Preserving differential privacy under finite-precision semantics *

Ivan Gazeau, Dale Miller, and Catuscia Palamidessi
INRIA and LIX, Ecole Polytechnique

The approximation introduced by finite-precision representation of continuous data can induce arbitrarily large information leaks even when the computation using exact semantics is secure. Such leakage can thus undermine design efforts aimed at protecting sensitive information. We focus here on differential privacy, an approach to privacy that emerged from the area of statistical databases and is now widely applied also in other domains. In this approach, privacy is protected by the addition of noise to a true (private) value. To date, this approach to privacy has been proved correct only in the ideal case in which computations are made using an idealized, infinite-precision semantics. In this paper, we analyze the situation at the implementation level, where the semantics is necessarily finite-precision, i.e. the representation of real numbers and the operations on them, are rounded according to some level of precision. We show that in general there are violations of the differential privacy property, and we study the conditions under which we can still guarantee a limited (but, arguably, totally acceptable) variant of the property, under only a minor degradation of the privacy level. Finally, we illustrate our results on two cases of noise-generating distributions: the standard Laplacian mechanism commonly used in differential privacy, and a bivariate version of the Laplacian recently introduced in the setting of privacy-aware geolocation.

Keywords: Differential privacy, floating-point arithmetic, robustness to errors.

1 Introduction

It is well known that, due to the physical limitations of actual machines, in particular the finiteness of their memory, real numbers and their operations cannot be implemented with full precision. While for traditional computation getting an approximate result is not critical when a bound on the error is known, we argue that, in security applications, the approximation error can became a fingerprint potentially causing the disclosure of secrets.

Obviously, the standard techniques to measure the security breach do not apply, because an analysis of the system in the *ideal* (aka *exact*) semantics does not reveal the information leaks caused by the implementation. Consider, for instance, the following simple program

$$\text{if } f(h) > 0 \text{ then } \ell = 0 \text{ else } \ell = 1$$

where h is a high (i.e., confidential) variable and ℓ is a low (i.e., public) variable. Assume that h can take two values, v_1 and v_2, and that both $f(v_1)$ and $f(v_2)$ are strictly positive. Then, in the ideal semantics, the program is perfectly secure, i.e. it does not leak any information. However, in the implementation, it could be the case that the test succeeds in the case of v_1 but not in the case of v_2 because, for instance, the value of $f(v_2)$ is below the smallest representable positive number. Hence, we would have a total disclosure of the secret value.

*This work has been partially supported by the project ANR-09-BLAN-0345-02 CPP, by the INRIA Action d'Envergure CAPPRIS, and by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS).

Submitted to:
QAPL 2013

© I. Gazeau, D. Miller, and C. Palamidessi
This work is licensed under the
Creative Commons Attribution License.

The example above is elementary but it should give an idea of the pervasive nature of the problem, which can have an impact in any confidentiality setting, and should therefore receive attention by those researchers interested in (quantitative) information flow. In this paper, we initiate this investigation with an in-depth study of the particular case of *differential privacy*.

Differential privacy [9, 10] is an approach to the protection of private information that originated in the field of statistical databases and is now investigated in many other domains, ranging from programming languages [3, 11] to social networks [18] and geolocation [15, 13, 2]. The key idea behind differential privacy is that whenever someone queries a dataset, the reported answer should not allow him to distinguish whether a certain individual record is in the dataset or not. More precisely, the presence or absence of the record should not change significantly the probability of obtaining a given answer. The standard way of achieving such a property is by using an *oblivious mechanism*[1] which consists in adding some noise to the true answer. Now the point is that, even if such a mechanism is proved to provide the desired property in the ideal semantics, its implementation may induce errors that alter the least significant digits of the reported answer and cause significant privacy breaches. Let us illustrate the problem with an example.

Example 1.1. Consider the simplest representation of reals: the fixed-point numbers. This representation is used on low-cost processors which do not have floating-point arithmetic module. Each value is stored in a memory cell of fixed length. In such cells, the last d digits represent the fractional part. Thus, if the value (interpreted as an integer) stored in the cell is z, its semantics (i.e., the true real number being represented) is $z \cdot 2^{-d}$.

To grant differential-privacy, the standard technique consists of returning a random value with probability $p(x) = 1/2b \cdot e^{-|x-r|/b}$ where r is the true result and b is a scale parameter which depends on the degree of privacy to be obtained and on the sensitivity of the query. To get a random variable with any specific distribution, in general, we need to start with an initial random variable provided by a primitive of the machine with a given distribution. To simplify the example, we assume that the machine already provides a Laplacian random variable X with a scale parameter 1. The probability distribution of such an X is $p_X(x) = 1/2e^{-|x|}$. Hence, if we want to generate the random variable bX with probability distribution $p_{bX}(x) = 1/2b \cdot e^{-|x|/b}$, we can just multiply by b the value $x = z \cdot 2^{-d}$ returned by the primitive.

Assume that we want to add noise with a scale parameter $b = 2^n$ for some fixed integer n (b can be big when the sensitivity of the query and the required privacy degree are high). In this case, the multiplication by 2^n returns a number $2^n z \cdot 2^{-d}$ that, in the fixed-point representation terminates with n zeroes. Hence, when we add this noise to the true result, we return a value whose representation has the same n last digits as the secret. For example, assume $b = 2^2 = 4$ and $d = 6$. Consider that the true answers are $r_1 = 0$ and $r_2 = 1 + 2^{-5}$. In the fixed-point representation, the last two digits of r_1 are 00, and the last two digits of r_2 are 10. Hence, even after we add the noise, we can always tell whether the true value was r_1 or r_2. Note that the same example holds for every $b = 2^n$ and every pair of true values r_1 and r_2 which differ by $(2^n k+h)/2^d$ where k is any integer and h is any integer between 1 and $2^n - 1$. Figure 1 illustrates the situation for $b = 4$, $k = 3$ and $h = 2$.

Another attack, based on the IEEE standard floating-point representation [14], was presented in [17]. In contrast to [17], we have chosen an example based on the fixed point representation because it allows to illustrate more distinctively a problem for privacy which rises from the finite precision[2] and which

[1] The name "oblivious" comes from the fact that the final answer depends only on the answer to the query and not on the dataset.

[2] More precisely, the problem is caused by scaling a finite set of randomly generated numbers. It is easy to prove that the problem raises for any implementation of numbers, although it may not raise *for every point* like in the case of the fixed-point representation.

