# Privacy Enhancing Technologies
# Lecture 10 – Differentially Private Algorithms

Florian Tramèr

AGENDA

1. Recap
2. The Laplace Mechanism
3. Approximate DP
4. The Gaussian Mechanism

## 1 Recap on Differential Privacy

Last time we saw the definition of differential privacy and some of its nice properties as a privacy measure. We also saw one simple algorithm that achieves differential privacy for collecting bits: Randomized Response. This algorithm allowed us to estimate the true fraction of yes-answers among a group of surveyed individuals with an error that grows as $O(1/\varepsilon\sqrt{n})$.

We will now see other popular algorithms (or "mechanisms") that achieve differential privacy, which can achieve better error rates for collecting one (or more) summary statistics. We will improve upon the randomized response algorithm in two ways:

1. Recall that Randomized Response achieves a strong notion of *local* differential privacy, where each individual's data release satisfies DP. Here, we will look at algorithms that provide privacy for the summary statistics, but where the data is first collected in the clear by a trusted curator.
2. We will see a relaxation of DP, called approximate DP, which allows for much better error rates in some settings, at the cost of a very small probability of privacy violation.

## 2 The Laplace Mechanism

Recall that to achieve DP, we need the output distributions of our algorithm on two neighboring databases to be close to each other. The general idea of many DP algorithms is to first compute the true answer of the function in question, and then add noise to *hide* which database was used. For this, we need a sense of how much the output of the function can change when a single input changes, which we call the *sensitivity* of a function.

**Definition 1** ($\ell_1$-Sensitivity)**.** The $\ell_1$-sensitivity of a function $f\colon \mathcal{X}^n \to \mathbb{R}^k$ is defined as

$$\Delta_1 = \max_{D \sim D'} \|f(D) - f(D')\|_1 = \max_{D,D'} \sum_{i=1}^{k} |f(D)_i - f(D')_i| \,,$$

where $D$ and $D'$ are neighboring databases.

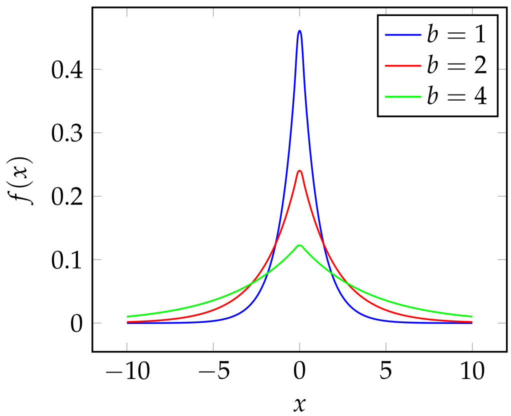
When $f$ outputs a scalar, this is simply the maximum difference in the outputs of $f$ on two neighboring databases. For example, if $f$ is the sum of $n$ bits, the sensitivity is 1.

Now, to achieve DP, intuitively we want to add noise to the output of $f$ taken from a probability distribution whose density changes by at most $e^{\varepsilon}$ when the input changes by $\Delta_1$. There is a distribution that exactly satisfies this property: the Laplace distribution.

**Definition 2** (Laplace Distribution)**.** The Laplace distribution $\texttt{Laplace}(\mu, b)$ with location $\mu$ and scale $b > 0$ has density

$$f(x \mid \mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}, \quad x \in \mathbb{R}\,.$$

This distribution has mean $\mu$ and variance $2b^2$. The distribution is plotted below for a few values of $b$ and $\mu = 0$.




As a point of comparison, the Gaussian distribution (which we'll see later) has density $\propto e^{-x^2}$ instead of $e^{-|x|}$, and thus decays much faster than the Laplace distribution.

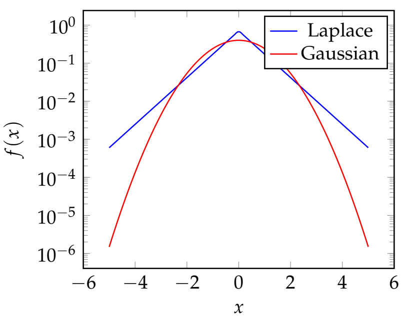



Figure 1: Laplace and Gaussian of variance 1. The Gaussian decays much faster.

