# Secure Noise Generation

Differential Privacy Team
Google

June 2020

**Abstract**

This technical report describes the sampling algorithms implemented in Google's Basic DP Building Blocks Library for Laplace and Gaussian distributions. These algorithms circumvent problems with naïve floating point implementations, first identified by Mironov [Mir12], that can cause differential privacy violations.

## Contents

# 1 Introduction

Theoretical results on differential privacy (DP) [DMNS06, DKM+06] (see [DR14] for an overview and formal definitions of $(\epsilon, \delta)$-Differential Privacy) typically assume precise sampling and computation using real numbers, but in practice we have only finite resolution. Mironov [Mir12] showed how this leads to problems whereby strong theoretical DP guarantees can be catastrophically violated in practice, and gave a practical implementation of the Laplace mechanism that avoids these problems by rounding the result after adding noise. Although this rounding step uses a resolution that is tied to the width of the noise already added, and therefore only multiplies the overall error by a constant factor, the net effect is still a significant reduction in practical utility. This report proposes an alternative approach that gives significantly lower error and can be applied to both the Laplace and Gaussian mechanisms (and perhaps others). This method underlies the implementations in Google's Basic DP Building Blocks Library.

We note a different algorithm for sampling Gaussians was recently suggested by Canonne et al. [CKS20].

# 2 Floating Point Arithmetic

While a full description of the IEEE floating point standard is outside the scope of this report, we briefly highlight the properties that our mechanisms depend on.

Floating point numbers have an exponent and a significand. This works like scientific notation: the significand is multiplied by $2^{\text{exponent}}$ to get the represented value. Double-precision floating point numbers, which we assume throughout the rest of this report, have 52 bits of significand and 11 bits of exponent; this means that they can represent the first $2^{52}$ multiples of any power of two between $2^{-1022}$ and $2^{1023}$. See, e.g., `https://en.wikipedia.org/wiki/Double-precision_floating-point_format` for more details.

A key property that we rely on is that the IEEE floating point standard guarantees that the results of basic arithmetic operations are the same as if the computation was performed exactly and then rounded to the closest floating point number.

# 3 From Continuous to Discrete Distributions

While generating privacy-grade continuous noise is fraught with floating point complexities, noise generated in a finite space (like the bounded integers) is much easier to implement safely. In particular, we can ensure that every element in the space is sampled with a probability close to its theoretically correct probability; for instance, a binary search accumulates only a logarithmic number of (possibly inexact) probability calculations. Although small errors in sampling probabilities will still have a nonzero effect on privacy measurements, this effect is bounded when the errors are bounded—in contrast, naïve floating point samplers for continuous noise may simply never sample large regions of the output space, resulting in catastrophic privacy violations. In this report our focus is mainly on avoiding these types of violations.

Our approach is to therefore to develop an additive noise mechanism that operates on integers as much as possible, allowing us to then rely on simple properties of the IEEE floating point standards to argue durability against numerical attacks. (Note that, while the set of representable floating point numbers is itself a finite space, their large dynamic range and non-uniform spacing make it difficult to use this fact directly.)

**Resolution parameter.** Our algorithms take a *resolution parameter* $r$. It will be important that $r$ is selected according to the scale of the continuous distribution we are simulating, and not in a data-dependent way. In this section, we think of $r$ as being fixed ahead of time.

Let $f$ be a function mapping a database to a real number, and $f_r(D)$ be $f(D)$ rounded to the nearest multiple of $r$. Define $\Delta = \max_{D,D'} ||f(D) - f(D')||$ as the *sensitivity* of $f$, and $\Delta_r = \max_{D,D'} ||f_r(D) - f_r(D')||$ as the *r-sensitivity* of $f$.

**Our approach.** Let Sampler$(\epsilon, \delta, r, \Delta_r)$ denote a subroutine that draws an integer $i$ according to some distribution. Most of this report is dedicated to designing such samplers. In particular, we want the distribution generated by Sampler to approximate a scaled version of the targeted continuous distribution, and for the approximation to improve as the resolution parameter $r$ gets smaller.

For now, we focus on the fact that most of the complexity in simulating numerically secure additive noise can be eliminated by generating integer noise at a specified resolution.

Suppose the following conditions are met:

1. $r = 2^k$, with integer $k$ satisfying $-1022 \leq k \leq 1023$.
2. If $i \sim$ Sampler$(\epsilon, \delta, r, \Delta_r)$ then $f_r(D) + ir$ is $(\epsilon, \delta)$-Differentially Private.
3. $\mathbb{P}[|i| > 2^{52}] < \exp(-1000)$.

