in \{0,1\}^{\mathbb{N}}$ and a differential privacy parameter $\epsilon > 0$. For each time step $t \in \mathbb{N}$, the mechanism samples an independent random variable γ_t with Laplace distribution $\mathsf{Lap}(\frac{1}{\epsilon})$. Define $\alpha_t := \sigma(t) + \gamma_t$. Then, the mechanism $\mathcal{M}$ gives the output $\mathcal{M}(\sigma)(t) := \sum_{i \leq t} \alpha_i$ at time t. A similar idea has been proposed as a survey technique by Warner [19].

It is not hard to see that Simple Mechanism II can be implemented with $O(1)$ words of memory and is unbounded and ϵ-differentially private. We use Corollary 1 to analyze the utility of the mechanism. Fix some time T. Observe that $\mathcal{M}(\sigma)(T) - c_\sigma(T) = \sum_{t \leq T} \gamma_t =: Y$. In this case, all $\gamma_t \sim \mathsf{Lap}(\frac{1}{\epsilon})$. Hence, all $b_t := \frac{1}{\epsilon}$.

Theorem 1. *Let $0 < \delta < 1$, $\epsilon > 0$. the Simple Counting Mechanism II is ϵ-differentially private, and is $(O(\frac{\sqrt{t}}{\epsilon} \cdot \log \frac{1}{\delta}), \delta)$-useful at any time $t \in \mathbb{N}$.*

3.2 Intuition

We will describe the Two-Level Counting Mechanism and the Binary Counting Mechanism. Informally, the Two-Level Mechanism achieves ϵ-differential privacy and $O(t^{\frac{1}{4}})$ error. The Binary Mechanism is a further improvement, and achieves $O((\log t)^{1.5})$ error while maintaining ϵ-differential privacy. We now explain the intuitions for the Two-Level Mechanism and the Binary Mechanism.

A framework for describing mechanisms. We will describe our counting mechanisms using a common framework. Recall that the job of the mechanism is to output an approximate count at every time. However, from now on, we will think of our mechanisms as releasing noisy "p-sums" instead of counts. One can think of p-sums as intermediate results from which an observer can estimate the count at every time step herself.

Definition 6 (p-sum). *A p-sum is a partial sum of consecutive items. Let $1 \leq i \leq j$. We use the notation $\Sigma[i,j] := \sum_{k=i}^{j} \sigma(k)$ to denote a partial sum involving items i through j.*

Furthermore, once we add noise to a p-sum, we obtain a noisy p-sum denoted as $\widehat{\Sigma}$.

The mechanisms we consider will release noisy versions of these p-sums as new items arrive. When an observer sees the sequence of p-sums, she can compute an estimate for the count at each time step, in particular, by summing up an appropriate selection of p-sums. For example, if an observer sees a noisy p-sum $\widehat{\Sigma}[1,k] = \Sigma[1,k] +$ *noise* released at time step k, and another noisy p-sum $\widehat{\Sigma}[k+1,t] = \Sigma[k+1,t] +$*noise* released at time step t, then she can estimate the count at time t by summing up these two noisy p-sums, i.e., $\widehat{\Sigma}[1,k] + \widehat{\Sigma}[k+1,t]$. Notice that the observer needs to be able to do this not only for a specific time t, but also for every time step in $\mathbb{N}$.

Now we rethink Simple Mechanism I using this framework. The noisy p-sums released are noisy versions of the true count for each time step, that is, $\{\widehat{\Sigma}[1,t] = \Sigma[1,t] + \textit{noise}\}_{1 \leq t \leq T}$, where $\Sigma[1,t] = c(t)$ is the true count at time t. In this case, the $\widehat{\Sigma}[1,t]$ itself is the estimated count at time t; and therefore can be regarded as a sum of noisy p-sums (with only one summand). Notice that each item $\sigma(t)$ appears in $O(T)$ of these p-sums. This means that when you flip an item in the incoming stream, $O(T)$ of these p-sums will be affected – this is the reason why the privacy loss is linear in T.

