Differentially Private Algorithms for k-Anonymity Server

Alessandro Epasto <aepasto@google.com>,
Kevin Graney <kmg@google.com>,
Jieming Mao <maojm@google.com>,
Andres Munoz Medina <ammedina@google.com>,
Martin Pál <mpal@google.com>,
Miroslava Sotakova<mirka@google.com>,

January 2023

1 Introduction

In this note, we describe a differentially private algorithm for the FLEDGE k-anonymity server which we call AboveThresholdWithPeriodicRestart. We provide theoretical guarantees on their differential privacy and utilities. This note focuses on one particular interest group and an instance of the algorithm described here is applied to each interest group independently.

The server for the interest updates per update frequency $p = 1$ hour. We discretize the time horizon into T virtual steps, each representing a period of an hour. There are n user join events for the interest group. Each user join event i is represented by $Join(u_i, t_i)$, in which u_i represents the user and t_i represents the at which step this event happens. T and n can be very big to cover the whole period of the server run.

The goal is to differentially privately release in steps $0, 1, ..., T - 1$ whether the number of distinct users in the past w steps is above threshold k for the interest group.

Our algorithm uses the AboveThreshold algorithm [1] as a main component. The AboveThreshold algorithm is a well-known differentially private algorithm in the continual release setting. It keeps answering threshold queries until the answer is "above" and halts. It can be easily transformed into an algorithm which does not halt but produces monotone outputs (i.e. always outputting "above" after starting to output "above"). We design a periodic restart algorithm to periodically start new AboveTreshold algorithm instances.

This algorithm is also related to our research on the more general problem of differentially private estimation of counting queries in continual release (we refer to [2] for more information).

This note is organized as following. In Section 2, we give formal definitions of the problem's inputs and outputs, and the differential privacy guarantee. In Section 3, we describe the periodic restart algorithm. In Section 4, we include the AboveThreshold algorithm for completeness (described in our setup) and we extend the analysis of it to work with truncated Laplace noise.

2 Preliminaries

In this note, we consider T virtual steps: $0, 1, 2, ..., T-1$. They are abstracted from the hourly updates. And we focus on algorithm design and analysis for a single interest group.

2.1 Input datasets (raw and aggregated)

The raw input dataset S_{raw} consists of a set of joins in the form of $\{Join(u_i, t_i)\}_i$ representing that user u_i joins at step t_i.

Definition 2.1 (Neighboring Datasets)**.** *We say that two raw input datasets S_{raw} and S'_{raw} are **neighboring** datasets if one of them has one more join than the other one, i.e. $S_{raw} = S'_{raw} \cup Join(u,t)$ or $S'_{raw} = S_{raw} \cup Join(u,t)$ for some u and t.*

For each raw input dataset S_{raw}, we define the aggregated input dataset $S_{agg} = AGG(S_{raw})$ as a function which maps a starting step i and an ending step j for $i < j$ and $i, j \in \{0, 1, ..., T-1\}$, to a count. In particular, $S_{agg}[i,j]$ represents the number of distinct users with Join between step i and step j in S_{raw}. Here $AGG(\cdot)$ is the procedure of converting a raw dataset to an aggregated dataset. Our algorithm runs on aggregated datasets. For notation convenience in the rest of the doc, we set $S_{agg}[i,j] = S_{agg}[0,j]$ when $i < 0$.

Observation 2.2. *If S_{raw} and S'_{raw} are neighboring datasets ($S_{raw} = S'_{raw} \cup Join(u,t)$), and S_{agg} and S'_{agg} are the corresponding aggregated datasets (i.e. $S_{agg} = AGG(S_{raw})$ and $S'_{agg} = AGG(S'_{raw})$), we have that for any $i < j$, $i, j \in \{0, 1, ..., T-1\}$,*

- *If $t \in [i,j]$, $S_{agg}[i,j] - S'_{agg}[i,j] \in \{0, 1\}$.*
- *If $t \notin [i,j]$, $S_{agg}[i,j] = S'_{agg}[i,j]$.*

2.2 Algorithm Output

In this note, we consider a window size $w \in \mathbb{N}$ and a threshold $k \in \mathbb{N}$. For input S_{raw} and its corresponding aggregated input S_{agg}, the algorithm outputs o with T booleans such that o_t approximates whether $S_{agg}[t-w+1, t] \geq k$.

2.3 Differential Privacy

We use the standard definition of differential privacy based on our definition of neighboring datasets.

