Snapping Mechanism and Problems of Finite Precision

Christian Covington

September 30, 2019

Harvard University Privacy Tools Project

Overview

Problem Statement

IEEE 754 Floating Point

Problems Implementing the Laplace Mechanism

Snapping Mechanism

Implementation Considerations

Utility Analysis

Introduce Laplace, that the promises break down when moving to implementation, and Mironov/Snapping

Problem Statement

What is Differential Privacy and how do we achieve it?

Let $M : \mathcal{X}^n \to \mathcal{R}$ be a randomized algorithm, D and D' be neighboring data sets (differing in one row), and $S \subseteq \mathcal{R}$. Then M satisfies (ϵ, δ) differential privacy if

$$\mathbb{P}(M(D) \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(M(D') \in S) + \delta \text{ [DMNS06]}$$

What is Differential Privacy and how do we achieve it?

Let $M : \mathcal{X}^n \rightarrow \mathcal{R}$ be a randomized algorithm, D and D' be neighboring data sets (differing in one row), and $S \subseteq \mathcal{R}$. Then M satisfies (ϵ, δ) differential privacy if

$$\mathbb{P}(M(D) \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(M(D') \in S) + \delta \text{ [DMNS06]}$$

We will focus on the Laplace Mechanism, which satisfies $(\epsilon, 0)$ differential privacy:

$$M_{Lap}(\mathcal{D}, f, \epsilon) = f(\mathcal{D}) + Lap\left(\frac{\Delta f}{\epsilon}\right) \text{ [DMNS06]}$$

where $f : \mathcal{D} \rightarrow \mathbb{R}$.

What is Differential Privacy and how do we achieve it?

Let $M : \mathcal{X}^n \to \mathcal{R}$ be a randomized algorithm, D and D' be neighboring data sets (differing in one row), and $S \subseteq \mathcal{R}$. Then M satisfies (ϵ, δ) differential privacy if

$$\mathbb{P}(M(D) \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(M(D') \in S) + \delta \text{ [DMNS06]}$$

We will focus on the Laplace Mechanism, which satisfies $(\epsilon, 0)$ differential privacy:

$$M_{Lap}(\mathcal{D}, f, \epsilon) = f(\mathcal{D}) + Lap\left(\frac{\Delta f}{\epsilon}\right) \text{ [DMNS06]}$$

where $f : \mathcal{D} \to \mathbb{R}$.

For $(\epsilon, 0)$-DP, it is necessary (but not sufficient) for $\operatorname{supp}(M(D)) = \operatorname{supp}(M(D'))$.

Moving from Theory to Practice

Consider additive noise N. When $\mathrm{supp}(N) = \mathbb{R}$, the supports of mechanism outputs on neighboring data sets are equivalent. This is not necessarily true when $\mathrm{supp}(N) \neq \mathbb{R}$.[1]

Throughout the presentation, we will refer to an "idealized mechanism" as a mechanism that has access to infinite precision.

[1] E.g. let $f(D) = 0, f(D) = \frac{1}{2}$, and $\mathrm{supp}(N) = \mathbb{Z}$.

We will be considering IEEE-754 double-precision floating point numbers.

IEEE 754 Floating Point

IEEE 754 Floating Point

The IEEE 754 standard (referred to as *double* or *binary64*) floating point number has 3 components:
sign: 1 bit
significand/mantissa: 53 bits (only 52 are explicitly stored)
exponent: 11 bits

Let S be the sign bit, $m_1 \ldots m_{52}$ be the bits of the mantissa, and $e_1 \ldots e_{11}$ be the bits of the exponent. Then a double is represented as

$$(-1)^S (1.m_1 \ldots m_{52})_2 \times 2^{(e_1 \ldots e_{11})_2 - 1023}$$

Note that doubles ($\mathbb{D}$) are not uniformly distributed over their range, so arithmetic precision is not constant across $\mathbb{D}$.

Problems Implementing the Laplace Mechanism

Generating the Laplace: Overview

The most common method of generating Laplace noise is to use inverse transform sampling. Let Y be the random variable representing our Laplace noise with scale parameter λ. Then,

$$Y \leftarrow F^{-1}(U) = -\lambda ln(1-U)$$

where F^{-1} is the inverse cdf of the Laplace and $U \sim Unif(0,1)$.