Now consider Simple Mechanism II. The noisy p-sums released are noisy versions of each item $\widehat{\Sigma}_t = \Sigma[t,t] +$ *noise*, where $\Sigma[t,t] = \sigma(t)$ is the t-th item itself. In this

Mechanism	**Each item appears in ? p-sums**	**Each count is the sum of ? p-sums**	**Asymptotic error** (while maintaining ϵ diff. priv.)
Simple I	$O(T)$	$O(1)$	$O(T)$
Simple II	$O(1)$	$O(T)$	$O(\sqrt{T})$
Two-Level	$O(1)$	$O(\sqrt{T})$	$O(T^{\frac{1}{4}})$
Binary	$O(\log T)$	$O(\log T)$	$O((\log T)^{1.5})$

Table 1. Informal intuition for the Two-Level Mechanism and the Binary Mechanism. For simplicity, we omit the parameters ϵ and δ from the bounds.

case, each item appears in only one p-sum, however, each count is the sum of $O(T)$ p-sums. More specifically, to estimate the count at time t, the observer sums up t noisy p-sums $\widehat{\Sigma}_1, \ldots \widehat{\Sigma}_t$. As each noisy p-sum contains some fresh independent noise, the noises add up. In fact, over t time steps, the error would be $O(\sqrt{t})$ with high probability.

Observation 1 *(Informal.) Suppose a mechanism $\mathcal{M}$ adds* $\mathsf{Lap}(\frac{1}{\epsilon})$ *noise to every p-sum before releasing it. In $\mathcal{M}$, each item in the stream appears in at most x p-sums, and each estimated count is the sum of at most y p-sums. Then, the mechanism $\mathcal{M}$ achieves $x \cdot \epsilon$ differential privacy. Moreover, from Corollary 1, the error is $O(\frac{\sqrt{y}}{\epsilon})$ with high probability. Alternatively, to achieve ϵ-differential privacy, one can scale appropriately by having $\epsilon' = \frac{\epsilon}{x}$. Now if the mechanism instead adds* $\mathsf{Lap}(\frac{1}{\epsilon'})$ *noise to each p-sum, we achieve ϵ-differential privacy, and $O(\frac{x\sqrt{y}}{\epsilon})$ error with high probability.*

Goal. From the above analysis, it appears that an inherent tension exists between utility (i.e., small error) and privacy, and our challenge is how to strike a balance between the two conflicting goals. We would like to achieve the following goals.

– *Each item appears in a small number of p-sums* . Intuitively, this limits the influence of any item and guarantees small privacy loss. More specifically, when one flips an item in the incoming stream, not too many p-sums will be affected.
– *Each count is a sum of a small number of p-sums* . Each noisy p-sum contains some noise, and the noises add up as one sums up several noisy p-sums. If each output count is the sum of a small number of noisy p-sums, the accumulation of noises is bounded. In this way, we can achieve small error.

3.3 Two-level Counting Mechanism

We can use the p-sum idea to construct a Two-level Mechanism. As this is not our main construction, we give an overview of the Two-level mechanism below and leave details of the construction to the full version [18]. The basic idea is to release two types of noisy p-sums: 1) a noisy p-sum for every item, that is, $\widehat{\Sigma}[1,1]$, $\widehat{\Sigma}[2,2]$, ..., $\widehat{\Sigma}[T,T]$ where T is an upper bound on the number of time steps; and 2) a noisy p-sum for each contiguous block of $B = \Theta(\sqrt{T})$ items, namely, $\widehat{\Sigma}[kB{+}1, (k{+}1)B]$ for $k \geq 0$. It is not hard to see that each item appears in only 2 p-sums, and each count can be expressed as the sum of $O(\sqrt{T})$ p-sums. According to Observation 1, such a mechanism is 2ϵ-differentially private, and achieves $O(T^{\frac{1}{4}}/\epsilon)$ error with high probability.