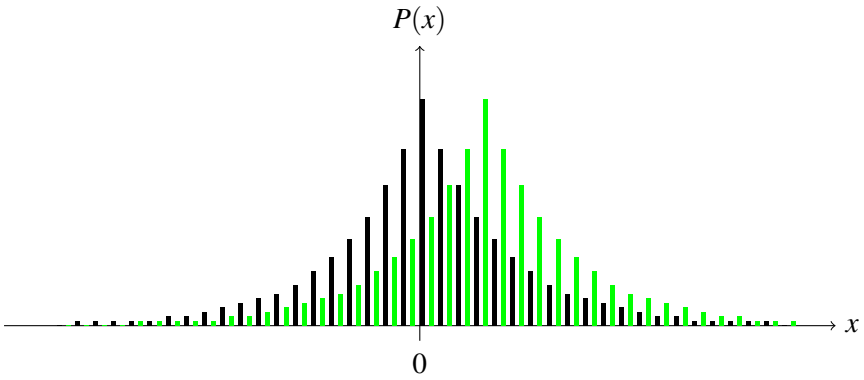

Figure 1: The probability distribution of the reported answers after the addition of Laplacian noise for the true answer $r_1 = 0$ (black) and $r_2 = 3 \cdot 2^{-4} + 2^{-5}$ (green).

is, therefore, pandemic. (This is not the case for the example in [17]: fixed-point and integer-valued algorithms are immune to that attack.)

In this paper, we propose a solution to fix the privacy breach induced by the finite-precision implementation of a differentially-private mechanism for any kind of implementation. Our main concern is to establish a bound on the degradation of privacy induced by both the finite representation and by the computational errors in the generation of the noise. In order to achieve this goal, we use the concept of *closeness* introduced by the authors in [12], which allows us to reason about the approximation errors and their accumulation. We make as few assumptions as possible about the procedure for generating the noise. In particular, we do not assume that the noise has a linear Laplacian distribution: it can be any noise that provides differential privacy and whose implementation satisfies a few properties (normally granted by the implementation of real numbers) which ensure its closeness. We illustrate our method with two examples: the classic case of the univariate (i.e., linear) Laplacian, and the case of the bivariate Laplacian. The latter distribution is used, for instance, to generate noise in privacy-aware geolocation mechanisms [2].

1.1 Related work

As far as we know, the only other work that has considered the problem introduced by the finite precision in the implementation of differential privacy is [17]. As already mentioned, that paper showed an attack on the Laplacian-based implementation of differential privacy within the IEEE standard floating-point representation[3]. To thwart such an attack, the author of [17] proposed a method that avoids using the standard uniform random generator for floating point (because it does not draw all representable numbers but only multiple of 2^{-53}). Instead, his method generates two integers, one for the mantissa and one for the exponent in such a way that every representable number is drawn with its correct probability. Then it computes the linear Laplacian using a logarithm implementation (assumed to be full-precision), and finally it uses a snapping mechanism consisting in truncating large values and then rounding the final result.

The novelties of our paper, w.r.t. [17], consist in the fact that we deal with a general kind of noise, not necessarily the linear Laplacian, and with any kind of implementation of real numbers, not necessarily

[3]We discovered our attack independently, but [17] was published first.

the IEEE floating point standard. Furthermore, our kind of analysis allows us to measure how safe an existing solution can be and what to do if the requirements needed for the safety of this solution are not met. Finally, we consider our correct implementation of the bivariate Laplacian also as a valuable contribution, given its practical usefulness for location-based applications.

The only other work we are aware of, considering both computational error and differential privacy, is [7]. However, that paper does not consider at all the problem of the loss of privacy due to implementation error: rather, they develop a technique to establish a bound on the error, and show that this technique can also be used to compute the sensitivity of a query, which is a parameter of the Laplacian noise.

1.2 Plan of the paper

This paper is organized as follow. In section 2, we recall some mathematical definitions and introduce some notation. In section 3, we describe the standard Laplacian-based mechanism that provides differential privacy in a theoretical setting. In section 4, we discuss the errors due to the implementation, and we consider a set of assumptions which, if granted, allows us to establish a bound on the irregularities of the noise caused by the finite-precision implementation. Furthermore we propose a correction to the mechanism based on rounding and truncating the result. Section 5 contains our main theorem, stating that with our correction the implementation of the mechanism still preserves differential privacy, and establishing the precise degradation of the privacy parameter. The next two sections propose some applications of our result: Section 6 illustrates the technique for the case of Laplacian noise in one dimension and section 7 shows how our theorem applies to the case of the Euclidean bivariate Laplacian. Section 8 concludes and discusses some future work.

2 Preliminaries and notation

In this section, we recall some basic mathematical definitions and we introduce some notation that will be useful in the rest of the paper. We will assume that the the queries give answers in $\mathbb{R}^m$. Examples of such queries are the tuples representing, for instance, the average height, weight, and age. Another example comes from geolocation, where the domain is $\mathbb{R}^2$.

2.1 Distances and geometrical notations

There are several natural definitions of distance on $\mathbb{R}^m$ [19]. For $m \in \mathbb{N}$ and $x = (x_1, \ldots, x_m) \in \mathbb{R}^m$, the L_p norm of x, which we will denote by $\|x\|_p$, is defined as

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{m} |x_i|^p}$$

The corresponding distance function is

$$d_p(x,y) = \|x-y\|_p$$

We extend this norm and distance to $p = \infty$ in the usual way: $\|x\|_\infty = \max_{i\in\{1,\ldots,m\}} |x_i|$ and $d_\infty(x,y) = \|x-y\|_\infty$. The notion of L_∞ norm is extended to functions in the following way: given $f : A \to \mathbb{R}^m$, we define $\|f\|_\infty = \max_{x\in A} \|f(x)\|$. When clear from the context, we will omit the parameter p and write simply $\|x\|$ and $d(x,y)$ for $\|x\|_p$ and $d_p(x,y)$, respectively.

Let $S \subseteq \mathbb{R}^m$. We denote by S^c the *complement of* S, i.e., $S^c = \mathbb{R}^m \setminus S$. The *diameter of* S is defined as

$$\varnothing(S) = \max_{x,y \in S} d(x,y).$$

For $\varepsilon \in \mathbb{R}^+$, the $+\varepsilon$-*neighbor* and the $-\varepsilon$-*neighbor* of S are defined as

$$S^{+\varepsilon} = \{x \mid \exists s \in S, d(x,s) \leq \varepsilon\} \qquad S^{-\varepsilon} = \{x \mid \forall s \in \mathbb{R}^m, d(x,s) \leq \varepsilon \implies s \notin S\} = ((S^c)^{+\varepsilon})^c$$

For $x \in \mathbb{R}^m$, the *translations* of S by x and $-x$ are defined as

$$S + x = \{y + x \mid y \in S\} \qquad S - x = \{y - x \mid y \in S\}$$

2.2 Measure theory

We recall here some basic notions of measures theory that will be used in this paper.