Suppose further that the claim that $f_r(D) + ir$ is $(\epsilon, \delta)$-Differentially Private holds under exact real number arithmetic. We argue that it makes little difference if we instead use floating point arithmetic. Indeed, if $|f(D)| \leq r2^{52}$, $i \leq 2^{52}$, and $r$ is a power of two between $2^{-1022}$ and $2^{1023}$, then $f_r(D)$ and $ir$ can be represented exactly as double resolution floating point numbers. Letting $\oplus$ denote floating point addition, the IEEE standard guarantees that $f_r(D) \oplus ir$ is equal to $f_r(D) + ir$ followed by rounding. Since $f_r(D) + ir$ is differentially private, $f_r(D) \oplus ir$ must be too, since differential privacy is preserved under postprocessing.

The event that $|f(D)| > r2^{52}$ can be controlled via the choice of $r$ and, if necessary, by constraining $f$; we will assume this does not occur. The event that $|i| > 2^{52}$ can either be ignored or charged to the $\delta$ guarantee of the mechanism. For simplicity, we choose the former in this report.

# 4 Approximating Laplace

We now describe how to implement Sampler$(\epsilon, 0, r, \Delta_r)$ to approximate the Laplace distribution. We simply require a differentially private discrete analog to the Laplace distribution, and the geometric distribution immediately comes to mind.

We must also select the resolution parameter $r$. We will set the resolution based on the scale of the noise (i.e., sensitivity / $\epsilon$). This is the same thing that happens in Mironov's paper, but rather than setting the resolution to be equal to the scale of the noise, we set it to some tiny fraction, significantly reducing impact on utility. Given a fixed integer $k$:

1. Set $r$ smallest power of 2 exceeding $(\Delta/r)2^{-k}$.
2. Sampler$(\epsilon, r, \Delta_r)$: Sample an integer $i$ with probability proportional to $\exp(-|i|r\epsilon/\Delta_r)$

**Proposition 1.** *The mechanism M(D) is $\epsilon$-differentially private.*

*Proof.* The output value is guaranteed to be a multiple of $r$. Consider an arbitrary output value $y = jr$ and two adjacent databases $D, D'$. We have:

$$\Pr[M(D) = y] = \Pr[z = y - x(D)] = \Pr[i = j - x(D)/r] \tag{1}$$

Thus,

$$\frac{\Pr[M(D) = y]}{\Pr[M(D') = y} = \exp\left(-\left(|j - x(D)/r| - |j - x(D')/r|\right)\frac{r\epsilon}{\Delta + r}\right) \tag{2}$$

$$= \exp\left(-\left(|y - x(D)| - |y - x(D')|\right)\frac{\epsilon}{\Delta + r}\right) \tag{3}$$

$$\leq \exp\left(|x(D) - x(D')|\frac{\epsilon}{\Delta + r}\right) \ . \tag{4}$$

Noting that

$$|x(D) - x(D')| \leq |x(D) - f(D)| + |f(D) - f(D')| + |f(D') - x(D')| \tag{5}$$

$$\leq r/2 + \Delta + r/2 \tag{6}$$

$$= \Delta + r \ , \tag{7}$$

the probability ratio is bounded by $\exp(\epsilon)$ and we have the result. □

Note that the rounding is forcing us to add some extra noise on the order of $r/\epsilon$, or about $\Delta/(2^k\epsilon^2)$. In practice, as long as $\epsilon$ is larger than about $2^{-k}$, this is unlikely to matter. This is quantified in the next proposition.

**Proposition 2.** *The expected error of $M(D)$ is close to the standard Laplace mechanism; specifically:*

$$\frac{\mathbb{E}|M(D) - f(D)|}{\mathbb{E}|Lap(\Delta/\epsilon)|} \leq 1 + \frac{1 + 2/\epsilon}{2^k} \ . \tag{8}$$

*Proof.* Using Lemma 3 from below:

$$\mathbb{E}|M(D) - f(D)| \leq r/2 + \mathbb{E}|z| \tag{9}$$

$$= r\left(1/2 + \mathbb{E}|i|\right) \tag{10}$$

$$\leq r\left(1/2 + \frac{\Delta + r}{\epsilon r}\right) \tag{11}$$

$$= r/2 + (\Delta + r)/\epsilon \ . \tag{12}$$

Since $r \leq \Delta/(2^{k-1}\epsilon)$, we have

$$\mathbb{E}|M(D) - f(D)| \leq \frac{\Delta}{\epsilon}\left(1 + \frac{1 + 2/\epsilon}{2^k}\right) \ . \tag{13}$$

Since the expected absolute value of $Lap(\Delta/\epsilon)$ is $\Delta/\epsilon$, we have the desired result. □

Note that Proposition 2 is loose, and in many realistic cases our mechanism is actually better than the Laplace mechanism. This is because the geometric distribution is inherently preferable to Laplace for differential privacy on discrete values (see this paper).

Note also that the error for Mironov's mechanism is roughly $2\Delta/\epsilon$ (see the paragraph just after the proof of Theorem 1 in the paper), so for reasonable parameter values $M(D)$ should have about 50% less error.