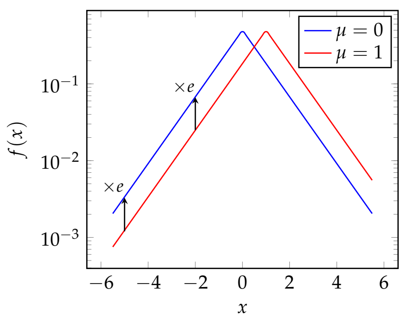



Figure 2: Laplace distributions shifted by 1 stay within a factor of $e$ ($b = 1$).

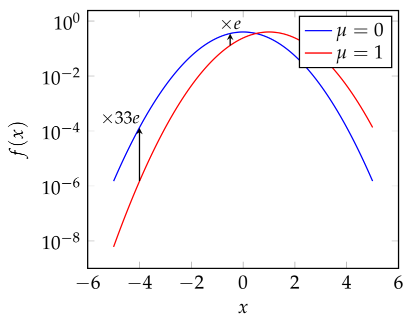



Figure 3: Gaussians shifted by 1 deviate by an increasing factor ($\sigma = 1$).

We are now ready to introduce the Laplace mechanism, which adds noise from a Laplace distribution scaled to the sensitivity of the function:

**Definition 3** (Laplace Mechanism [DMNS06])**.** Let $f\colon \mathcal{X}^n \to \mathbb{R}^k$ be a function with $\ell_1$-sensitivity $\Delta_1$, and let $\varepsilon > 0$. The Laplace mechanism is defined as

$$M(D) = f(D) + (Y_1, \ldots, Y_k)$$

where the $Y_i$ are i.i.d. samples from the $\texttt{Laplace}(0, \Delta_1/\varepsilon)$ distribution.

We can now prove that the Laplace mechanism is differentially private.

**Theorem 1** ([DMNS06])**.** The Laplace mechanism is $\varepsilon$-DP.

*Proof.* Let $D$ and $D'$ be neighboring databases. Let $p_D(y)$ and $p_{D'}(y)$ be the probability density functions of the mechanism outputs $M(D)$ and $M(D')$ respectively, evaluated at an arbitrary point $y \in \mathbb{R}^k$. Let $b = \Delta_1/\varepsilon$. Then,

$$\begin{aligned}
\frac{p_D(y)}{p_{D'}(y)} &= \frac{\prod_{i=1}^{k} \frac{1}{2b} \exp\left(-\frac{|y_i - f(D)_i|}{b}\right)}{\prod_{i=1}^{k} \frac{1}{2b} \exp\left(-\frac{|y_i - f(D')_i|}{b}\right)} \\
&= \prod_{i=1}^{k} \exp\left(\frac{|y_i - f(D')_i| - |y_i - f(D)_i|}{b}\right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{|f(D')_i - f(D)_i|}{b}\right) && \text{by reverse triangle inequality} \\
&= \exp\left(\frac{\sum_{i=1}^{k} |f(D')_i - f(D)_i|}{b}\right) \\
&= \exp\left(\frac{\|f(D') - f(D)\|_1}{b}\right) \\
&\leq \exp\left(\frac{\Delta_1}{b}\right) = \exp(\varepsilon)\,. && \text{by def. of sensitivity}
\end{aligned}$$

□

## 2.1 Answering Counting Queries With the Laplace Mechanism

So we now have two differentially private algorithms: Randomized Response and the Laplace mechanism. Recall that for computing a mean of $n$ private bits, the randomized response algorithm we analyzed achieves error $O(1/\varepsilon\sqrt{n})$.

To achieve a similar privacy level with the Laplace mechanism, we need to add Laplace noise with scale parameter $b = \Delta_1/\varepsilon = O(1/\varepsilon n)$ (the sensitivity of the mean is $1/n$). The variance of the noise is $2b^2 = O(1/\varepsilon^2 n^2)$, and so the error in our estimate is $O(1/\varepsilon n)$ with high probability (see Appendix A). This is *quadratically smaller* than the error of the randomized response algorithm! This is the power of the central model of DP over the local model.