Definition 2.3 (Differential Privacy)**.** *We say that an algorithm Alg is (ε, δ)-differential privacy ((ε, δ)-DP) if for any neighboring datasets S_{raw}, S'_{raw} and any output set $\mathcal{O}$,*

$$\Pr\left[Alg(AGG(S_{raw})) \in \mathcal{O}\right] \leq e^{\varepsilon} \cdot \Pr\left[Alg(AGG(S'_{raw})) \in \mathcal{O}\right] + \delta.$$

3 Above Threshold with Periodic Restart Algorithm

We state our main algorithm which uses AboveThreshold (details in Section 4) as sub-routines.

```
Algorithm 1: AboveThresholdWithPeriodicRestart Algorithm
  Input: Number of steps $T$, window size $w$, aggregate dataset $S_{agg}$, target threshold $k$, noise
         setting $\rho_1$ and $\rho_2$
  Output: Boolean vector $o$ for steps $0, 1, ..., T-1$

  for $t = 0$ to $T - 1$ do
      if $t \equiv 0 \pmod{w}$ then
          Start an instance of the AboveThreshold algorithm $AT_t$ (see Algorithm 2).
      Output $o_t$ to be the output of $AT_{\lceil t/w \rceil \cdot w}$ on step $t$.
  end
  return $o$;
```

Corollary 3.1. *If each AboveThreshold algorithm is (ε, δ)-DP, the AboveThresholdWithPeriodicRestart algorithm is $(2\varepsilon, 2\delta)$-DP.*

Proof. It is easy to check that AT_s only uses $S_{agg}[i, j]$ for $i, j \in [s - w + 1, s + w - 1]$. So for any $Join(u, t)$, it only affects inputs used in at most two AboveThreshold instances. By Observation 2.2, together with simple composition of differential privacy, we know that the Periodic Restart Algorithm is $(2\varepsilon, 2\delta)$-DP. □

4 AboveThreshold Algorithm

We apply the AboveThrehold algorithm [1] in our scenario as a main sub-routine of our algorithm.

Algorithm 2: AboveThreshold Algorithm AT_s

```
Input: Number of steps $T$, window size $w$, aggregate dataset $S_{agg}$, target threshold $k$, noise
       setting $\rho_1$ and $\rho_2$, and starting step $s$.
Output: Boolean vector $o$ for steps $s, s+1, ..., s+w-1$

Sample noise $\nu$ according to $\rho_1$;
$k' \leftarrow k + \nu$ ;
AlreadyAbove $\leftarrow false$ ;
for $t = s$ to $\min(s+w-1, T-1)$ do
    if AlreadyAbove then
        Output $o_t = true$;
    else
        Sample noise $\nu_t$ according to $\rho_2$;
        $c_t \leftarrow S_{agg}[t-w+1, t]$;
        $c'_t \leftarrow c_t + \nu_i$;
        if $c'_t \geq k'$ then
            Output $o_t = true$;
            AlreadyAbove $\leftarrow true$;
        else
            Output $o_t = false$;
        end
    end
end
return $o$;
```

The proof in [1] sets ρ_1 and ρ_2 to be Laplace noise. Here we extend its result to a general scenario which includes truncated Laplace noise and Laplace noise as sub-cases.

Definition 4.1. *We say that a noise with pdf f is bound$(\varepsilon, \delta, A, d)$, if for all $x \in \mathbb{R}$, and all integers $i \in \{-d, -d+1, ..., d-1, d\}$, and any measurable set S,*

$$\int_{x \in S} f(x)dx \leq \delta + e^{\varepsilon} \int_{x \in S} f(x+i)dx$$

and

$$\int_{-A}^{A} f(x)dx = 1.$$

Theorem 4.2. *If we set ρ_1 to be bound$(\varepsilon_1, \delta_1, A_1, 1)$ and ρ_2 to be bound$(\varepsilon_2, \delta_2, A_2, 1)$, and let $\varepsilon = \varepsilon_1 + \varepsilon_2$, $\delta = \delta_1 + \delta_2$, and $A = A_1 + A_2$, we have that AT_s (Algorithm 2) is $(\varepsilon, \delta \cdot (w+1))$-DP. And for all t such that $o_t = true$, $S_{agg}[t-w+1, t] \geq k - A$ and for all t such that $o_t = false$, $S_{agg}[t-w+1, t] \leq k + A$.*

Proof. The error bound simply follows from the fact that $|\nu| + \max_{i=s}^{s+w-1} |\nu_i| < A_1 + A_2 = A$. In the rest of the proof, we focus on proving the algorithm is DP.