3.4 Binary Counting Mechanism

We could extend the idea of the Two-Level Counting Mechanism to a Multi-level Counting Mechanism, and compute the optimal number of levels given an upper bound on time. However, we take a better approach called the Binary Mechanism. The idea is that at any time t, the counting mechanism internally groups the items that have arrived to form $\mathsf{p\text{-}sums}$ of different sizes. The precise grouping of the items depends on the binary representation of the number t – hence the name Binary Mechanism

Given any number $t \in N$, let $\mathsf{Bin}_i(t) \in \{0,1\}$ be the ith digit in the binary representation of t, where $\mathsf{Bin}_0(t)$ is the least significant digit. Hence, $t = \sum_i \mathsf{Bin}_i(t) \cdot 2^i$. Informally, if $\mathsf{Bin}_i(t) = 1$, then there is a $\mathsf{p\text{-}sum}$ involving 2^i items. We formally describe the Binary Mechanism in Figure 1.

```
**Input**: A time upper bound $T$, a privacy parameter $\epsilon$, and a stream $\sigma \in \{0,1\}^T$.
**Output**: At each time step $t$, output estimate $\mathcal{B}(t)$.
**Initialization:** Each $\alpha_i$ and $\widehat{\alpha_i}$ are (implicitly) initialized to 0.
$\epsilon' \leftarrow \epsilon / \log T$
for $t \leftarrow 1$ to $T$ do
    Express $t$ in binary form: $t = \sum_j \mathsf{Bin}_j(t) \cdot 2^j$.
    Let $i := \min\{j : \mathsf{Bin}_j(t) \neq 0\}$.
        $\alpha_i \leftarrow \sum_{j<i} \alpha_j + \sigma(t)$        (1)
    // previous value (if any) of $\alpha_i$ is overwritten
    // $\alpha_i = \Sigma[t - 2^i + 1, t]$ is a p-sum of involving $2^i$ items
    for $j \leftarrow 0$ to $i - 1$ do
        $\alpha_j \leftarrow 0, \quad \widehat{\alpha_j} \leftarrow 0$
    end
        $\widehat{\alpha_i} \leftarrow \alpha_i + \mathsf{Lap}(\frac{1}{\epsilon'})$        (2)
    // $\widehat{\alpha_i}$ is the noisy p-sum $\widehat{\Sigma}[t - 2^i + 1, t]$
    Output the estimate at time $t$:
        $\mathcal{B}(t) \leftarrow \sum_{j:\mathsf{Bin}_j(t)=1} \widehat{\alpha_j}$        (3)
end
```

Figure 1: Binary Mechanism $\mathcal{B}$

Binary mechanism: the $\mathsf{p\text{-}sum}$ *view.* The best way to understand the Binary Mechanism is to think in terms of the $\mathsf{p\text{-}sum}$ framework described earlier. Basically, instead of outputting the estimated counts, the mechanism could equivalently release a sequence of noisy $\mathsf{p\text{-}sums}$ which provide sufficient information for an observer to estimate the count at each time step t. In particular, at any time t, the Binary Mechanism "releases" a new noisy $\mathsf{p\text{-}sum}$ $\widehat{\Sigma}[t-2^i+1,t]$ of length 2^i and ending at position t, where i denotes the position of the least significant non-zero bit in the binary representation of t. This $\mathsf{p\text{-}sum}$ and its noisy version are (temporarily) saved in the variables α_i and $\widehat{\alpha}_i$.

It remains to specify how to estimate the count at each time step from previously "released" noisy $\mathsf{p\text{-}sums}$. Let $i_1 < i_2 < \ldots < i_m$ denote the positions of non-zero bits in the binary representation of t. It is not hard to see that the count at time t is the sum of m $\mathsf{p\text{-}sums}$ of size $2^{i_1}, 2^{i_2}, \ldots, 2^{i_m}$ respectively. They correspond to the values of variables $\alpha_{i_1}, \ldots, \alpha_{i_m}$ maintained by the mechanism at time t. Specifically, the $\mathsf{p\text{-}sums}$ are 1) α_{i_1}, of the most recent 2^{i_1} items; 2) α_{i_2}, of the preceding 2^{i_2} items; 3) α_{i_3}, of the further preceding 2^{i_3} items and so on.

In the full version [18], we show that the Binary Mechanism can be implemented with small memory, as the mechanism can reuse the variables α's and $\widehat{\alpha}$'s.