We can reduce sampling from the Laplace to thinking about how uniform random number generation and arithmetic operations differ on $\mathbb{D}$ as opposed to $\mathbb{R}$.

Sampling from Uniform

Sampling from $\mathbb{D} \cap (0, 1)$ is not particularly well-defined or consistent across implementations. Typically, the output of a uniform random sample is confined to a small subset of possible elements of $\mathbb{D}$. [Mir12]

Reference and Library	Uniform from $[0, 1)$
Knuth [Knu97]	multiples of 2^{-53}
"Numerical Recipes" [PTVF07]	multiples of 2^{-64}
C#	multiples of $1/(2^{31} - 1)$
SSJ (Java) [L'E]	multiples of 2^{-32} or 2^{-53}
Python	multiples of 2^{-53}
OCaml	multiples of 2^{-90}

Figure 1: Uniform random number generation [Mir12]

Already, see that the set of possible draws from Laplace will differ by implementation.

Natural Logarithm

When implemented on uniform random numbers as normally generated, the natural log produces some values repeatedly and skips over others entirely. [Mir12]

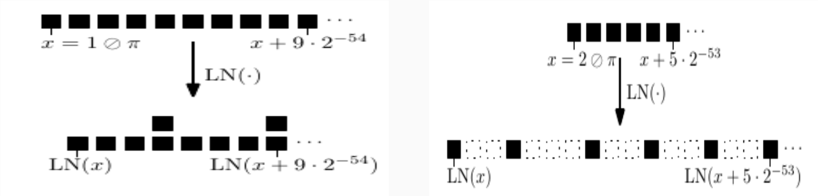


Figure 2: Artefacts of natural logarithm on $\mathbb{D}$ [Mir12]

Attack

Imagine we want to release a private version of the output of a function f with $\Delta f = 1$ and $\epsilon = \frac{1}{3}$. Let $f(D) = 0, f(D') = 1$.

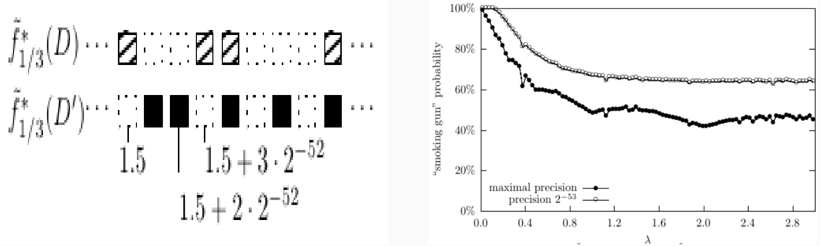


Figure 3: Attack on Laplace Mechanism [Mir12]

Mironov performed an attack on PINQ, reconstructing 18K records in fewer than 1000 queries with total $\epsilon < 10^{-6}$

Figure shows probability that output is in the support of only one of the data sets after 1 release.

This is a lower bound on the probability that the DP guarantee is broken (this is effectively showing that $\delta \neq 0$, but it could also be that ϵ is too low).

White circles represent a common sampling method, black circles are from full $\mathbb{D} \cap (0, 1)$.

Note that attack is still reasonably effective for large λ (low purported privacy loss). This allows an attacker to slowly use their privacy budget and possibly reconstruct an entire database.

Inadequate Fixes

- Rounding Noise?
  - Consider rounding noise to the nearest integer multiple of 2^{-32}. Then, if $|f(D) - f(D')| < 2^{-32}$, then the supports of the mechanism outputs under the two data sets are completely disjoint.
- Smoothing Noise?
  - If $f(D), f'(D)$ are in different bands of precision, the support of the mechanism on one will be a proper subset of the support of the mechanism on the other.

Snapping Mechanism

Mechanism Statement

The Snapping Mechanism [Mir12] is defined as follows:

$$\tilde{f}(D) \triangleq clamp_B\left(\lfloor clamp_B(f(D)) \oplus S \otimes \lambda \otimes LN(U^*)\rceil_\Lambda\right)$$

where $clamp_B$ restricts output to the range $[-B, B]$, $S \otimes \lambda \otimes LN(U^*)$ is Laplace noise generated with our improved random number generator (more on this later), and $\lfloor \cdot \rceil_\Lambda$ rounds to the nearest Λ, where Λ is the smallest power of two at least as large as λ.