**Lemma 3.** *If $i$ is an integer sampled with probability proportional to $\exp(-c|i|)$ for some constant $c > 0$, then*

$$\mathbb{E}|i| = \frac{2}{\exp(c) - \exp(-c)} \leq \frac{1}{c} \ . \tag{14}$$

*Proof.* We start with some geometric series calculations:

$$A := \sum_{i=-\infty}^{\infty} \exp(-c|i|) \tag{15}$$

$$= 1 + 2\sum_{i=1}^{\infty} \exp(-ci) \tag{16}$$

$$= 1 + \frac{2\exp(-c)}{1 - \exp(-c)} \tag{17}$$

$$= \frac{1 + \exp(-c)}{1 - \exp(-c)} \tag{18}$$

$$B := \sum_{i=-\infty}^{\infty} |i| \exp(-c|i|) \tag{19}$$

$$= 2\sum_{i=1}^{\infty} i \exp(-ci) \tag{20}$$

$$= 2\sum_{j=1}^{\infty}\sum_{i=j}^{\infty} \exp(-ci) \tag{21}$$

$$= 2\sum_{j=1}^{\infty} \frac{\exp(-cj)}{1 - \exp(-c)} \tag{22}$$

$$= \frac{2\exp(-c)}{(1 - \exp(-c))^2} \tag{23}$$

So we have

$$\mathbb{E}|i| = B/A \tag{24}$$

$$= \frac{2\exp(-c)}{(1 - \exp(-c)) * (1 + \exp(-c))} \tag{25}$$

$$= \frac{2\exp(-c)}{1 - \exp(-2c)} \tag{26}$$

$$= \frac{2}{\exp(c) - \exp(-c)} \ . \tag{27}$$

Finally, note that when $c = 0$, $2c = 0 = \exp(c) - \exp(-c)$, and for $c \geq 0$ we have

$$\frac{\partial}{\partial c}[\exp(c) - \exp(-c)] = \exp(c) + \exp(-c) \geq 2 \ . \tag{28}$$

Thus $2c \leq \exp(c) - \exp(-c)$ for all $c >= 0$. □

### 4.1 Discussion

In summary, all of the steps required to implement $M(D)$ can be performed exactly on a modern computer, allowing that (a) we assume away (or $\delta$ away) astronomically unlikely events and (b) we accept a mandatory post-processing step that rounds the final result to the nearest double. Neither of these is of serious concern.

Note that we now have a tradeoff for the choice of $k$, which controls the accuracy of the discretization. If $k$ is small, we discretize coarsely and potentially suffer utility loss due to rounding. If $k$ is too large, we discretize finely and might have an unacceptably high probability of the noise level stretching beyond the capacity of a double, leaving us with a privacy violation. Luckily, in practice, we have a very wide range of acceptable values; probably anything between 10 and 45 is perfectly fine.

For example, if we select $k$ as a function of $\epsilon$, say, $k = 10 + \log_2(1 + 2/\epsilon)$, the utility guarantee becomes

$$\frac{\mathbb{E}|M(D) - f(D)|}{\mathbb{E}|Lap(\Delta/\epsilon)|} \leq 1 + (1 + 2/\epsilon)/2k \tag{29}$$

$$\leq 1 + 2^{-10} \tag{30}$$

$$\leq 1.001\ . \tag{31}$$

At the same time if $\epsilon \geq 2^{-9}$, then the privacy violation probability in step 3 is bounded by

$$\exp(-2^{50-k}) \leq \exp(-2^{40-\log_2(1+2^{10})}) \tag{32}$$

$$\leq \exp(-2^{29}) \tag{33}$$

$$\leq 1/[\text{atoms in the universe}]^3\ . \tag{34}$$

## 5 Efficient Binomial/Gaussian Sampling

The main idea is to realize hole-free Gaussian sampling using binomial sampling. There are three ingredients to make this work. The first is to argue this does not hurt accuracy. The second is to show this can be done in a hole-free way. The third is regarding the computational efficiency.

Assume for simplicity that $n$ is even. Let $X = U - \frac{n}{2}$ where $U \sim Bin(n, 1/2)$ and let $V \sim N(0, \frac{n}{4})$. Note that the random variables $X$ and $V$ have the same mean and standard deviation. Intuitively, we wish to show that the random variables $X$ and $V$ are "close". Since the total variation distance between a discrete and a continuous random variable is infinite, we start by considering a natural discretization of the distribution of $V$. Specifically, let $p_V(\cdot)$ denote the probability density function of $V$. Note that for all $v \in \mathbb{R}$, it holds that $p_V(v) = \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2v^2}{n}}$. Let $Y$ be the discrete random variable obtained by sampling a random variable from $N(0, \frac{n}{4})$ and taking the ceiling in $\mathbb{Z}$. We next show that the (discrete) random variables $X$ and $Y$ are at a small total variation distance. Note that we need an explicit (non-asymptotic) bound on the total-variation distance in order to obtain a concrete guarantee for the particular parameter settings of our algorithm.