**Multiple queries.** For now, we asked a single query of our database. What if we wanted to answer many queries? Our multi-dimensional definition of the Laplace mechanism makes this easy to reason about. Suppose we had $k$ counting functions $f = (f_1, \dots, f_k)$ executed on the same dataset.[1] We would then output the vector $f(D) + Y$, where the $Y_i$'s are i.i.d. Laplace random variables.

What's the sensitivity of the function $f$? Each individual counting query $f_i$ has sensitivity $1/n$, but they are all computed on the same dataset. So swapping a single individual might affect the result of all queries. More formally, for any two neighboring databases $D$ and $D'$ we have $\|f(D) - f(D')\|_1 = \sum_{i=1}^{k} |f_i(D) - f_i(D')| \le k/n$. So we can add noise $Y_i \sim \texttt{Laplace}(0, {}^{k}/_{\varepsilon n})$ to each coordinate. A standard concentration bound for the Laplace distribution shows that, with high probability, the maximum noise across all coordinates is $O(\frac{k \log k}{\varepsilon n}) = \tilde{O}({}^{k}/_{\varepsilon n})$.[2] See Appendix A for the proof.

Note that we would get the exact same result if we answer each query one after the other and compute the total privacy budget using the basic composition theorem. The advantage of this latter approach is that it holds even when the queries are chosen *adaptively*, i.e., the output of one query is used to inform the next queries.

# 3 Approximate Differential Privacy

It turns out that above analysis is tight: one can show that *any* algorithm that is $\varepsilon$-differentially private and answers $k$ normalized counting queries must incur error of magnitude $\Omega({}^{k}/_{\varepsilon n})$ (this result is due to [HT10], see [Vad17, §5.2] for a nice overview).

Unless we relax our notion of privacy, we cannot do any better. So that's exactly what we'll do! Notice that the definition of DP asks that *any* event can be at most $e^{\varepsilon}$ times more likely if we swap out one individual. This includes, in particular, events with a tiny probability. For example, suppose that some $y$ has probability $2^{-128}$ of being output on database $D$ and probability 0 on database $D'$. Of course, observing $y$ is a "smoking gun" that the input database cannot have been $D$, and so this algorithm is not $\varepsilon$-DP for any finite $\varepsilon$. And yet, should we really care about this? Our cryptographic schemes also fail with similar probability.

This motivates the following relaxation of DP, where we allow events to also deviate by a small *additive* factor $\delta$:

**Definition 4** (Approximate Differential Privacy [DKM+06])**.** An algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private if, for all neighboring databases $D$ and $D'$ and all events $S \subseteq \mathcal{Y}$, we have

$$\Pr[M(D) \in S] \le e^{\varepsilon} \cdot \Pr[M(D') \in S] + \delta \,.$$

**Interpreting approximate DP.**

- $(\varepsilon, 0)$-DP is equivalent to the "pure" $\varepsilon$-DP definition from the previous lecture.
- We can think of $\delta$ as the "failure probability" of the DP guarantee, and (informally) interpret $(\varepsilon, \delta)$-DP as saying that the algorithm is $\varepsilon$-DP with probability at least $1 - \delta$, while providing no guarantee with probability at most $\delta$. However, this analogy can be

[1] We could think of these as multiple subset queries as in the lecture on reconstruction attacks, or as counting queries over different columns of a database.

[2] The notation $\tilde{O}$ hides log factors in the asymptotic notation.

overly pessimistic, and many algorithms that are $(\varepsilon, \delta)$-DP (such as the Gaussian mechanism below) actually degrade "gracefully". That is, with probability $\delta$, the algorithm still provides some DP guarantee, albeit with a larger $\varepsilon$.