Consider two neighboring datasets S_{raw} and S'_{raw} and their corresponding aggregated datasets S_{agg} and S'_{agg}. Let O and O' to be the random variables denoting the algorithm's outputs on S_{agg} and S'_{agg}. And consider some specific output $o = \underbrace{false, ..., false}_{\ell}, \underbrace{true, ..., true}_{w-\ell}$ for any $\ell \in \{0, 1, 2, ..., w-1\}$. We are going to compare the probability of O and O' being o, conditioned on the noise $\nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}$. For notation convenience, here $\nu_{[i,j]}$ represents $(\nu_i, ..., \nu_j)$ and similarly for $u_{[i,j]}$.

$$
\begin{aligned}
&\Pr[O = o | \nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}] \\
=& \Pr\left[\max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i) < k + \nu \text{ and } c_{s+\ell}(S_{agg}) + \nu_{s+\ell} \geq k + \nu \;\;\middle|\;\; \nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}\right] \\
=& \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pr[\nu = u] \cdot \Pr[\nu_{s+\ell} = u_{s+\ell}] \cdot \\
&\qquad \mathbf{1}\left\{\max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + u_i) - k < u < c_{s+\ell}(S_{agg}) + u_{s+\ell} - k\right\} \cdot du \cdot du_{s+\ell} \\
=& \int_{u > \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + u_i) - k} \int_{u_{s+\ell} > u - c_{s+\ell}(S_{agg}) + k} \Pr[\nu = u] \cdot \Pr[\nu_{s+\ell} = u_{s+\ell}] \cdot du \cdot du_{s+\ell}
\end{aligned}
$$

We also have

$$
\begin{aligned}
&\Pr[O' = o | \nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}] \\
=& \int_{u > \max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + u_i\right) - k} \int_{u_{s+\ell} > u - c_{s+\ell}(S'_{agg}) + k} \Pr[\nu = u] \cdot \Pr[\nu_{s+\ell} = u_{s+\ell}] \cdot du \cdot du_{s+\ell} \\
=& \int_{u > \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + u_i) - k} \int_{u_{s+\ell} > u - c_{s+\ell}(S_{agg}) + k} \Pr\left[\nu = u + \max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + \nu_i\right) - \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i)\right] \cdot \\
&\qquad \Pr\left[\nu_{s+\ell} = u_{s+\ell} + \max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + \nu_i\right) - \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i) + c_{s+\ell}(S_{agg}) - c_{s+\ell}(S'_{agg})\right] \cdot du \cdot du_{s+\ell}
\end{aligned}
$$

By Observation 2.2, we know $\max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + \nu_i\right) - \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i) \in \{-1, 0, 1\}$, and $\max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + \nu_i\right) - \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i) + c_{s+\ell}(S_{agg}) - c_{s+\ell}(S'_{agg}) \in \{-1, 0, 1\}$.

The last bound comes the fact that if S'_{agg} is obtained from removing an entry in S_{agg}, the term is $\{-1, 0\}$ + $\{1, 0\}$ which is in $\{-1, 0, 1\}$ (the addition case is specular).

Therefore, by the definition of noise ρ_2,

$$
\begin{aligned}
&\int_{u_{s+\ell} > u - c_{s+\ell}(S_{agg}) + k} \Pr\left[\nu_{s+\ell} = u_{s+\ell} + \max_{i=s}^{s+\ell-1} \left(c_i(S'_{agg}) + \nu_i\right) - \max_{i=s}^{s+\ell-1} (c_i(S_{agg}) + \nu_i) + c_{s+\ell}(S_{agg}) - c_{s+\ell}(S'_{agg})\right] \\
&\qquad \cdot du_{s+\ell} \\
\leq& \min\left(1, \delta_2 + e^{\varepsilon_2} \cdot \int_{u_{s+\ell} > u - c_{s+\ell}(S_{agg}) + k} \Pr[\nu_{s+\ell} = u_{s+\ell}] \cdot du_{s+\ell}\right).
\end{aligned}
$$