Definition 2.1 (σ-algebra and measurable space)**.** *A σ-algebra $\mathscr{T}$ for a set M is a nonempty set of subsets of M that is closed under complementation (wrt to M) and (potentially empty) enumerable union. The tuple $(M, \mathscr{T})$ is called a* measurable space*.*

Definition 2.2 (Measure)**.** *A* positive measure *μ on a measurable space $(M, \mathscr{T})$ is a function $\mathscr{T} \to \mathbb{R}^+ \cup \{0\}$ such that $\mu(\emptyset) = 0$ and whenever (S_i) is a enumerable family of disjoint subset of M then*

$$\sum \mu(S_i) = \mu(\bigcup S_i).$$

A positive measure μ where $\mu(X) = 1$ is called a probability measure*.*

A tuple $(M, \mathscr{T}, P)$ where $(M, \mathscr{T})$ is a measurable space and P a probability measure is called *probability space*.

In this paper we will make use of the Lebesgue measure λ on $(\mathbb{R}^m, \mathscr{S})$ where $\mathscr{S}$ is the Lebesgue σ-algebra. The Lebesgue measure is the standard way of assigning a measure to subsets of $\mathbb{R}^m$.

Definition 2.3 (Measurable function)**.** *Let $(M, \mathscr{T})$ and (V, Σ) be two measurable spaces. A function $f : M \to V$ is measurable if $f^{-1}(v) \in \mathscr{T}$ for all $v \in \Sigma$.*

Definition 2.4 (Absolutely continuous)**.** *A measure ν is absolutely continuous according to a measure μ, if for all $M \in \mathscr{S}$, $\mu(M) = 0$ implies $\nu(M) = 0$.*

If a measure is absolutely continuous according to the Lebesgue measure then by the Radon-Nikodym theorem, we can express it as an integration of a density function f:

$$\mu(M) = \int_M f(x)\, d\lambda$$

2.3 Probability theory

Definition 2.5 (Random variable)**.** *Let $(\Omega, \mathscr{F}, P)$ be a probability space and $(E, \mathscr{E})$ a measurable space. Then a random variable is a measurable function $X : \Omega \to E$. We shall use the expression $P[X \in B]$ to denote $P\left(X^{-1}(B)\right)$.*

Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be a measurable function and let $X : \Omega \to \mathbb{R}^m$ be a random variable. In this paper, we will use the notation $f(X)$ to denote the random variable $Y : \Omega \to \mathbb{R}^m$ such that $f(X)(\omega) = f(X(\omega))$. In particular, for $m \in \mathbb{R}^m$ we denote by $m + X$ the random variable $Y : \Omega \to \mathbb{R}^m$ such that $\omega \mapsto X(\omega) + m$.

Definition 2.6 (Density function)**.** *Let $X : \Omega \to E$ be a random variable. If there exists a function f such that, for all $S \in \mathscr{S}$, $P[X \in S] = \int_S f(u)\,du$, then f is called the* density function *of X.*

In this paper, we use the following general definition of the Laplace distribution (centered at zero).

Definition 2.7 (Laplace distribution)**.** *The density function F of a Laplace distribution with scale parameter b is $F_b(x) = K(b)e^{-b\|x\|}$ where $K(b)$ is a normalization factor which is determined by imposing $\int_S F_b(x)\,dx = 1$.*

Definition 2.8 (Joint probability)**.** *Let (X,Y) be a pair of random variable on $\mathbb{R}^m$. The joint probability on (X,Y) is defined for all $I,J \in \mathscr{S}$ as: $P[(X,Y) \in (I,J)] = P[X \in I \wedge Y \in J]$.*

Definition 2.9 (Marginals)**.** *Let X and $Y : (\Omega, \mathscr{F}, P) \to (\mathbb{R}^m, \mathscr{S})$. The marginal probability of the random variable (X,Y) for X is defined as:*

$$P[X \in B] = \int_{\mathbb{R}^m} P[(X,Y) \in (B, \mathrm{d}y)].$$

3 Differential privacy in the exact semantics

In this section we recall the definition of differential privacy and of the standard mechanisms to achieve it, and we discuss its correctness.

3.1 Differential privacy

We denote by $\mathscr{D}$ the set of databases and we assume that the domain of the answers of the query is $\mathbb{R}^m$ for some $n \geq 1$. We denote by $D_1 \sim D_2$ the fact that D_1 and D_2 differ by at most one element. Namely, D_2 is obtained from D_1 by adding or removing one element.

Definition 3.1 (ε-differential privacy)**.** *A randomized mechanism $\mathscr{A} : \mathscr{D} \to \mathbb{R}^m$ is ε-differentially private if for all databases D_1 and D_2 in $\mathscr{D}$ with $D_1 \sim D_2$, and all $S \in \mathscr{S}$ (the Lebegue σ-algebra), we have :*

$$P[\mathscr{A}(D_1) \in S] \leq e^{\varepsilon} P[\mathscr{A}(D_2) \in S]$$

Definition 3.2 (sensitivity)**.** *The sensitivity Δ_f of a function $f : \mathscr{D} \to \mathbb{R}^m$ is*

$$\Delta_f = \sup_{D_1, D_2 \in \mathscr{D}, D_1 \sim D_2} d(f(D_1), f(D_2)).$$

3.2 Standard technique to implement differential privacy

The standard technique to grant differential privacy is to add random noise to the true answer to the query. In the following, we denote the query by $f : \mathscr{D} \to \mathbb{R}^m$. This is usually a deterministic function. We represent the noise as a random variable $X : \Omega \to \mathbb{R}^m$. The standard mechanism, which we will denote by $\mathscr{A}_0$, returns a probabilistic value which is the sum of the true result and of a random variable X, namely:

Mechanism 1.

$$\mathscr{A}_0(D) = f(D) + X$$

3.3 Error due to the implementation of the query

The correctness of a mechanism $\mathscr{A}$, if we do not take the implementation error into account, consists in $\mathscr{A}$ being ε-differentially private. However, we are interested in analyzing the correctness of the implemented mechanism. We start here by discussing the case in which, in mechanism 1, the noise X is exact but we take into account the approximation error in the implementation of f.

Notation Given a function g, we will indicate by g' its implementation, i.e. the function that, for any x, gives as result the value actually computed for $g(x)$, with all the approximation and representation errors.

The first thing we observe is that the implementation of f can give a sensitivity $\Delta_{f'}$ greater than Δ_f and we need to take that into account. In fact, in the exact semantics the correctness of the mechanism relies on the fact that $d(f(D_1), f(D_2)) \leq \Delta_f$. However, with rounding errors, we may have $d(f'(D_1), f'(D_2)) > \Delta_f$. Hence we need to require the following property, usually stronger than differential privacy.