**Lemma 4** (Distance between Binomial and Discretized Gaussian; [CGS10])**.** *For any positive integer $n$, the total variation distance between $X$ and $Y$ is at most $\frac{15.2}{\sqrt{n}}$.*

From Lemma 4, in order to sample a Gaussian random variable in a hole-free way it suffices to:

1. Sample a binomial random variable $U \sim Bin(n, 1/2)$ and set $X = U - \frac{n}{2}$.

2. Normalize $X$ by the standard deviation obtaining $\hat{Y} = \frac{2 \cdot X}{\sqrt{n}}$.

In order for the output $\hat{Y}$ to be as close as possible to a true Gaussian, Lemma 4 requires setting $\sqrt{n}$ as large as possible (ideally so that the upper bound of $\frac{15.2}{\sqrt{n}}$ is close to machine precision). It turns out that the main challenge is sampling of the random variable $U$ in step 1 in a way that is at the same time:

(i) efficient, i.e., the sampling should run in $O(1)$ of steps,

(ii) implementable using operations on 64-bit integers, even for values of $\sqrt{n}$ larger than $2^{32}$.

(iii) hole-free.

Note that without the efficiency constraint (i), sampling $U$ could be done by flipping $n$ independent unbiased coins and aggregating the values in a 64-bit integer, though of course this approach would run in time $O(n)$ and would be absolutely infeasible for large values of $n$ (such as the ones we have to consider in order to guarantee small total variation error). On the other hand, to achieve the hole-freeness property (iii), it is sufficient to prove that the mechanism for sampling $U$ produces its desired distribution up to a very small total variation error, which can then be lumped in the "$\delta$ probability" of the differential privacy failure event. As acceptable values of $\delta$ can range between $10^{-6}$ and $10^{-9}$, the total variation error due to this sampling procedure has to be bounded by a very small quantity.

A rejection sampling based algorithm of Bringmann et al. [BKP+14] achieves the efficiency requirement (i). However, it cannot be directly implemented using operations on 64-bit integers while allowing values of $\sqrt{n}$ larger than $2^{32}$, as required in (ii) above. This is because its sampling probabilities are computed by approximating the binomial coefficients $\binom{n}{k}$ using Spouge's approximation of the factorial, which requires performing arithmetic operations on the value of $n$ itself (as opposed to $\sqrt{n}$)) which does not seem directly doable using with 64-bit integers. Instead, we will use a simple accurate approximation of the binomial probability mass function (Lemma 5 below) which can be computed by performing 64-bit arithmetic operations on $\sqrt{n}$ itself without having to explicitly store the value of $n$ (which will be larger than $2^{64}$). Note that asymptotic versions of the Lemma 5 are well-known in the literature. Nevertheless we work out the proof as we need to state explicit constants for concrete settings of $n$.

**Lemma 5** (Binomial Density Approximation)**.** *Assume that $n \geq 10^6$ and that $n$ is even*[1]*. For all $i \in \{-\frac{\sqrt{n \cdot \ln n}}{2}, \ldots, +\frac{\sqrt{n \cdot \ln n}}{2}\}$ the probability that $X$ is equal to $i$ is given by:*

$$p_X(i) = \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 + \zeta_n),$$

*where $|\zeta_n| \leq \frac{0.4 \cdot \ln^{1.5}(n)}{\sqrt{n}}$. In particular for $n = 2^{96}$, it holds that $|\zeta_n| \leq 2^{-40}$.*

We next summarize the rejection sampling procedure of [BKP+14] which we will use.

**Lemma 6** (Rejection Binomial Sampling; [BKP+14])**.** *Let $n$ be an even integer, $\kappa < 1/8$, and $\tilde{p} : \{-\frac{n}{2}, \ldots, +\frac{n}{2}\} \to [0, 1]$ be such that:*

*1. the $\ell_1$-distance between $\tilde{p}$ and $p_X$ is bounded, i.e.,*

$$\|\tilde{p} - p_X\|_1 := \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} |\tilde{p}(i) - p_X(i)| \leq \kappa.$$

[1]This assumption is made for ease of notation.

2. *$\tilde{p}$ is bounded point-wise by $p_X$, i.e., for all $i \in \{-\frac{n}{2}, \ldots, +\frac{n}{2}\}$,*

$$\tilde{p}(i) \leq p_X(i),$$

3. *the value of $\tilde{p}$ at every point in its support can be approximated up to machine precision using operations on 64-bit integers.*

*Then, there is an algorithm that outputs a random variable $\hat{X}$ such that:*

*(a) the total variation distance between $X$ and $\hat{X}$ is small, namely,*

$$d_{TV}(X, \hat{X}) \leq \kappa,$$

*(b) the expected number of steps executed by the algorithm is constant,*

*(c) the algorithm can be implemented using simple arithmetic operations on 64-bit integers.*

We now use Lemma 5 to construct a function $\tilde{p}$ satisfying properties 1, 2 and 3 that are listed in Lemma 6.