- Ideally, we would set $\delta$ to be "cryptographically small" (e.g., $2^{-80}$). In practice, however, we might set $\delta$ to be much larger because of the cost of composing many differentially private algorithms. For example, differentially private machine learning (which we will see later) commonly uses values such as $\delta = 1/n^2$.
- Notably, as you'll see in the homework, the factor $\delta$ *must* be $o(1/n)$ to get a meaningful privacy guarantee.
- Recall from the last lecture that when dealing with pure DP, the definition was equivalent (for discrete outputs) to just asking that $\Pr[M(D) = y] \leq e^{\varepsilon} \cdot \Pr[M(D') = y]$ for all $y \in \mathcal{Y}$. This is *not* the case for approximate DP, where we have to consider all possible subsets of outputs rather than just single outputs (see [DR14] for details).

**Properties of approximate DP.** Approximate DP has similar properties to pure DP. In particular, $(\varepsilon, \delta)$-DP is closed under postprocessing and scales gracefully to larger groups (while the multiplicative factor $\varepsilon$ still scales linearly with the group size, the scaling of $\delta$ is a bit weirder, and we won't dwell on it here, see [DR14] or [Vad17]).

Approximate DP also satisfies a basic composition theorem:

**Theorem 2** (Basic composition of approximate DP)**.** Let $M = (M_1, M_2, \dots, M_k)$ be a sequence of algorithms, where $M_i$ is $(\varepsilon_i, \delta_i)$-differentially private, and the algorithms can be chosen sequentially and adaptively. Then $M$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

We won't prove this one here. As we'll see in a later lecture, approximate DP actually allows for a much more powerful composition theorem, where the factor $\varepsilon$ scales as $O(\sqrt{k})$ instead of $O(k)$. This is a crucial advantage of approximate DP over pure DP: if we are composing many different differentially private algorithms, we can get away with a much smaller $\varepsilon$ than with pure DP.

## 4 The Gaussian Mechanism

The nice thing about approximate DP is that it will allow us to work with Gaussian noise, which is much more convenient than Laplace noise (for reasons we won't get into here).

Due to the geometry of the (multivariate) Gaussian distribution, we will now consider functions that have small sensitivity in the $\ell_2$ norm rather than the $\ell_1$ norm.

**Definition 5** ($\ell_2$-Sensitivity)**.** The $\ell_2$-sensitivity of a function $f \colon \mathcal{X}^n \to \mathbb{R}^k$ is defined as

$$\Delta_2 = \max_{D \sim D'} \|f(D) - f(D')\|_2 = \max_{D, D'} \sqrt{\sum_{i=1}^{k} (f(D)_i - f(D')_i)^2}\,,$$

where $D$ and $D'$ are neighboring databases.

An important thing to note is that the $\ell_2$ norm is never more than the $\ell_1$ norm, and can be as much as $\sqrt{k}$ times smaller:

$$\Delta_2 \leq \Delta_1 \leq \sqrt{k} \cdot \Delta_2$$

In particular, suppose that $f = (f_1, \ldots, f_k)$ is a sequence of $k$ counting queries. Then $f$ has $\ell_1$-sensitivity $\Delta_1 = k/n$, while the $\ell_2$-sensitivity is $\Delta_2 = \sqrt{k}/n$. As we will see, the Gaussian mechanism will let us scale the noise proportional to $\Delta_2$ (and not $\Delta_1$), and thus achieve a much better error.

**Definition 6** (Gaussian Distribution)**.** The univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean $\mu$ and variance $\sigma^2$ has density

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad x \in \mathbb{R}\,.$$

**Definition 7** (Gaussian Mechanism)**.** Let $f \colon \mathcal{X}^n \to \mathbb{R}^k$ be a function with $\ell_2$-sensitivity $\Delta_2$, and let $\varepsilon, \delta > 0$. The Gaussian mechanism is defined as

$$M(D) = f(D) + (Y_1, \ldots, Y_k)$$

where the $Y_i$ are i.i.d. samples from the $\mathcal{N}(0, 2\log(1.25/\delta)\Delta_2^2/\varepsilon^2)$ distribution.