Differential Privacy. Consider an item arriving at $t \in [T]$. We analyze which of the $\mathsf{p\text{-}sums}$ would be affected if $\sigma(t)$ is flipped. It is not hard to see that the item $\sigma(t)$ can be in at most $\log T$ $\mathsf{p\text{-}sums}$. In particular, it can be in at most 1 $\mathsf{p\text{-}sum}$ of size 2^j, where $j \leq \log T$. Observe that each noisy $\mathsf{p\text{-}sum}$ maintains $\frac{\epsilon}{\log T}$-differential privacy by Fact 1. Hence, we can conclude the ϵ-differential privacy of the Binary Mechanism.

Theorem 2 (Differential Privacy). *For $T \in \mathbb{N}$, the Binary Mechanism preserves T-bounded ϵ-differential privacy.*

Utility. We next consider the usefulness of the Binary Mechanism. Each estimated count $\mathcal{B}(t)$ is the sum of at most $\log T$ noisy $\mathsf{p\text{-}sums}$, and each noisy $\mathsf{p\text{-}sum}$ contains fresh, independent Laplace noise $\mathsf{Lap}(\frac{\log T}{\epsilon})$. Therefore, the error at time t is the summation of at most $O(\log t)$ i.i.d. Laplace distributions $\mathsf{Lap}(\frac{\log T}{\epsilon})$. We use the Corollary 1 to conclude the mechanism's usefulness.

Theorem 3 (Utility). *For each $t \in [T]$, the T-bounded Binary Mechanism is $(O(\frac{1}{\epsilon}) \cdot (\log T) \cdot \sqrt{\log t} \cdot \log \frac{1}{\delta}, \delta)$-useful at time $t \in [T]$.*

4 Unbounded Counting Mechanisms

Previously, we mainly considered time-bounded mechanisms, i.e., the mechanism requires a priori knowledge of an upper bound on the time. We now describe how to remove this assumption and derive unbounded counting mechanisms.

We describe the Hybrid Mechanism which gives a generic way to convert any time-bounded mechanism $\mathcal{M}$ into an unbounded one by running two mechanisms in parallel: (1) an unbounded mechanism that only outputs counts at time steps t being powers of 2; (2) a time-bounded mechanism $\mathcal{M}$ to take care of items arriving at time steps between successive powers of 2.

Logarithmic Counting Mechanism. First, consider an unbounded mechanism which only reports the estimated counts at sparse intervals, in particular, when t is a power of 2. We would expect such a mechanism to have better guarantees than one that has to report at every time step.

So what do we do when t is not a power of 2? We know the approximate count $\widehat{c}_1$ for the time period $[1, T]$ where $T = 2^k$ for some non-negative integer k. Suppose we also know the approximate count $\widehat{c}_2$ for the time period $[T + 1, t]$ where $T + 1 \leq t \leq 2T$. Then we can estimate the count at time t as $\widehat{c}_1 + \widehat{c}_2$. Therefore, it remains for us to count the 1's between $[T, t]$ for any $t \in [T + 1, 2T]$, We can simply apply a T-bounded mechanism (e.g., the Binary Mechanism) for this task.

We now design an unbounded mechanism called the *Logarithmic Mechanism* which reports the count only when the time t is a power of 2.

```
Input: Differential privacy parameter $\epsilon$, and a stream $\sigma \in \{0,1\}^{\mathbb{N}}$.
Output: $\forall k \in \mathbb{Z}$, at time $t = 2^k$, output estimate $\mathcal{L}(t)$.
Initialization: $\beta \leftarrow 0$.
foreach $t \in \mathbb{N}$ do
    $\beta \leftarrow \beta + \sigma(t)$
    if $t = 2^k$ for some $k \in \mathbb{Z}$ then
        $\beta \leftarrow \beta + \mathsf{Lap}(\frac{1}{\epsilon})$
        Output $\mathcal{L}(t) \leftarrow \beta$
    end
end
```

Figure 2: Logarithmic Mechanism $\mathcal{L}$

The idea for the Logarithmic Mechanism is quite simple. The mechanism internally keeps a value β which is initialized to 0. β is used to keep track of the approximate count at any point of time. As an item comes in, its value is added to β. At t equal to a power of 2, the mechanism adds fresh randomness to the value β (on top of randomness previously added), and outputs β.