The mechanism guarantees $\left(\frac{1+12B\eta+2\eta\lambda}{\lambda}, 0\right)$-DP, where η is machine epsilon.

Explain all the floating point symbols.

Mechanism Motivation/Explanation

$$\tilde{f}(D) \triangleq clamp_B\left(\lfloor clamp_B(f(D)) \oplus S \otimes \lambda \otimes LN(U^*)\rceil_\Lambda\right)$$

Let $\tilde{F}(\cdot)$ be the idealized version of the snapping mechanism. Then $\tilde{F}(\cdot)$ satisfies $(\epsilon, 0)$-DP. For a given $x \in supp(\tilde{F}(D))$, consider:

- $[L, R) \subset (0, 1)$ is the set mapped to x by $\tilde{F}(D)$
- $[l, r) \subset (\mathbb{D} \cap (0, 1))$ is the set mapped to x by $\tilde{f}(D)$

The sampling mechanism, exact rounding, and clamping ensure that $|R - L| \approx |r - l|$ in terms of relative error, which yields the DP-guarantee of $\tilde{f}(D)$.

$clamp_B$ is stable $|x - y| \leq c \implies |clamp_B(x) - clamp_B(y)| \leq c$, so the inner clamping preserves privacy guarantees.

Rounding and outer clamping are considered post-processing.

Implementation Considerations

Generating Uniform Random Numbers

Our goal is to sample from $\mathbb{D} \cap (0,1)$ while maintaining the properties of $\mathbb{R}$ as closely as possible.

IEEE 754 floating point numbers are of the form

$$(-1)^S (1.m_1 \ldots m_{52})_2 \times 2^{-E}$$

Let:
$S = 0$
$E \sim Geom(p = 0.5)$
$\forall i \in \{1, \ldots, 52\} : m_i \sim Bern(p = 0.5)$.

This means that every $d \in \mathbb{D} \cap (0,1)$ has a chance of being represented, and each is represented proportional to its unit of least precision. In order to sample from $\mathbb{D} \cap (0,1)$ in this way, we need only be able to generate cryptographically secure random bits.

Exact Calculations

Multiple points in the algorithm require exact (rather than accurate-faithful) rounding.

Arithmetic with the natural logarithm is done with 118 bits of precision as described in [DLM07] to ensure exact rounding.

All rounding is done via direct manipulation of the floating-point representation of the number.

Consider that for an arbitrary $x \in \mathbb{D}$ the natural log of x is not necessarily $\in \mathbb{D}$. Let $a < ln(x) < b$ where $a, b \in \mathbb{D}$ and $\nexists c \in \mathbb{D} : a < c < b$. Without loss of generality, assume that $|a - x| < |b - x|$, so that if we had infinite precision in calculating $ln(x)$ (but still had to output an element $\in \mathbb{D}$), we would output a. Many mathematical libraries do what is called *accurate-faithful* rounding, which means that in the scenario above our algorithm would output a with high probability. In an *exact rounding* paradigm, the algorithm outputs a with probability 1.

Ensuring correct functional ϵ

The mechanism guarantees $(\epsilon(1 + 12B\eta + 2\eta), 0)$-DP (relative to the nominal $(\epsilon, 0)$-DP if you were to use the Laplace Mechanism).

We want the Snapping Mechanism's guarantee to be $(\epsilon, 0)$-DP.

Ensuring correct functional ϵ

The mechanism guarantees $(\epsilon(1 + 12B\eta + 2\eta), 0)$-DP (relative to the nominal $(\epsilon, 0)$-DP if you were to use the Laplace Mechanism).

We want the Snapping Mechanism's guarantee to be $(\epsilon, 0)$-DP.

Rewrite Laplace inside of the Snapping Mechanism needs to be written as if it respects $\left(\frac{\epsilon - 2\eta}{1+12B\eta}, 0\right)$-DP.

We will refer to this rescaled ϵ as ϵ' and rewrite the Laplace random variable as Y'.

Utility Analysis

Difficulty of utility analysis

Snapping Mechanism:

$$\tilde{f}(D) \triangleq clamp_B\left(\lfloor clamp_B(f(D)) \oplus S \otimes \lambda' \otimes LN(U^*)\rceil_{\Lambda'}\right)$$

What can we say about $|f(D) - \tilde{f}(D)|$?