Putting this together with the definition of noise ρ_1, we have

$$\begin{aligned}
&\Pr[O' = o|\nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}]\\
\le & \int_{u>\max_{i=s}^{s+\ell-1}(c_i(S_{agg})+u_i)-k} \Pr\left[\nu = u + \max_{i=s}^{s+\ell-1}\left(c_i(S'_{agg})+\nu_i\right) - \max_{i=s}^{s+\ell-1}\left(c_i(S_{agg})+\nu_i\right)\right]\cdot\\
&\quad \min\left(1, \delta_2 + e^{\varepsilon_2}\cdot \int_{u_{s+\ell}>u-c_{s+\ell}(S_{agg})+k} \Pr[\nu_{s+\ell}=u_{s+\ell}]\cdot du_{s+\ell}\right)\cdot du\\
\le & \delta_2 + \int_{u>\max_{i=s}^{s+\ell-1}(c_i(S_{agg})+u_i)-k} \Pr\left[\nu = u + \max_{i=s}^{s+\ell-1}\left(c_i(S'_{agg})+\nu_i\right) - \max_{i=s}^{s+\ell-1}\left(c_i(S_{agg})+\nu_i\right)\right]\cdot\\
&\quad \min\left(1, e^{\varepsilon_2}\cdot \int_{u_{s+\ell}>u-c_{s+\ell}(S_{agg})+k} \Pr[\nu_{s+\ell}=u_{s+\ell}]\cdot du_{s+\ell}\right)\cdot du\\
\le & \delta_1+\delta_2+e^{\varepsilon_1}\cdot\int_{u>\max_{i=s}^{s+\ell-1}(c_i(S_{agg})+u_i)-k} \Pr[\nu=u]\cdot du\cdot\\
&\quad e^{\varepsilon_2}\cdot\int_{u_{s+\ell}>u-c_{s+\ell}(S_{agg})+k} \Pr[\nu_{s+\ell}=u_{s+\ell}]\cdot du_{s+\ell}.
\end{aligned}$$

Therefore, $\Pr[O' = o|\nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}] \le e^{\varepsilon}\Pr[O = o|\nu_{[s,s+\ell-1]} = u_{[s,s+\ell-1]}] + \delta$, for all $u_{[s,s+\ell-1]}$. To sum up, we get $\Pr[O' = o] \le e^{\varepsilon}\Pr[O = o]+\delta$. The privacy proof for $o = \underbrace{false, ..., false}_{w}$ is similar and simpler, and we omit the proof.

Finally for any output set $\mathcal{O}$, we know $\Pr[O' \in \mathcal{O}] \le e^{\varepsilon}\Pr[O \in \mathcal{O}] + (w + 1)\delta$. □

4.1 Truncated Laplace

In section, we apply truncated Laplace noise from [3]. Define noise $TLap(\varepsilon, \delta)$ according to the following pdf f:

$$f(x) = Be^{-\varepsilon|x|}, \text{ for } x \in [-A, A] \text{ and } 0 \text{ otherwise}$$

where

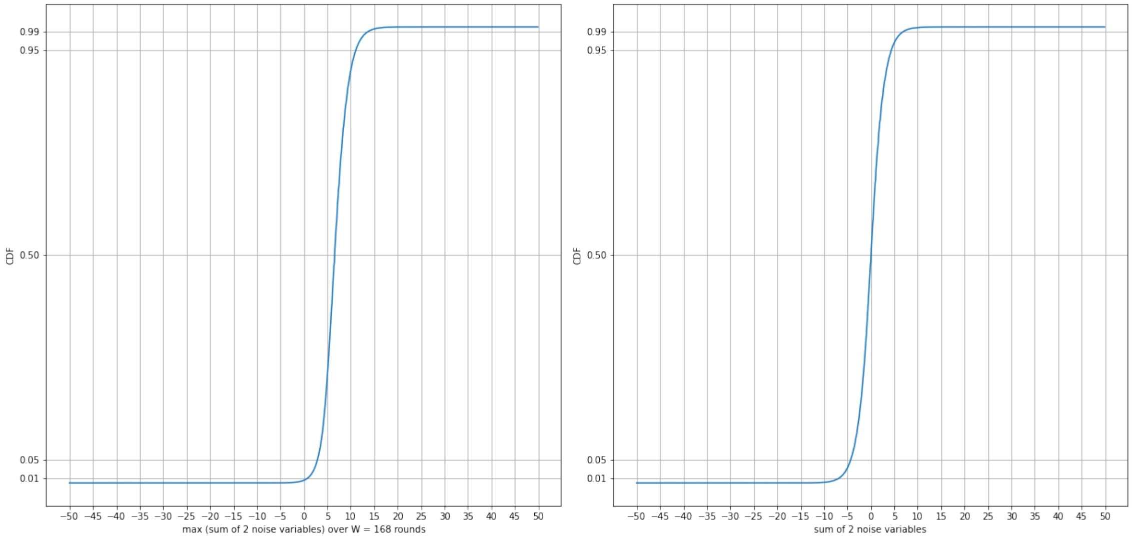
$$\begin{aligned}
A &= \frac{1}{\varepsilon}\cdot\log\left(1+\frac{e^{\varepsilon}-1}{2\delta}\right),\\
B &= \frac{\varepsilon}{2(1-e^{-\varepsilon A})}.
\end{aligned}$$