If t is a power of 2, it is clear that the accumulated error at time t is a sum of $O(\log t)$ independent Laplace distributions $\mathsf{Lap}(\frac{1}{\epsilon})$. Hence, we have the following guarantee from Corollary 1.

Theorem 4. *The Logarithmic Counting Mechanism is unbounded, preserves ϵ-differential privacy and is $(O(\frac{1}{\epsilon}) \cdot \sqrt{\log t} \cdot \log \frac{1}{\delta}, \delta)$-useful at time $t = 2^k$ for some $k \geq 0$.*

Logarithmic Mechanism: the p-sum *view.* The Logarithmic Mechanism also has a p-sum interpretation. Equivalently, one can think of it as releasing the noisy p-sums $\widehat{\alpha}_0 = \widehat{\Sigma}[1, 1]$, as well as $\widehat{\alpha}_k = \widehat{\Sigma}[2^{k-1} + 1, 2^k]$ for every $k \geq 1$, Now an observer can estimate the count at time $t = 2^k$ as $\sum_{i=0}^{k} \widehat{\alpha}_i$.

Hybrid Mechanism. We combine the Logarithmic Mechanism and a time-bounded counting mechanism to process a given stream σ. We run one copy of $\frac{\epsilon}{2}$-differentially private Logarithmic Mechanism, which reports an approximate count when t is a power of 2. Suppose the Logarithmic Mechanism has reported count $\mathcal{L}(T)$ at $T = 2^k$ for some non-negative integer k. For time t in the range $T + 1 \leq t \leq 2T$, we run an $\frac{\epsilon}{2}$-differentially private T-bounded counting mechanism denoted as $\mathcal{M}$ to count the number of 1's in the range $[T + 1, t]$. We write $\tau = t - T$. At time t, let $\mathcal{M}(\tau)$ be the number of 1's in $[T + 1, T + \tau]$ reported by the T-bounded counting mechanism $\mathcal{M}$. Then, the hybrid mechanism reports $\mathcal{L}(T) + \mathcal{M}(\tau)$ at time t.

```
Input: Differential privacy parameter $\epsilon$, a stream $\sigma \in \{0,1\}^{\mathbb{N}}$, Logarithmic
       Mechanism $\mathcal{L}$, and a time-bounded mechanism $\mathcal{M}$.
Output: For each $t \in \mathbb{N}$, output estimate $\mathcal{H}(t)$.
Initialization: $T \leftarrow 1$.
Initiate the mechanism $\mathcal{L}$ with privacy parameter $\frac{\epsilon}{2}$ on stream $\sigma$.
foreach $t \in \mathbb{N}$ do
    Feed $\sigma(t)$ to mechanism $\mathcal{L}$.
    if $t = 2^k$ for some $k \in \mathbb{Z}$ then
        Output $\mathcal{H}(t) \leftarrow \mathcal{L}(t)$
        $T \leftarrow t$
        Initiate an instance of the $T$-bounded mechanism $\mathcal{M}_T$ with time upper
        bound $T$, privacy parameter $\frac{\epsilon}{2}$ and stream $\sigma^{(T)} \in \{0,1\}^T$, where
        $\sigma^{(T)}(\tau) := \sigma(\tau + T)$ for $\tau \in [1, T]$.
    else
        $\tau \leftarrow t - T$
        Feed $\sigma^{(T)}(\tau) := \sigma(t)$ to mechanism $\mathcal{M}_T$.
        Output $\mathcal{H}(t) \leftarrow \mathcal{L}(T) + \mathcal{M}_T(\tau)$
    end
end
// At time $t$, $T$ is the largest power of 2 no bigger than $t$.
// $\sigma^{(T)}$ is the sub-stream of $\sigma$ for the duration $[T+1, 2T]$.
// $\mathcal{M}_T$ is a time-bounded mechanism that runs for $[T+1, 2T]$.
```

Figure 3: Hybrid Mechanism $\mathcal{H}$ (with Mechanism $\mathcal{M}$).