**Lemma 7.** *Let $n$ be an even integer. For $i \in \{-\frac{n}{2}, \ldots, +\frac{n}{2}\}$, we define the function*

$$\tilde{p}(i) = \begin{cases} \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 - \nu_n) & \text{if } -\frac{\sqrt{n \cdot \ln n}}{2} \leq i \leq -\frac{\sqrt{n \cdot \ln n}}{2}, \\ 0 & \text{otherwise,} \end{cases}. \tag{35}$$

*where $\nu_n := \frac{0.4 \cdot \ln^{1.5}(n)}{\sqrt{n}}$. Then, $\tilde{p}$ satisfies properties 1, 2 and 3 in Lemma 6 with $\kappa = \frac{0.88 \cdot \ln^{1.5}(n)}{\sqrt{n}}$.*

The pseudocode for the overall binomial sampling algorithm is given in Algorithm 1.

**Algorithm 1** Efficient Binomial Sampling

```
1:  procedure BinomialSample(√n)
2:      if √n < 4 then
3:          i ← 0
4:          for n iterations do
5:              i ← i + Ber(1/2)
6:          return i − n/2
7:      m ← ⌊√2 · √n + 1⌋.
8:      while true do
9:          s ∼ Geom(1/2)
10:         k ← s with probability 1/2 and k ← −s − 1 with probability 1/2.
11:         i ← km + ℓ where ℓ is a uniform random number between 0 and m − 1.
12:         if p̃(i) > 0 then
13:             f ← 4/(m·2^s).
14:             c ← Ber(p̃(i)/f)
15:             if c = 1 then
16:                 return i
```

**Remark 8.** *We note that there are in principle two additional possible sources of total variation error in implementations of Algorithm 1, namely:*

1. *The fact that the sampling of the random variable $i$ in line 11 might incur a total variation error of $\delta'$ in actual implementations (this includes the total variation error in sampling from the geometric distribution in line 9, the sampling of $k$ in line 10 and that of the uniform random number in line 11, in addition to the exponentially small probability that the results do not fit in a 64-bit integer which would result in overflows).*

2. *The fact that the sampling of the biased Bernouilli random variable in line 14 might suffer from a total variation error of $\delta''$.*

*These two errors would not significantly alter our overall total variation bound. As in [BKP+14], we define $f(i) := \frac{4}{2^{max(s,-s-1)m}}$ and $\overline{f}(i) = \frac{f(i)}{16}$, for all $i \in \{sm, sm+1, \ldots, sm+m-1\}$ and all $s \in \mathbb{Z}$. Note that $\overline{f}$ is a probability distribution. Thus, the total variation distance between a true outcome of the rejection sampling process in (a given iteration of the while loop in) Algorithm 1 and the outcome in the noisy process (that incorporates the aforementioned $\delta'$ and $\delta''$ events) can be bounded as*

$$\frac{1}{2}\left|\sum_i (\overline{f}(i)+\xi_i)(\frac{\tilde{p}(i)}{f(i)}+\delta'') - \sum_i \overline{f}(i)\frac{\tilde{p}(i)}{f(i)}\right| \leq \delta' + \frac{\delta''}{2} + \delta'\delta'',$$

*where $\sum_i |\xi_i| = 2\delta'$.*

### 5.1 Putting things together for $N(0,1)$

We apply Lemmas 6 and 7 with $n = 2^{96}$. For this setting, the random variable $\hat{Y} = \frac{2 \cdot X}{\sqrt{n}}$ is an integer multiple of $2^{-50}$ and it lies between $-5$ and $+5$. Thus, $\hat{Y}$ is exactly representible as a 64-bit floating point number (and its generation fits within the recipe of Section 3). Applying Lemmas 4 and 7 with the triangle inequality implies that the total variation distance between $\hat{Y}$ and the normalized discretized Gaussian $\frac{\sqrt{n}}{2} \cdot Y$ is at most

$$\frac{15.2}{\sqrt{n}} + \frac{0.88 \cdot \ln^{1.5}(n)}{\sqrt{n}},$$

which for $n = 2^{96}$ is at most $2^{-40}$.

Denote by $s$ the input to which the (approximate) $N(0,1)$ noise is to be added. As long as we first round $s$ to the nearest integer multiple of $2^{-50}$, the above implies if the mechanism analyzed under true $N(0,1)$ noise was $(\epsilon, \delta)$-differentially private, then the output of the above algorithm is $(\epsilon, \delta + 2^{-40})$-differentially private.