Condition 1. *Given a mechanism $\mathscr{A}(D) = f'(D) + X$, we say that $\mathscr{A}$ satisfies Condition 1 with degree ε (the desired degree of differential privacy) if the random variable X has a probability distribution which is absolutely continuous according to the Lebesgue measure, and*

$$\forall S \in \mathscr{S}, r_1, r_2 \in \mathbb{R}^m, P[r_1 + X \in S] \leq e^{\varepsilon \frac{d(r_1, r_2)}{\Delta_{f'}}} P[r_2 + X \in S]$$

Remark 1. *In general we expect that an analysis of the implementation of f will provide some bound on the difference between f and f', and that will allow us to provide a bound on $\Delta_{f'}$ in terms of Δ_f. For instance, if $\|f - f'\| \leq \delta_f$ then we get $\Delta_{f'} \leq \Delta_f + 2\delta_f$.*

Proposition 3.1. *Condition 1 implies that the mechanism $\mathscr{A}(D) = f'(D) + X$ is ε-differentially private (w.r.t. f').*

Proof Let D_1 and D_2 be two databases such that $D_1 \sim D_2$. Let $r_1 = f'(D_1)$ and $r_2 = f'(D_2)$ be two answers. By definition of sensitivity, $d(r_1, r_2) \leq \Delta_{f'}$ so $e^{\varepsilon \frac{d(r_1, r_2)}{\Delta_{f'}}} \leq e^{\varepsilon}$. Hence,

$$P[\mathscr{A}(D_1) \in S] \leq e^{\varepsilon} P[\mathscr{A}(D_2) \in S]$$

□

The following theorem shows that Condition 1 is actually equivalent to differential privacy in the case of Laplacian noise.

Theorem 3.1. *Let $\mathscr{A}(D) = f'(D) + X$ be a mechanism, and assume that X is Laplacian. If $\mathscr{A}$ is ε-differentially private (w.r.t. f'), then Condition 1 holds.*

Proof First, we show that if $\mathscr{A}$ is ε-differentially private then $b \leq \frac{\varepsilon}{\Delta_{f'}}$ holds for the scale parameter b of X. Let $D_1 \sim D_2$ with $d(f'(D_1), f'(D_2)) = \Delta_{f'}$. By ε-differential privacy we have, for any $S \in \mathscr{S}$:

$$P[f'(D_1) + X \in S] \leq e^{\varepsilon} P[f'(D_2) + X \in S]$$

From the density function of the Laplace noise (Definition 2.7), we derive:

$$K(n,d) d\lambda \leq e^{\varepsilon} K(n,d) e^{-b\Delta_{f'}} d\lambda$$

Hence,

$$b \leq \frac{\varepsilon}{\Delta_{f'}}. \tag{1}$$

Now, by definition of the density function, we have

$$P[r_2 + X \in S] = \int_{x \in S} K(n,d) e^{-bd(x,r_2)} d\lambda$$

From the triangular inequality, we derive:

$$P[r_2 + X \in S] \geq \int_{x \in S} K(n,d) e^{-b(d(x,r_1)+d(r_1,r_2))} d\lambda$$

Hence,

$$P[r_2 + X \in S] \geq e^{-bd(r_2,r_1)} \int_{x \in S} e^{-bd(r_1,x)} d\lambda$$

From inequality (1), we derive:

$$P[r_2 + X \in S] \geq e^{-\frac{\varepsilon d(r_2,r_1)}{\Delta_{f'}}} \int_{x \in S} e^{-bd(r_1,x)} d\lambda$$

Finally,

$$P[r_2 + X \in S] \geq e^{-\frac{\varepsilon d(r_2,r_1)}{\Delta_{f'}}} P[r_1 + X \in S]$$

□

4 Error due to the implementation of the noise

In this section we consider the implementation error in the noise, trying to make as few assumptions as possible about the implementation of real numbers and of the noise function.

We start with example which shows that any finite implementation makes it impossible for a mechanism to achieve the degree of privacy predicted by the theory (i.e. the degree of privacy it has in the exact semantics). This example is more general than the one in the introduction in the sense that it does not rely on any particular implementation of the real numbers, just on the (obvious) assumption that in a physical machine the representation of numbers in memory is necessarily finite. On the other hand it is less "dramatic" than the one in the introduction, because it only shows that the theoretical degree of privacy degrades in the implementation, while the example in the introduction shows a case in which ε-differential privacy does not hold (in the implementation) for any ε.

Example 4.1. Consider the standard way to produce a random variable with a given probability law, such as the Laplace distribution. Randomness on most computers is generated with integers. When we call a function that returns a uniform random value on the representation of reals, the function generates a random integer z (with uniform law) between 0 and N (in practice $N \geq 2^{32}$) and returns $u = z/N$. From this uniform random generator, we compute $n(z/N)$ where n depends on the probability distribution we want to generate. For instance, to generate the Laplace distribution we have $n(u) = -b\,\mathrm{sgn}(u - 1/2)\ln(1 - 2|u - 1/2|)$ which is the inverse of the cumulative function of the Laplace distribution. However the computation of n is performed in the finite precision semantics, i.e. n is a function $\mathbb{F} \to \mathbb{F}$ where $\mathbb{F}$ is the finite set of the representable numbers. In this setting, the probability of getting some value x for our noise depends on the number of integers z such that $n(z/N) = x$: if there are k values for z such

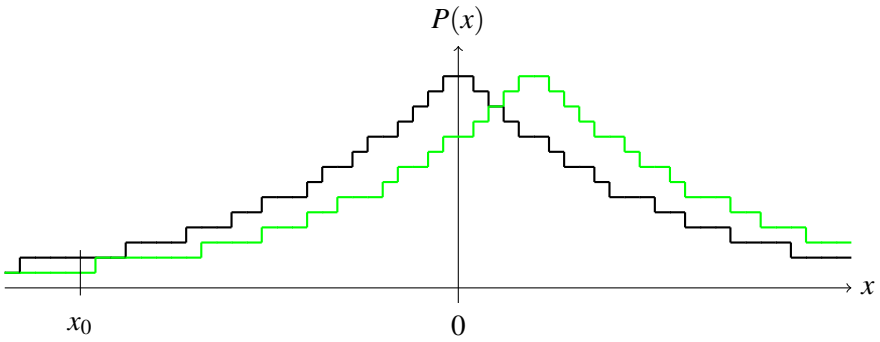

Figure 2: The probability distributions of Laplace noises generated from a discretized uniform generator

that $n(z/N) = x$ then the probability of getting x is k/N. This means, in particular, that, if the theoretical probability for a value x is $1.5/N$, then the closest probability actually associated with the drawing of x is either $1/N$ or $2/N$ and in both cases the error is at least 33%. In figure 2, we illustrate how the error on the distribution breaks the differential-privacy ratio that holds for the theoretical distribution. The ratio between the two theoretical Laplacian distributions is $4/3$. However, since the actual distribution is issued from a discretization of the uniform generator, the resulting distribution is a step function. So when the theoretical probability is very low like in x_0, the discretization creates an artificial ratio of 2 instead of $4/3$.