Theorem 5. *Assume that given any $\epsilon > 0$ and $0 < \delta < 1$, Logarithmic Mechanism $\mathcal{L}$ is ϵ-differentially private and is $(f(\epsilon, t, \delta), \delta)$-useful at time t. Similarly, assume that given any $\epsilon > 0$, the T-bounded mechanism $\mathcal{M}$ is ϵ-differentially private and is $(g(\epsilon, T, \tau, \delta), \frac{\delta}{2})$-useful at time $\tau \in [T]$, where g is monotonically increasing with T and τ. Then, the Hybrid Mechanism described above is unbounded, preserves ϵ-differential privacy, and is $(f(\frac{\epsilon}{2}, t, \frac{\delta}{2}) + g(\frac{\epsilon}{2}, t, t, \frac{\delta}{2}), \delta)$-useful at time t.*

The proof of Theorem 5 can be found in the full version [18].

Corollary 2. *If we instantiate the Hybrid Mechanism using the Binary Mechanism as the T-bounded mechanism, the resulting Hybrid Mechanism is unbounded, preserves ϵ-differential privacy, and is $(O(\frac{1}{\epsilon}) \cdot (\log t)^{1.5} \cdot \log \frac{1}{\delta}, \delta)$-useful at time t.*

For simplicity, in the remainder of the paper, when we refer to the Hybrid Mechanism, we mean the Hybrid Mechanism instantiated with the Binary Mechanism.

Hybrid Mechanism: the p-sum view. One can also interpret the Hybrid Mechanism naturally using the p-sum framework. Basically, one can equivalently think of the Hybrid Mechanism as releasing the union of the noisy p-sums of the Logarithmic Mechanism and the Binary Mechanism. From this set of noisy p-sums, an observer can compute the approximate count at every time step $t \in \mathbb{N}$.

5 Pan Privacy

Dwork *et.al.* formalized the notion of pan privacy [6, 10] to deal with intruders who can observe snapshots of the mechanism's internal states, e.g., in a subpoena. For mechanisms in the p-sum framework, we can apply techniques similar to those proposed in the work [6] to make them pan private against a single unannounced intrusion, with only a constant-factor loss in the error term. Furthermore, we extend the idea to construct pan private counting mechanisms resilient against multiple announced (afterwards) intrusions. The error scales with a square root factor on the number of intrusions made. We provide detailed definitions of pan privacy and constructions in the full version [18].

References

[1] J. A. Calandrino, A. Narayanan, E. W. Felten, and V. Shmatikov. Don't review that book: Privacy risks of collaborative filtering. Manuscript, 2009.

[2] E. D. Demaine, A. López-Ortiz, and J. I. Munro. Frequency estimation of internet packet streams with limited space. In *ESA*, 2002.

[3] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, 2003.

[4] C. Dwork. Differential privacy. Invited talk at *ICALP*, 2006.

[5] C. Dwork. The differential privacy frontier. In *TCC*, 2009.

[6] C. Dwork. Differential privacy in new settings. Invited presentation at *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[7] C. Dwork. A firm foundation for private data analysis. In *Communications of the ACM*, 2010.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[9] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *STOC*, 2010.

[10] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science (ISC)*, 2010.

[11] C. Dwork and S. Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *CRYPTO*, 2008.

[12] R. Jones, R. Kumar, B. Pang, and A. Tomkins. Vanity fair: privacy in querylog bundles. In *CIKM*, 2008.

[13] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, 2009.

[14] G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In *VLDB*, 2002.

[15] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the netflix prize contenders. In *KDD*, 2009.

[16] A. Metwally, D. Agrawal, and A. E. Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT*, 2005.

[17] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[18] D. S. T-H. Hubert Chan, Elaine Shi. Private and continual release of statistics. Full version avalaible online at `http://eprint.iacr.org/2010/076.pdf`, 2010.

[19] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 1965.