- if user sets B poorly (e.g. $|B| << |f(D)|$), then $|f(D) - \tilde{f}(D)|$ could be arbitrarily bad
  - We are currently setting B within the mechanism, rather than leaving it to the user
- $\lfloor \cdot \rceil_{\Lambda'}$ makes distribution of noise more difficult to reason about (becomes dependent on $f(D)$)
  - We will make conservative statements based on worst-case

Talk about why setting B automatically helps both for empirical utility and the theoretical analysis.

Automatically setting B

Every statistic in *PSI* that uses the Laplace Mechanism asks the user for bounds on the range of their data. We can use these to get a maximum possible value for each statistic.[2]

User provides $[D_{min}, D_{max}]$ as upper/lower bounds on the min/max value of D.[3] For a given statistic $T(\cdot)$ we set B such that

$$\forall D \text{ s.t. } \min(D) \geq D_{min} \text{ and } \max(D) \leq D_{max} : B \geq \max T(D)$$

This prevents the inner clamping bound from binding, so we rewrite the mechanism as

$$\tilde{f}(D) \triangleq clamp_B\left(\lfloor f(D) \oplus Y' \rceil_{\Lambda'}\right).$$

[2]Not immediately clear to me whether or not we would be able to do this in general.

[3]Values outside of this range are clipped.

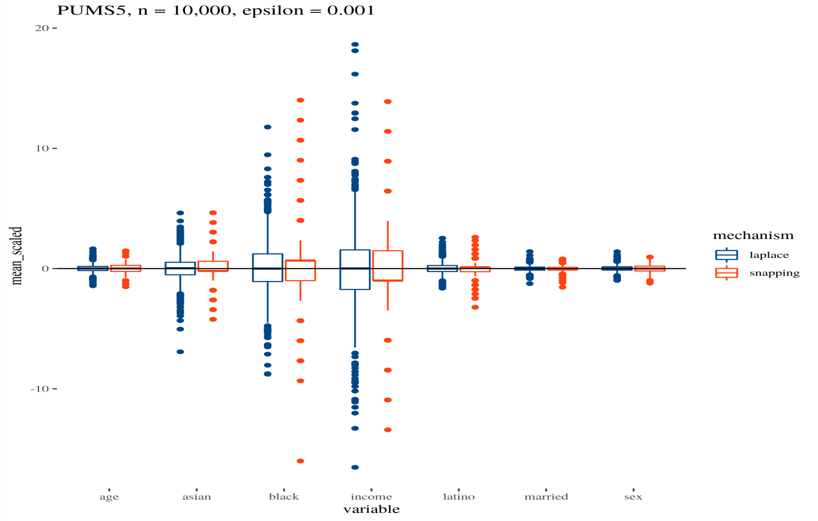
Error

Assume $\Delta f = 1$. We define error to be the absolute difference between the true statistic and our mechanism release. We want to compare Snapping error vs Laplace error:

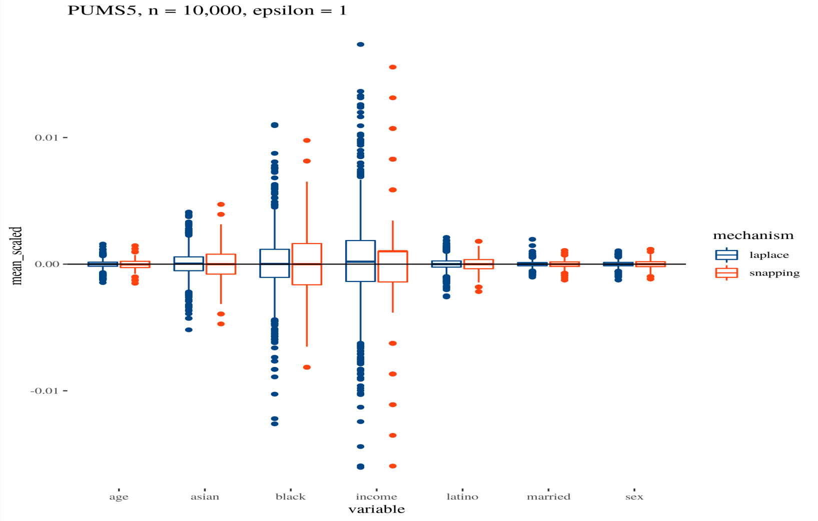
$$\begin{aligned}\left|f(D) - \lfloor f(D) + Y' \rceil_{\Lambda'}\right| &\leq \left|f(D) - (f(D) + Y')\right| + \left|f(D) + Y' - \lfloor f(D) + Y' \rceil_{\Lambda'}\right| \\ &\leq |-Y'| + \frac{\Lambda'}{2} \\ &= |Y'| + \frac{\Lambda'}{2}\end{aligned}$$