### 5.2 Sampling of $N(\mu, \sigma^2)$ with general $\mu$ and $\sigma$

The above handles the case where the desired distribution is $N(0,1)$. We next describe how to handle the case of $N(\mu, \sigma^2)$. Let $s$ be the input value to which an (approximate) $N(\mu, \sigma^2)$ noise term has to be added. We can first divide $s$ by $\sigma$, then add (approximate) noise sampled from $N(0,1)$ as in the above. The previous subsection guarantees that the result in differentially private. We can then multiply the noise outcome by $\sigma$ and then add $\mu$. The post-processing property of differential privacy guarantees that the scaled and shifted value is still differentially private. The degree to which the distribution of the output approximates a $N(\mu, \sigma^2)$ random variable remains the same as in the previous subsection as long as $\mu$ and $\sigma$ are accurately representible as 64-bit floating point numbers. Moreover, for the above scheme to be accurate, $s$ and $\sigma$ should be in a

reasonable range so that dividing $s$ by $\sigma$ and then multiplying the result by $\sigma$ should produce a value close to $s$ (e.g., the intermediate result $\frac{s}{\sigma}$ does not go out-of-range).

Note that in the actual implementation we set the scaling parameters slightly differently in order to parallel the treatment for the Laplace mechanism described in Section 3. Specifically, we divide $s$ by a "granularity" parameter that is set to the smallest power of 2 greater than or equal to $\frac{\sigma}{2^{56}}$. We then round the result to the closest integer, add to it the binomial sample with $\sqrt{n}$ set to the ratio of $\sigma$ to the granularity. It can be verified that $\sqrt{n}$ is guaranteed to be between $2^{56}$ and $2^{57}$ (and hence a total variation bound even stronger than the one stated in Section 5.1 holds). Finally, we multiply the sum by the granularity parameter. By tracking the cancellations, this operation can be seen to be equivalent to the one described in the previous paragraph.

### 5.3 Proof of Lemma 5

In order to prove Lemma 5, we will need the following explicit version of Stirling's approximation of the factorial.

**Lemma 9** ([Rob55])**.** *For every positive integer $n$, it holds that*

$$\sqrt{2\pi} \cdot n^{n+\frac{1}{2}} \cdot e^{-n} \cdot e^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi} \cdot n^{n+\frac{1}{2}} \cdot e^{-n} \cdot e^{\frac{1}{12n}}.$$

We are now ready to prove Lemma 5.

*Proof of Lemma 5.* Throughout the proof, we denote $t := \frac{\sqrt{n \cdot \ln n}}{2}$. We have that:

$$\begin{aligned} p_X(i) &:= \Pr[U = i + \frac{n}{2}] \\ &= \binom{n}{i + \frac{n}{2}} \cdot \frac{1}{2^n} \\ &= \frac{n!}{(\frac{n}{2}+i)! \cdot (\frac{n}{2}-i)!} \cdot \frac{1}{2^n} \end{aligned} \tag{36}$$

Applying Stirling's approximation in Lemma 9, we get that

$$n! = \sqrt{2\pi} \cdot n^{n+\frac{1}{2}} \cdot e^{-n} \cdot \nu(n),$$

where

$$e^{\frac{1}{12n+1}} \leq \nu(n) \leq e^{\frac{1}{12n}}, \tag{37}$$

and similarly, for $(\frac{n}{2}+i)!$ and $(\frac{n}{2}-i)!$. Plugging back in Equation (36), we get that

$$\begin{aligned} p_X(i) &= \sqrt{\frac{n}{2\pi \cdot (\frac{n}{2}+i) \cdot (\frac{n}{2}-i)}} \cdot \frac{n^n}{(\frac{n}{2}+i)^{\frac{n}{2}+i} \cdot (\frac{n}{2}-i)^{\frac{n}{2}-i}} \cdot \frac{\nu(n)}{\nu(\frac{n}{2}+i) \cdot \nu(\frac{n}{2}-i)} \\ &= \sqrt{\frac{1}{2\pi \cdot (\frac{n}{4} - \frac{i^2}{n})}} \cdot \frac{1}{(1+\frac{2i}{n})^{\frac{n}{2}+i} \cdot (1-\frac{2i}{n})^{\frac{n}{2}-i}} \cdot \kappa(n,i), \end{aligned} \tag{38}$$

where $\kappa(n,i) = \frac{\nu(n)}{\nu(\frac{n}{2}+i) \cdot \nu(\frac{n}{2}-i)}$ satisfies

$$e^{-\frac{1}{6 \cdot (\frac{n}{2}-t)}} \leq \kappa(n,i) \leq e^{\frac{1}{12n}}, \tag{39}$$

for all $i \in \{-t, \ldots, t\}$ (this follows from the inequality in (37)). We next bound the middle factor in (38) which we denote as

$$\gamma(n,i) := \frac{1}{(1+\frac{2i}{n})^{\frac{n}{2}+i} \cdot (1-\frac{2i}{n})^{\frac{n}{2}-i}} = \frac{1}{\alpha(n,i) \cdot \beta(n,i)} \tag{40}$$

where

$$\alpha(n,i) := (1+\frac{2i}{n})^{\frac{n}{2}+i} = e^{(\frac{n}{2}+i)\cdot \ln(1+\frac{2i}{n})} \tag{41}$$

and

$$\beta(n,i) := (1-\frac{2i}{n})^{\frac{n}{2}-i} = e^{(\frac{n}{2}-i)\cdot \ln(1-\frac{2i}{n})}. \tag{42}$$