Theorem 4.3 ([3]). *The above f satisfies, for any measurable set,*

$$\int_{x\in S} f(x)dx \le \delta + e^{\varepsilon}\int_{x\in S} f(x \pm 1)dx.$$

If we apply Theorem 4.2 with truncated Laplace noise, we get the following corollary,

Corollary 4.4. *If we set ρ_1 and ρ_2 to be $TLap(\varepsilon, \delta)$, we have that Algorithm 2 is $(2\varepsilon, 2\delta\cdot(w+1))$-DP. And for all t such that $o_t = true$, $S_{agg}[t - w + 1, t] \ge k - A$ and for all t such that $o_t = false$, $S_{agg}[t - w + 1, t] \le k + A$, for $A = \frac{2}{\varepsilon}\cdot\log\left(1+\frac{(e^{\varepsilon}-1)}{2\delta}\right)$.*



(a) CDF of the max additive error over w steps (false positives)

(b) Additive error in one step (false negatives)

Figure 1: CDF for the additive error (sum of the 2 noise variables) for the parameter setting described in Section 5.

5 Bounds on the accuracy of the algorithm

We observe that the introduction of noise (needed for DP) can cause both false positives (i.e., groups with < k users resulting in a $true$ output) and false negatives (i.e., groups with >= k users resulting in a $false$ output). In this section, we bound mathematically the probability of these events for the sample parameter setting of $p = 1$ hour, $w = 168$ hours, $k = 50$, $A = 25$, and $\epsilon = 3$, $\delta < 10^{-5}$.

5.1 False positives

We first review the false positives. Notice that a false positive happens when a group with $< k$ users is reported as above threshold because of the effect of the noise. This can happen at any update over a window (i.e., in one of the 168 hourly updates of the week). Notice that we have 2 sources of noise: 1) the fixed threshold noise, and 2) the independent noise sampled at each update.

Suppose a interest group has a fixed size of $k - x$ for the entire duration of the window. It is possible to observe that the interest group will result in a false positive, if and only if, at any update time, the sum of both noise variables result in at least x.

To bound this event, we show the CDF of the maximum additive error over the window (i.e. the max of the 168 independent noise variables plus the fixed threshold noise).

The CDF is reported in Figure 1a. It is possible to see that, with 99% probability, the noise is below 15. This means, for this settings, that a group with less than $k - 15$ users will not experience a false positive.

5.2 False negative

A false negative happens, instead, when a group with $\geq k$ users is reported as below threshold. We consider the probability of this error happening at the first update of the window when a group is above threshold (notice that if a group is correctly reported true at the first step, then for the entire window the group will not have a false negative).

As before, suppose a interest group has a size of $k + x$, the group will result in a false negative if and only if the sum of both noise variables is less than $-x$. In Figure 1b, we report the CDF of the sum of the noise variables in one step. We observe that with 99% probability the noise is not smaller than -8. This means that a group with at least than $k + 8$ users will not experience a false negative.

Caveats We observe that this analysis accounts only for the directed effect of the algorithm and assumes exact counting of the interest group size in a given step. Other aspects of the FLEDGE API may result in different sources of errors and delays in updates. For instance the browser's updates may not be report immediately to the server. Moreover, the limits to the number of bits used for user identifiers results in further errors not accounted for in this document.

Acknowledgements

We would like to thank Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong for their comments on this work.

References

[1] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014.

[2] Alessandro Epasto, Jieming Mao, Andres Munoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. In *Proceedings of the 14th Innovations in Theoretical Computer Science (ITCS) conference*, 2023.

[3] Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. Tight analysis of privacy and utility tradeoff in approximate differential privacy. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 89–99. PMLR, 26–28 Aug 2020.