Noting that $Y' = \frac{\epsilon}{\epsilon'} Y$, we have that, conditional on a privacy loss parameter ϵ, the Snapping error is at most $\frac{\epsilon(1+12B\eta)}{\epsilon - 2\eta} y + \frac{\Lambda'}{2}$ for a given amount of Laplace error y.

Empirical Utility Testing - $\epsilon = 0.001$



Empirical Utility Testing - $\epsilon = 1$



Accuracy (part 1)

Let $Z = |Y'| + \frac{\Lambda'}{2}$ (the Snapping error) and F_Z its CDF.

$$\begin{aligned} F_Z(z) &= \mathbb{P}(Z \leq z) \\ &= \mathbb{P}\left(|Y'| + \frac{\Lambda'}{2} \leq z\right) \\ &= \mathbb{P}\left(|Y'| \leq z - \frac{\Lambda'}{2}\right) \\ &= 1 - \exp\left(-\epsilon'(z - \frac{\Lambda'}{2})\right) \end{aligned}$$

Accuracy (part 2)

For a given α, let accuracy be the a such that $\alpha = \mathbb{P}(Z > a)$.

$$\begin{aligned}
\mathbb{P}(Z > a) &= 1 - \mathbb{P}(Z \leq a) \\
&= 1 - F_Z(a) \\
&= \exp\left(-\epsilon'(a - \frac{\Lambda'}{2})\right)
\end{aligned}$$

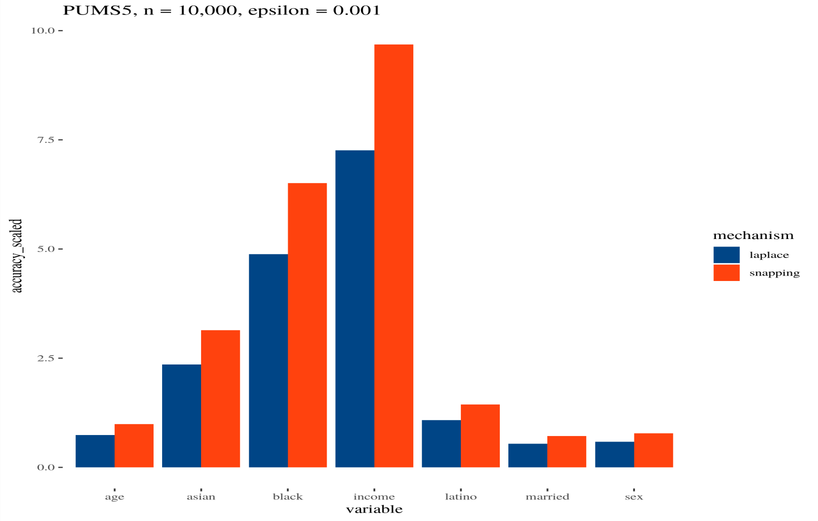
So, we have $a_{Snapping} = \frac{\ln(\frac{1}{\alpha})}{\epsilon'} + \frac{\Lambda'}{2}$, compared to $a_{Laplace} = \frac{\ln(\frac{1}{\alpha})}{\epsilon}$.