Consider the function $f(x) := \ln(1+x)$ where $|x| < 1$. The second-order Taylor approximation of $f$ is given by $f(x) = x - \frac{x^2}{2} + R(x)$ where the remainder term satisfies $|R(x)| \leq \frac{|x|^3}{3}$. This approximation implies that

$$\ln(1+\frac{2i}{n}) = \frac{2i}{n} - \frac{2i^2}{n^2} + \theta(n,i), \tag{43}$$

and

$$\ln(1-\frac{2i}{n}) = -\frac{2i}{n} - \frac{2i^2}{n^2} + \eta(n,i), \tag{44}$$

where

$$|\theta(n,i)|, |\eta(n,i)| \leq \frac{8|i|^3}{3n^3} \tag{45}$$

Plugging (43) and (44) back in (41) and (42) respectively, we get that

$$\alpha(n,i) = e^{i+\frac{i^2}{n}-\frac{2i^3}{n^2}+(\frac{n}{2}+i)\cdot\theta(n,i)}, \tag{46}$$

and

$$\beta(n,i) = e^{-i+\frac{i^2}{n}+\frac{2i^3}{n^2}+(\frac{n}{2}-i)\cdot\eta(n,i)}. \tag{47}$$

Plugging (46) and (47) back into (40) yields

$$\gamma(n,i) = e^{-\frac{2i^2}{n}-(\frac{n}{2}+i)\cdot\theta(n,i)-(\frac{n}{2}-i)\cdot\eta(n,i)}. \tag{48}$$

Using (38) and (48), we obtain that

$$\begin{aligned}
p_X(i) &= \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot \frac{1}{\sqrt{1-\frac{4i^2}{n^2}}} \cdot e^{-(\frac{n}{2}+i)\cdot\theta(n,i)-(\frac{n}{2}-i)\cdot\eta(n,i)} \cdot \kappa(n,i) \\
&\leq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1+\frac{8t^2}{n^2}) \cdot e^{\frac{8t^3}{3n^2}} \cdot e^{\frac{1}{12n}} && (49) \\
&\leq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1+\frac{16t^2}{n^2}) \cdot (1+\frac{16t^3}{3n^2}) \cdot (1+\frac{1}{6n}) && (50) \\
&= \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1+\frac{16t^2}{n^2}+\frac{16t^3}{3n^2}+\frac{1}{6n}+\frac{256t^5}{3n^4}+\frac{8t^2}{3n^3}+\frac{8t^3}{9n^3}+\frac{128t^5}{9n^5}), && (51)
\end{aligned}$$

where inequality (49) follows from the fact that $\frac{1}{\sqrt{1-a}} \leq 1+2a$ for all $a \in [0,1)$, the assumption that $|i| \leq t$, as well as (39) and (45), and inequality (50) follows from the fact that $e^x \leq 1+2x$ for all $x \in [0,1]$.

On the other hand, we have that:

$$p_X(i) \geq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot e^{-\frac{8t^3}{3n^2}} \cdot e^{-\frac{1}{6\cdot(\frac{n}{2}-t)}} \tag{52}$$

$$\geq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 - \frac{8t^3}{3n^2}) \cdot (1 - \frac{1}{3n-6t}) \tag{53}$$

$$\geq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 - \frac{8t^3}{3n^2}) \cdot (1 - \frac{1}{2n}) \tag{54}$$

$$= \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 - \frac{8t^3}{3n^2} - \frac{1}{2n} + \frac{4t^3}{3n^3})$$

$$\geq \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 - \frac{8t^3}{3n^2} - \frac{1}{2n}), \tag{55}$$

where inequality (52) follows from (39) and (45), inequality (53) applies twice the fact that $e^{-a} \geq 1-a$ for any real number $a$, and inequality (54) follows from the fact that $t \leq n/6$. Putting together inequalities (51) and (55), we get that

$$p_X(i) = \sqrt{\frac{2}{\pi n}} \cdot e^{-\frac{2i^2}{n}} \cdot (1 + \zeta_n),$$

with

$$\begin{aligned} |\zeta_n| &\leq \frac{16t^2}{n^2} + \frac{8t^3}{3n^2} + \frac{1}{2n} + \frac{128t^5}{3n^4} + \frac{8t^2}{3n^3} + \frac{4t^3}{9n^3} + \frac{64t^5}{9n^5} \\ &\leq \frac{1.2 \cdot \ln^{1.5}(n)}{3\sqrt{n}} \\ &= \frac{0.4 \cdot \ln^{1.5}(n)}{\sqrt{n}}. \end{aligned} \tag{56}$$

where inequality (56) follows from the fact that $t = \frac{\sqrt{n \cdot \ln n}}{2}$ and the assumption that $n \geq 10^6$. □

### 5.4 Proof of Lemma 7

To prove Lemma 7, we will need the following standard probabilistic inequality.