4.1 The initial uniform random generator

To generate a random variable, programing languages have only one primitive that generates a random value between 0 and 1 that aims to be uniform and independent across several calls.Hence, to get a random variable with a non uniform distribution, we generate it with a function that makes calls to this random generator. For instance, to draw a value from a random variable X on $\mathbb{R}$ distributed according to the cumulative function $C : \mathbb{R} \to]0,1]$, it is sufficient to pick a value u from the uniform generator in $]0,1]$ and then return $C^{-1}(u)$.

We identify three reasons why a uniform random generator may induce errors. The first has been explained in the introduction: finite precision allows generating only N different numbers such that when we apply a function on the value picked some values are missing and other are over represented. The second reason comes from the generator itself which can returns the N values with different probabilities even though we might assume that they are returned with probability $1/N$: furthermore, some values may not even be returned at all. A third error is due to the dependence of returned results when we pick several random values. Indeed most of the generator implementations are indeed pseudo generators: when a value is picked the next one is generated as a hash function of the first one. This means that if we have N possibilities for one choice then we also have N possible pairs of successive random values.

To reason about implementation leakage, we have to take into account all of these sources of errors. We propose the following model. In the exact semantics, the uniform random variable U^q is generated from a cross product of q uniform independent variables U (with $q \geq m$). We denote by $u_1, \ldots, u_q \in [0,1]^q$ the values picked by our perfect random generator. Then we consider the random variable $U^{q'}$ actually provided as generated from a function $n_0 : \mathbb{R}^q \to \mathbb{R}^q$, $(u'_1, \ldots, u'_q) = n_0(u_1, \ldots, u_q)$. We assume that the

bias, i.e., the difference between n_0 and the identity, is bounded by some $\delta_0 \in \mathbb{R}^+$:

$$\|n_0 - \mathrm{Id}\|_\infty \leq \delta_0. \tag{2}$$

4.2 The function n for generating the noise

From the value u (resp. u') drawn according to the distribution U^q in the exact semantics (resp. according to $U^{q'}$ in the actual implementation), we generate another value by applying the functions n and n' respectively. Let $X = n(U^q)$ be the random variable with the exact distribution and $X' = n'(U^{q'})$ the random variable with the actual one.

Definition 4.1. *We denote by μ and ν the probability measure of X and X', respectively: for all $S \in \mathscr{S}$, $\mu(S) = P[n(U^q) \in S] = \lambda(n^{-1}(S))$ and $\nu(S) = P[n'(U^{q'}) \in S] = \lambda(n_0^{-1}(n^{-1}(S)))$.*

In order to establish a bound on the difference between the probability distribution of X and X' we need some condition on the implementation n' of n. For this purpose we use the notion of closeness that we defined in [12].

Definition 4.2 ((k,δ)-close, [12])**.** *Let A and B be metric spaces with distance d_A and d_B, respectively. Let n and n' be two functions from A to B and let $k, \delta \in \mathbb{R}^+$. We say that n' is (k,δ)-close to n if*

$$\forall u, v \in A, d_B(n(u), n'(v)) \leq k\, d_A(u,v) + \delta.$$

This condition is a combination of the k-Lipschitz property, that states a bound between the error on the output and the error on the input, see below, and the implementation errors of n:

Definition 4.3 (k-Lipschitz)**.** *Let (A, d_A) and (B, d_B) be two metrics spaces and $k \in \mathbb{R}$: A function $n : A \to B$ is k-Lipschitz if:*

$$\forall u, v \in A, d_B(n(u), n(v)) \leq k d_A(u,v)$$

In [12], we have proven the following relation between the properties of being k-Lipschitz and of being (k,δ)-close.

Theorem 4.1 ([12])**.** *If n is k-Lipschitz and $\|n - n'\|_\infty \leq \delta$ then n and n' are (k,δ)-close.*

We strengthen now the relation by proving that (a sort of) the converse is also true.

Theorem 4.2. *If there exist u, v, $d(n(u), n(v)) > kd(u,v) + 2\delta$ then there exist no function n' such that n and n' are (k,δ)-close.*

Proof Let u, v such that $d(n(u), n(v)) > kd(u,v) + 2\delta$. Let n' such that n and n' are (k,δ)-close. From the definition of closeness, we get $d(n(u), n'(u)) \leq \delta$ and $d(n'(u), n(v)) \leq kd(u,v) + \delta$. From a triangular inequality, we derive $d(n(u), n(v)) \leq kd(u,v) + 2\delta$. Hence, we obtain a contradiction. □

Now, we would like n and n' to be (k,δ)-close on $\mathbb{R}^m$. However, this implies that $\mathscr{A}_0$ (mechanism 1) cannot be ε-differentially-private. In fact, the latter would imply $\|n([0,1]^q)\|_\infty = \infty$ otherwise certain answers could be reported (with non-null probability) only in correspondence with certain true answers and not with others. However, $\|n([0,1]^q)\|_\infty = \infty$ and $[0,1]^q$ bounded implies there exist u, v, such that $d(n(u), n(v)) > kd(u,v) + 2\delta$: we derive from theorem 4.2 that n' cannot exist.

In order to keep computed results in a range where we are able to bound the computational errors, one possible solution consists of a truncation of the result. The traditional truncation works as follows: choose a subset $\mathbb{M}_r \subset \mathbb{R}^m$ and, whenever the reported answer x is outside $\mathbb{M}_r$ return the closest point to x in $\mathbb{M}_r$. However, while such a procedure is safe in the exact semantics because remapping does not alter differential privacy, problems might appears when n and n' are not close. Furthermore, while in the

uni-dimensional case there are two disjoints set that are mapped one on the minimal value and the other on the maximal value, in higher dimensions we have a connected set that is mapped on several points, and on which the error is not bounded.