The values are a bit uglier here than for the Laplace mechanism, so let's quickly unwrap this. The standard deviation of the noise grows linearly with $\Delta/\varepsilon$ as for the Laplace mechanism, but we now use the $\ell_2$-sensitivity instead of the $\ell_1$-sensitivity, and we have an extra $O(\log(1/\delta))$ factor. So, in the univariate case (where $\Delta_2 = \Delta_1$), the Gaussian mechanism always adds noise of higher variance than the Laplace mechanism. The true benefit will come from answering multiple queries. Also note that the noise dependence on $\delta$ is only logarithmic, so we can set $\delta$ to be exponentially small while only incurring a constant multiplicative factor in the error.

Somewhat unsurprisingly at this stage:

**Theorem 3.** The Gaussian mechanism is $(\varepsilon, \delta)$-differentially private.

We won't prove this here as the proof is a bit more involved than the one we saw for the Laplace mechanism. The main idea is to show that the densities of the Gaussian mechanism on neighboring databases differ by a factor less than $e^\varepsilon$ unless the Gaussian noise exceeds a certain threshold. And in turn, the probability that the noise exceeds this threshold is at most $\delta$ by concentration of the Gaussian distribution (see e.g., these notes).

Another way of writing Theorem 3 is that, if we add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the output of a function $f$ with $\ell_2$-sensitivity $\Delta_2$, then the resulting function is $(\varepsilon, \delta(\varepsilon))$-differentially private for every $\varepsilon > 0$, where $\delta(\varepsilon) = 1.25\exp(-\varepsilon^2\sigma^2/2\Delta_2^2)$. Thus, the same mechanism can be interpreted as providing many different $(\varepsilon, \delta)$-DP guarantees. This can make the analysis of the optimal composition of such mechanisms somewhat tricky; hence, there also exist definitions that more cleanly characterize the privacy of such mechanisms (e.g., Rényi DP [Mir17]).

### 4.1 Answering Counting Queries With the Gaussian Mechanism

So what have we gained? Let's go back to our (multiple) counting queries example, where we apply $k$ normalized counting functions $f = (f_1, \ldots, f_k)$ to the same dataset. The $\ell_1$-sensitivity of the function $f$ is $\Delta_1 = k/n$, while the $\ell_2$-sensitivity is $\Delta_2 = \sqrt{k}/n$.

So to get the same privacy level $\varepsilon$ as with the Laplace mechanism (but with the additional $\delta$ probability of error), we need to add Gaussian noise of standard deviation $\tilde{O}(\Delta_2/\varepsilon) = \tilde{O}(\sqrt{k}/\varepsilon n)$ to each coordinate of the output (we ignore the $\log(1/\delta)$ factor here for simplicity). We thus get error $\tilde{O}(\sqrt{k}/\varepsilon n)$ in each coordinate with high probability—a factor of $\sqrt{k}$ smaller than the error of the Laplace mechanism! It turns out that this is also the best we can do, for arbitrary counting queries.

## 5 So How Many Counting Queries Can We Answer Privately?

The table below summarizes the privacy-utility trade-offs for answering $k$ counting queries, along with matching lower bounds (ignoring logarithmic factors in $\delta$ and $k$). A few notes:

- The Randomized Response mechanism we saw had a fixed privacy budget of $\varepsilon = 3$, but this can be generalized to any $\varepsilon > 0$ with an appropriate probability of flipping the answer.
- The error bound for the Gaussian mechanism can further be beaten in settings where the data domain $|\mathcal{X}|$ is not too large. The state-of-the-art here is the *Private Multiplicative Weights* algorithm [HR10] which we won't cover. This algorithm's error has only a logarithmic dependence on the number of queries $k$, and can thus answer *exponentially many* queries! (as long as the data domain is not exponentially larger than the database size $n$).