**Lemma 10** (Hoeffding's inequality [Hoe94])**.** *For any positive integer $n$ and real number $p \in [0,1]$, if $Z \sim Bin(n,p)$ then for any positive real number $\epsilon$,*

$$\Pr[Z \leq (p-\epsilon) \cdot n] \leq e^{-2 \cdot \epsilon^2 \cdot n}.$$

We are now ready to prove Lemma 7.

*Proof of Lemma 7.* Denote $t := \frac{\sqrt{n \cdot \ln n}}{2}$. We have that

$$\begin{aligned} \|\tilde{p} - p_X\|_1 &:= \sum_{i \in \mathbb{Z}} |\tilde{p}(i) - p_X(i)| \\ &= \sum_{i=-t}^{t} |\tilde{p}(i) - p_X(i)| + \sum_{i \in \mathbb{Z} \setminus \{-t,\ldots,t\}} |\tilde{p}(i) - p_X(i)| \end{aligned} \tag{57}$$

We next separately upper-bound the two sums in (57). For the first, we have

$$\begin{aligned}\sum_{i=-t}^{t}|\tilde{p}(i)-p_X(i)| &= \sum_{i=-t}^{t} p_X(i)\cdot|\frac{\tilde{p}(i)}{p_X(i)}-1| \\ &\leq \sum_{i=-t}^{t} p_X(i)\cdot\nu_n \qquad (58)\\ &\leq \nu_n, \qquad (59)\end{aligned}$$

where the inequality in (58) follows from Lemma 5 and the setting $\nu_n = \frac{0.4\cdot\ln^{1.5}(n)}{\sqrt{n}}$. For the second sum in (57), we have

$$\begin{aligned}\sum_{i\in\mathbb{Z}\setminus\{-t,\ldots,t\}}|\tilde{p}(i)-p_X(i)| &\leq \sum_{i\in\mathbb{Z}\setminus\{-t,\ldots,t\}} p_X(i) + \sum_{i\in\mathbb{Z}\setminus\{-t,\ldots,t\}}\tilde{p}(i)\\ &\leq \Pr[X\notin\{-t,\ldots,t\}] \qquad (60)\\ &\leq \frac{2}{\sqrt{n}}, \qquad (61)\end{aligned}$$

where the inequality in (60) follows the fact that $\tilde{p}(i)=0$ for all $i\in\mathbb{Z}\setminus\{-t,\ldots,t\}$, and inequality (61) follows from Hoeffding's inequality (Lemma 10). Plugging (59) and (61) back into (57) and using the assumption that $n\geq 10^6$, we deduce that property 1 of Lemma 6 holds with $\kappa = \frac{11}{5}\cdot\nu_n = \frac{0.88\cdot\ln^{1.5}(n)}{\sqrt{n}}$.

Property 2 of Lemma 6 directly follows from the setting of $\tilde{p}$ in (35) and from Lemma 5. Finally, to prove that $\tilde{p}$ satisfies property 3 of Lemma 6, we separately consider the 3 factors in (35):

- The factor of $\sqrt{\frac{2}{\pi n}}$ can be represented as a 64-bit floating point number up to machine precision by computing $\sqrt{\frac{2}{\pi}}$ and multiplying it by $\frac{1}{\sqrt{n}}$.
- The last factor of $(1-\nu_n)$ is also computed naturally as a 64-bit floating point number, e.g., compute $\nu_n := \frac{0.4\cdot\ln^{1.5}(n)}{\sqrt{n}}$ up to machine precision and subtract it from 1.
- The middle factor of $e^{-\frac{2i^2}{n}}$ is slightly trickier to compute. This is because both $i^2$ and $n$ can be larger than $2^{64}$ while the ratio $\frac{2i^2}{n}$ can still be small (and could still be represented as a floating point number up to machine precision). To circumvent this, we can proceed by computing a 64-bit floating point representation of the factor $\frac{i}{\sqrt{n}}$, then squaring it and multiplying the result by 2. This guarantees that the result of the computation is accurate up to machine precision. Since exact rounding algorithms for the exponential function are available, we conclude that we can approximate the middle factor of $e^{-\frac{2i^2}{n}}$ up to machine precision.

Putting the three approximations together, we conclude that property 3 of Lemma 6 holds. □

# References

[BKP+14] Karl Bringmann, Fabian Kuhn, Konstantinos Panagiotou, Ueli Peter, and Henning Thomas. Internal dla: Efficient simulation of a physical growth model. In *International Colloquium on Automata, Languages, and Programming*, pages 247–258. Springer, 2014.

[CGS10] Louis HY Chen, Larry Goldstein, and Qi-Man Shao. *Normal approximation by Stein's method*. Springer Science & Business Media, 2010.

[CKS20] Clément Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[DR14] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers Inc., 2014.

[Hoe94] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[Mir12] Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 650–661, 2012.

[Rob55] Herbert Robbins. A remark on stirling's formula. *The American mathematical monthly*, 62(1):26–29, 1955.