Therefore, to remain in a general framework where we do not have any additional knowledge about computational errors for large numbers, we decide here to return an exception value when the result is outside of some compact subset $\mathbb{M}_r$ of $\mathbb{R}^m$. We denote by ∞ the value returned by the mechanism when $f'(D)+X' \notin \mathbb{M}_r$. Hence, the truncated mechanism $\mathscr{A}$ returns the randomized value or ∞:

Mechanism 2.

$$\mathscr{A}(D) = \begin{cases} f'(D)+X' & \text{if } f'(D)+X' \in \mathbb{M}_r \\ \infty & \text{otherwise} \end{cases}$$

We truncate the result because we want to exclude non-robust computations from our mechanism. However, such a procedure is effective only if unsafe computations remain outside the safe domain. To grant this property we need two more conditions. One requires the implementation to respect the monotonicity of the computed functions:

Condition 2. *We say that a function $g : \mathbb{R}^m \to \mathbb{R}^k$ satisfies Condition 2 if, for all $x, y \in \mathbb{R}^m$, $\|g(x)\| \leq \|g(y)\|$ implies $\|g'(x)\| \leq \|g'(y)\|$.*

With this property, even if the implementation is not robust for large values, if we know some result is not in $\mathbb{M}_r$ then the result for any greater value is not in $\mathbb{M}_r$ either.

The other condition is about the closeness of the implementation of the noise and its exact semantics in a safe area. For any $\delta_r \in \mathbb{R}^+$, we consider the set $U_r \subset U^q$ defined as $\forall u \in U_r \ \|n(u)\| \leq \varnothing(\mathbb{M}_r) + \delta_r$ i.e. : $U_r = n^{-1}(\{y \,|\, \|y\| \leq \varnothing(\mathbb{M}_r) + \delta_r\})$.

Condition 3. *We say that a noise n satisfies Condition 3 if n and n′ are (k, δ_n)-close on a set U_r such that*

$$\forall u \in U_r^c \ f'(D) + n(u) \notin \mathbb{M}_r^{+k\delta_0+\delta_n}$$

To find such a set U_r, one possible way is by a fix point construction. We begin by finding the smallest k_0 and δ_{n0} such that n and n' are (k_0, δ_{n0})-close on $\mathbb{M}_r$. Then for the generic step $m > 0$, we compute the smallest k_{m+1} and δ_{nm+1} such that n and n' are (k_{m+1}, δ_{nm+1})-close on $\mathbb{M}_r^{k_m\delta_0+\delta_{nm}}$.

If Conditions 2 and 3 hold, then from (2) we derive

$$\forall u \in U_r^c \ f'(D) + n'(n_0(u)) \notin \mathbb{M}_r \tag{3}$$

So whatever happens outside of U_r, the result will be truncated. We can then consider that there is no implementation error outside U_r. Finally, we have a bound δ_t for the maximal shift between the exact and the actual semantics:

$$\delta_t = k\delta_0 + \delta_n \tag{4}$$

4.3 A distance between distributions

Given that we are in a probabilistic setting, the round-off errors cannot be measured in terms of numerical difference as they can be in the deterministic case, they should rather be measured in terms of distance between the theoretical distribution and the actual distribution. Hence, we need a notion of distance between distributions. We choose to use the ∞-Wassertein distance [5] which, as we will show, is the natural metric to measure our deviation.

Definition 4.4 (∞-Wassertein distance)**.** *Let μ, ν two probability measures on $(\mathbb{R}^m, \mathscr{S})$ such that there exist a compact Ω, $\mu(\Omega) = \nu(\Omega) = 1$, the ∞-Wassertein distance between μ and ν is defined as follows:*

$$W_\infty(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \left(\inf_{t\geq 0} \left(\gamma\left(\left\{ (x,y) \in (\mathbb{R}^m)^2 \middle| d(x,y) > t \right\} \right) \right) = 0 \right)$$

Where $\Gamma(\mu,\nu)$ denotes the collection of all measure on $M \times M$ with marginals μ and ν respectively.

If we denote by $\mathrm{Supp}(x,y)$, the support where $\gamma(x,y)$ is non zero, we have an equivalent definition [5] for the ∞-Wassertein distance:

$$W_\infty(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \left(\sup_{\mathrm{Supp}(x,y)} d(x,y) \right)$$

We extend this definition to any pair of measures that differ only on a compact ($\mathbb{M}_r$ in our case) by considering the subset of $\Gamma(\mu,\nu)$ containing only measure $\gamma(x,y)$ with $\gamma(x,y) = 0$ if $x \neq y$ and either $x \in \mathbb{M}_r^c$ or $y \in \mathbb{M}_r^c$.

We have introduced this measure because it has a direct link with the computational error as expressed by the following theorem.

Theorem 4.3. *Let X and X' be two random variables with distribution μ and ν respectively. We have that $\|X - X'\|_\infty \leq \delta$ implies $d(\mu,\nu) \leq \delta$.*

Proof We consider the measure γ on $M \times M$, $\forall A, B \in \mathscr{S}, \gamma(A,B) = P(X \in A \wedge X' \in B)$. The marginals of γ are μ and ν. Moreover, the support of γ is δ since $P(X \in A \wedge X' \in B) = 0$ when A and B are distant by more than δ. Since we have such a γ the minimum on all the $\gamma \in \Gamma(\mu,\nu)$ is less than δ. □

In our case, according to (4), we have $d(\mu,\nu) \leq \delta_t$. The following theorem allow us to bound the μ measure of some set with the measure ν.

Theorem 4.4.

$$d(\mu,\nu) \leq \varepsilon \implies \forall S \in \mathbb{R}^m, \nu(S^{-\varepsilon}) \leq \mu(S) \leq \nu(S^\varepsilon)$$

Proof The property of marginals is $\nu(S) = \int_{\mathbb{R}^m \times S} d\gamma(x,y)$. Since $\gamma(x,y) = 0$ if $d(x,y) > \varepsilon$, we derive $\nu(S) = \int_{S^\varepsilon \times S} d\gamma(x,y)$. Then we get $\nu(S) \leq \int_{S^\varepsilon \times \mathbb{R}^m} d\gamma(x,y)$. The last expression is the marginal of γ in S^ε, hence by definition of marginal: $\nu(S) \leq \mu(S^\varepsilon)$. The other inequality is obtain by considering the complement set of S ($\mathbb{R}^m \setminus S$). □

4.4 Rounding the answer

Once the computation of $\mathscr{A}(D)$ is achieved, we cannot yet return the answer, because it could still leak some information. Indeed, the distribution of X and X' are globally the same, but, on a very small scale, the distributions could differ a lot. We prevent this problem by rounding the result:

Mechanism 3. *The mechanism rounds the result by returning the value closest to $f(D) + n'$ in some discrete subset S'. So $\mathscr{K}(D) = r(\mathscr{A}(D))$ where r is the rounding function.*

From the above rounding function we define the set $\mathscr{S}'_0$ of all sets that have the same image under r. Then we define the σ-algebra $\mathscr{S}'$ generated by $\mathscr{S}'_0$: it is the closure under union of all these sets. Observe now that it is not possible for the user to measure the probability that the answer belongs to a set which is not in $\mathscr{S}'$. Hence our differential privacy property becomes:

$$\forall S \in \mathscr{S}', P[\mathscr{A}(D_1) \in S] \leq e^\varepsilon P[\mathscr{A}(D_2) \in S] \tag{5}$$

In this way we grant that any measurable set has a minimal measure and we prevent the inequality from being violated when probabilities are small. The following value R represents the robustness of the rounding.

$$R = \max_{S \in \mathscr{S}'_0, S \neq \emptyset} \frac{\lambda(S^{\delta_t} \setminus S^{-\delta_t})}{\lambda(S^{-\delta_t})} \tag{6}$$

5 Preserving differential privacy

In this section, we prove that if all conditions are met, then the implementation of the mechanism satisfies differential privacy.