| Guarantee | Error per query | Mechanism | Lower bound |
|---|---|---|---|
| Local $\varepsilon$-DP | $O\left(\frac{1}{\varepsilon\sqrt{n}}\right)$ | Randomized Response | $\Omega\left(\frac{1}{\varepsilon\sqrt{n}}\right)$ |
| $\varepsilon$-DP | $\tilde{O}\left(\frac{k}{\varepsilon n}\right)$ | Laplace | $\Omega\left(\frac{k}{\varepsilon n}\right)$ |
| $(\varepsilon,\delta)$-DP | $\tilde{O}\left(\frac{\sqrt{k}}{\varepsilon n}\right)$ | Gaussian | $\Omega\left(\frac{\sqrt{k}}{\varepsilon n}\right)$ |
| | $\tilde{O}\left(\frac{\sqrt{\log\lvert\mathcal{X}\rvert}\,\log k}{\varepsilon n}\right)^{1/2}$ | Private Multiplicative Weights | $\tilde{\Omega}\left(\frac{\sqrt{\log\lvert\mathcal{X}\rvert}\,\log k}{\varepsilon n}\right)^{1/2}$ |

# References


[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third*

*Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[HR10] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 61–70. IEEE, 2010.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714, 2010.

[Mir17] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[Vad17] Salil Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017.

# A Bounding the Error of DP Algorithms

Our analysis of the error bounds of various mechanisms glossed over some details of how to apply concentration bounds. Let's formalize these things here.

**Randomized Response.** In the last lecture, we argued that the error of the randomized response algorithm is $O(1/\sqrt{n})$ with high probability, and hinted at a Chernoff/Hoeffding bound argument (see the background sheet for a description on these bounds).

Without loss of generality, assume that all responses are zero, i.e., $x_i = 0, \forall i$ (it's easy to see that this is the worst case, as individual errors cannot "cancel out"). The error of the randomized response algorithm is then simply $\hat{p} = \frac{2}{n}\sum_{i=1}^{n} y_i - \frac{1}{2} = \sum_{i=1}^{n} \frac{2}{n}(y_i - \frac{1}{4})$. This is the sum of $n$ i.i.d. Bernoulli random variables taking values in $[-1/2n, 3/2n]$. Also, recall that we showed that $\mathbb{E}[\hat{p}] = p = 0$. So we have that:

$$\begin{aligned}\Pr[|\hat{p}| \geq t] &\leq 2\exp\left(-\frac{2t^2}{n \cdot (2/n)^2}\right) \\ &= 2\exp\left(-nt^2/2\right) \ .\end{aligned}$$

We wanted to show that errors larger than $O(1/\sqrt{n})$ are unlikely. And indeed, for any constant $\lambda > 0$, we have that:

$$\Pr[|\hat{p}| \geq \lambda/\sqrt{n}] \leq 2e^{-\lambda^2/2} \ ,$$

which drops off exponentially fast in $\lambda$.

**Laplace or Gaussian noise for one query.** In Section 2.1, we said that if we add Laplace noise of scale $b = O(1/\varepsilon n)$ to a counting query, then the error is $O(1/\varepsilon n)$ with high probability. This follows directly from the following tail bound for the Laplace distribution:

> If $Z \sim \texttt{Laplace}(0, b)$, then for every $t > 0$: $P(|Z| \geq t \cdot b) \leq \exp(-t)$.

So for constant $\lambda > 0$, the error exceeds $\lambda b = O(1/\varepsilon n)$ with probability at most $e^{-\lambda}$.

The Gaussian distribution has an even sharper tail bound:

> If $Z \sim \mathcal{N}(0, \sigma^2)$, then for every $t > 0$: $P(|Z| \geq t \cdot \sigma) \leq 2\exp(-t^2/2)$.

**Laplace or Gaussian noise for multiple queries.** In Section 2.1 and Section 4.1, we said that if we answer $k$ counting queries with the Laplace or Gaussian mechanism, the maximum error is only a factor $O(\log k)$ larger.

We can derive this from a union bound. We focus on the Laplace mechanism here. Let $Z_1, \dots, Z_k$ be the Laplace noise with scale $b = O(k/\varepsilon n)$ added for the $k$ queries. Combining the above tail bound with a union bound, we have that:

$$\Pr[\max_i |Z_i| \geq t \cdot b] \leq k\exp(-t) \ .$$

Setting $t = \lambda + \log k$ for a constant $\lambda > 0$, we get that:

$$\Pr[\max_i |Z_i| \geq \tilde{O}(k/\varepsilon n)] \leq k\exp(-(\lambda + \log k)) = e^{-\lambda} \ .$$