Accuracy (part 3)

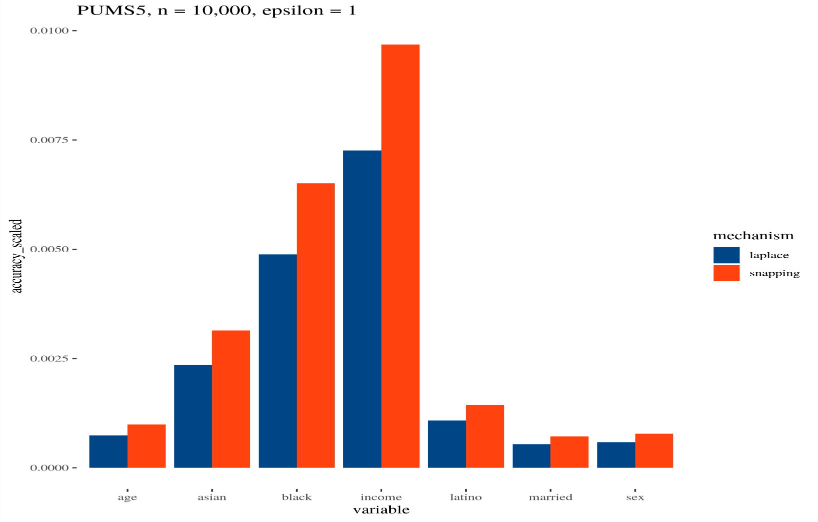
Recalling that $\epsilon' = \frac{\epsilon - 2\eta}{1+12B\eta}$ we can represent the difference between the accuracy of the Snapping and Laplace mechanisms as follows:

$$\begin{aligned}
a_{Snapping} &= \frac{\ln\left(\frac{1}{\alpha}\right)}{\epsilon'} + \frac{\Lambda'}{2} \\
&= \frac{\ln\left(\frac{1}{\alpha}\right)}{\left(\frac{\epsilon-2\eta}{1+12B\eta}\right)} + \frac{\Lambda'}{2} \\
&= \frac{(1+12B\eta)\left(\ln\left(\frac{1}{\alpha}\right)\right)}{\epsilon - 2\eta} + \frac{\Lambda'}{2} \\
&= \frac{\epsilon(1+12B\eta)}{\epsilon - 2\eta} \cdot \left(\frac{\ln\left(\frac{1}{\alpha}\right)}{\epsilon}\right) + \frac{\Lambda'}{2} \\
&= \frac{\epsilon(1+12B\eta)}{\epsilon - 2\eta} \cdot a_{Laplace} + \frac{\Lambda'}{2}
\end{aligned}$$

Accuracy Testing - $\epsilon = 0.001$



Accuracy Testing - $\epsilon = 1$



Bias

We write the bias of the snapping mechanism as

$$Bias = \mathbb{E}(\tilde{f}(D) - f(D)) = \mathbb{E}\left(clamp_B\left(\lfloor f(D) + Y' \rceil_{\Lambda'}\right) - f(D)\right)$$

where $Y' \sim Laplace(\lambda')$ and the expectation is over the randomness of the snapping mechanism.

Now, we define an upper bound on the Bias:

$$Bias^{+} = \mathbb{E}\left(clamp_B\left(f'(D) + Y^{*}\right) - \hat{f}(D)\right)$$

where $Y^{*} \sim Laplace(-\frac{\Lambda}{2}, \lambda')$.

Now, define the following:

$$p_L = F_{Y^*}(-B - \hat{f}(D))$$

$$p_U = 1 - F_{Y^*}(B - \hat{f}(D))$$

such that F_{Y^*} is the CDF of Y^* and p_L, p_U are the probabilities that the lower/upper bounds are binding (respectively). Then we can write

$$Bias^{+} = p_L \cdot (-B - \hat{f}(D)) + p_U \cdot (B - \hat{f}(D)) + (1 - p_l - p_U) \cdot \int_{-B-\hat{f}(D)}^{B-\hat{f}(D)} y^{*} f(y^{*}) dy^{*}$$

where f is the PDF of Y^*.

Possible Next Steps

- Continue integration into PSI
- More considered choice of B
- Tighter accuracy bounds
- Extend to mechanisms other than Laplace

[DLM07] Florent de Dinechin, Christoph Lauter, and Jean-Michel Muller.
Fast and correctly rounded logarithms in double-precision.
RAIRO - Theoretical Informatics and Applications, 41(1):85–102, 2007.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith.
Calibrating noise to sensitivity in private data analysis.

In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[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*, CCS '12, pages 650–661, New York, NY, USA, 2012. ACM.