Theorem 5.1. *Any mechanism that respects Conditions 1–3 is ε'-differentially private, with:*

$$\forall S \in \mathscr{S}, P[\mathscr{A}'(D_1) \in S] \leq e^{\varepsilon'} P[\mathscr{A}'(D_2) \in S']$$

where $\varepsilon' = \varepsilon + \ln(1 + Re^{\varepsilon \frac{L+\delta_t}{\Delta_{f'}}})$, $\delta_t = k\delta_0 + \delta_n$ and $L = \max_{S \in \mathscr{S}'_0} \varnothing S$.

Proof Let S in $\mathscr{S}$. We first consider the case $S \neq \infty$.

Define $P_1 = P[\mathscr{A}'(D_1) \in S]$ and $P_2 = P[\mathscr{A}'(D_2) \in S]$. Since the result has been rounded (Definition 3), it is equivalent to consider the set $S' \in \mathscr{S}'$ with $S' = r^{-1}(S)$ instead of S.

Now we have $P_i = P[f'(D_i) + n'(X) \in S'] = P[n'(X) \in S' - f'(D_i)]$ where i is 1 or 2. Since ν is the measure associated to n', we have

$$P_i = \nu(S' - f'(D_i))$$

From (4) and Theorem 4.3, $d(\nu, \mu) \leq \delta_t$. From Theorem 4.4 we derive

$$P_1 \leq \mu(S^{\delta_t} - f'(D_1)) \qquad \text{and} \qquad P_2 \geq \mu(S^{-\delta_t} - f'(D_2)).$$

The additivity property of measures grants us $\mu(S^{\delta_t}) = \mu(S^{-\delta_t}) + \mu(S^{\delta_t} - S^{-\delta_t})$. Condition 1 can be expressed in term of the measure as:

$$\forall S \in \mathscr{S}, r \in \mathbb{R}^m \|r\|, \mu(S) \leq e^{\varepsilon \frac{\|r\|}{\Delta_{f'}}} \mu(S - r)$$

From this inequality, we can derive, since $\|r\| = \Delta_{f'}$:

$$\mu(S^{\varepsilon}) \leq e^{\varepsilon} P_2 + \mu(S^{\delta_t} \setminus S^{-\delta_t})$$

Since the probability is absolutely continuous according to the Lebesgue measure (Condition 1), we can express the probability with a density function p:

$$\forall S \in \mathscr{S}, \mu(S) = \int_S p(x) d\lambda$$

We derive:

$$\forall S \in \mathscr{S}, \min_{x \in S} p(x) \leq \frac{\mu(S)}{\lambda(S)}$$

By applying this property on $S^{-\delta_t} - f'(D_2)$, we get:

$$\min_{x \in S^{-\delta_t} - f'(D_2)} p(x) \leq \frac{\mu(S^{-\delta_t} - f'(D_2))}{\lambda(S^{-\delta_t} - f'(D_2))}$$

We derive:

$$\exists x_0 \in S - f'(D_2), p(x_0) \leq \frac{P_2}{\lambda(S)}$$

By the triangular inequality, we can bound the distance between x_0 and any point of S^{δ_t} by $\Delta_{f'} + L + \delta_t$ Hence, from Condition 1 we derive:

$$\forall x \in S^{\delta_t} - f'(D_1), p(x) \leq e^{\varepsilon \frac{\Delta_{f'}+L+\delta_t}{\Delta_{f'}}} p(x_0)$$

Then by integration:

$$\mu(S^{\delta_t} - f'(D_1) \setminus S^{-\delta_t}) \leq e^{\varepsilon \frac{\Delta_{f'}+L+\delta_t}{\Delta_{f'}}} \frac{\lambda(S^{\delta_t} \setminus S^{-\delta_t})}{\lambda(S^{-\delta_t})} P_2$$

We apply the condition 6:

$$\mu(S^{\delta_t} - f'(D_1) \setminus S^{-\delta_t}) \leq e^{\varepsilon \frac{\Delta_{f'}+L+\delta_t}{\Delta_{f'}}} RP_2$$

Finally we obtain :

$$P_1 \leq (1 + Re^{\varepsilon \frac{L+\delta_t}{\Delta_{f'}}}) e^{\varepsilon} P_2$$

In case S is ∞, due to (3), $P[\mathscr{A}'(D) = \infty]$ is the same as $P[f'(D) + X' \in \mathbb{M}_r^c]$ where $d(\mu, \nu) \leq \delta_t$. Moreover, $\mathbb{M}_r^c$ can be decomposed in a enumerable disjoint union of element of $\mathscr{S}_0$. Therefore, the first part of the proof applies: ε'-differential privacy holds for all these elements. By the additivity of the measure of disjoint union we conclude.

□

6 Application to the Laplacian noise in one dimension

In this section we illustrate how to use our result in the case in which the domain of the answers is $\mathbb{R}$. The noise added for the protocol, stated in the mechanism 1, is the Laplacian centered in 0 with scale parameter $\Delta_{f'}/\varepsilon$. Theorem 3.1 implies that Condition 1 holds for ε. We truncate the result outside of some interval $\mathbb{M}_r = [m, M]$.

Implementation of the n function To generate a centered Laplacian distribution from a uniform random variable U in $]0,1]$, a standard method consists in using the inverse of the cumulative function, i.e. $X = n(U) = -b\,\mathrm{sgn}(U - 1/2)\ln(1 - 2|U - 1/2|)$, where b is the intended scale parameter ($\frac{\Delta_{f'}}{\varepsilon}$ in our case). Hence our exact function n is

$$n(u) = \frac{\Delta_{f'}}{\varepsilon} \mathrm{sgn}(u - 1/2)\ln(1 - 2|u - 1/2|). \tag{7}$$

Closeness of n and n' In order to apply our theorem, we need to prove that Condition 3 is satisfied. By theorem 4.1, it is sufficient to prove that, in the interval of interest, $n(u)$ is k-Lipschitz and that $|n(u) - n'(u)| \leq \delta_n$. Note that the values of δ_n and k in general depend on n and on its implementation (often the logarithm is implemented by the CORDIC algorithm).

The logarithm function used by n is not k-Lipschitz for any k. However, we are interested in the behavior of n when $|n(u)| \leq M - m$. From the definition of n in (7), we have:

$$\frac{dn}{du}(u) \leq dn_{\max} = \frac{2\Delta_{f'}}{\varepsilon} e^{\frac{\varepsilon \varnothing(\mathbb{M}_r)}{\Delta_{f'}}}$$

in $U_r = \{u | n(u) \leq M - m\}$. So our function n is $dn_{\max}$-Lipschitz. Finally, our global error is

$$\delta_t = \frac{2\Delta_{f'}}{\varepsilon} e^{\frac{\varepsilon \varnothing(\mathbb{M}_r)}{\Delta_{f'}}} \delta_0 + \delta_n$$

Rounding the result The rounding process generates a σ-algebra $\mathscr{S}'$ composed by small intervals of length L where L is the accuracy step of the rounding. In that case, the value defined in (6) is $R = \frac{L+2\delta_t}{L-2\delta_t}$.

Differential privacy By Theorem 5.1, the implementation of our mechanism is ε'-differentially private with

$$\varepsilon' = \varepsilon + \ln(1 + \frac{L+2\delta_t}{L-2\delta_t} e^{\varepsilon \frac{L+\delta_t}{\Delta_{f'}}})$$

Remark 2. *In case our answer is not in $[m, M]$, we can return $-\infty$ or $+\infty$ instead of ∞. The reason is that even if the algorithm is not robust when $|u - 0.5|$ is small the sign is still correct. Then we can remap $-\infty$ to m and $+\infty$ to M to get the usual truncation procedure.*

7 Application to the Laplacian noise in $\mathbb{R}^2$

When the domain of the answers are the points of a map, like in the case of location-based applications, it is natural to formalize it as the space $\mathbb{R}^2$ equipped with the Euclidean distance.

According to the protocol, we sanitize the results by adding a random variable X. In this case, we will use for X the bivariate Laplacian defined for the Euclidean metric [2] whose density function is:

$$p(x, y) = Ke^{b\sqrt{|x-x_0|^2 + |y-y_0|^2}}$$

where K is the normalization constant and b the scale parameter. Since we are using a Laplacian noise, by Theorem 3.1, Condition 1 holds.

Truncation Since most of the time the domain studied is bound (for instance the public transportation of a city is inside the limit of the city), we can do a truncation. However, we recall that our truncation is made for robustness purpose and not just for utility reasons. Hence, if our domain of interest is a circle, we will not choose $\mathbb{M}_r$ to be the same circle because the probability the truncation would return an exception would be too high (more than one half if the true result is on the circumference).

Implementation of the n function Following [2], we compute the random variable by drawing an angle and a distance independently. The angle θ is uniformly distributed in $[-\pi, \pi[$. The radius r has a probability density $D_{\varepsilon,R}(r) = \varepsilon^2 r e^{-\varepsilon r}$ and cumulative function $C_\varepsilon(r) = 1 - (1 + \varepsilon r)e^{-\varepsilon r}$. The radius can therefore be drawn by setting $r = C_\varepsilon^{-1}(u)$ where u is generated uniformly in $]0, 1]$.

Robustness of n As in the previous section, we do not analyze an actual implementation but we care about the k factor used for Condition 3. First, we analyze for which $k_C(\varepsilon, \varnothing(\mathbb{M}_r))$ the function C_ε^{-1} is k-Lipschitz in $[0, \varnothing(\mathbb{M}_r)]$. Since C is differential, this question is equivalent to find the inverse of the minimal value taken by its derivative function on the interval $C_\varepsilon^{-1}([0, \varnothing(\mathbb{M}_r)])$. By computing this minimum value, we get:

$$K_C(\varepsilon, \varnothing(\mathbb{M}_r)) = \frac{e^{\varepsilon \varnothing(\mathbb{M}_r)}}{2\varepsilon + r\varepsilon^2}$$

On the other hand, the computation of θ is just a multiplication by 2π of the uniform generator hence $k_\theta = 2\pi$. Then, with the conversion $(r, \theta) \mapsto (r\cos(\theta), r\sin(\theta))$ from polar coordinates to Cartesian coordinates we obtain the global k factor:

$$k = \sqrt{K_C(\varepsilon, \varnothing(\mathbb{M}_r))^2 + 2\pi\varnothing(\mathbb{M}_r)}$$

Let δ_n be the distance between n and n', and δ_0 be the error of the uniform generator. From (4) we get:

$$\delta_t = \sqrt{K_C(\varepsilon, \varnothing(\mathbb{M}_r))^2 + 2\pi\varnothing(\mathbb{M}_r)}\delta_0 + \delta_n.$$

Rounding the answer We now compute the parameter R in (6). The rounding is made in the Cartesian coordinates, hence the inverse image of any returned value is a square S of length L. Note that S^{δ_t} is included in the square of length $L + 2\delta_t$ and $S^{-\delta_t}$ is a square of length $L - 2\delta_t$. Hence the ratio value is smaller than $R = (\frac{L+2\delta_t}{L-2\delta_t})^2$.

Differential privacy By Theorem 5.1 we get that (the implementation of) our mechanism is ε'-differentially private with

$$\varepsilon' = \varepsilon + \ln(1 + (\frac{L+2\delta_t}{L-2\delta_t})^2 e^{\varepsilon \frac{L+\delta_t}{\Delta_{f'}}})$$

8 Conclusion and future work

In this paper we have shown that, in any implementation of mechanisms for differential privacy, the finite precision representation of numbers in any machine induces approximation errors that cause the loss of the privacy property. To solve this problem, we have proposed a method based on rounding the answer and raising an exception when the result is outside some values. The main result of our paper is that the above method is sound in the sense that it preserves differential privacy at the price of a degradation of the privacy degree. To prove this result, we needed to pay special attention at expressing the problem in terms of probability theory and at defining the link between computational error and distance between probability distributions. Finally, we have shown how to apply our method to the case of the linear Laplacian and to that of bivariate Laplacian.

As future developments of this work, we envisage two main lines of research:

- Deepening the study of the implementation error in differential privacy: there are several directions that seem interesting to pursue, including:
  - Improving the mechanisms for generating basic random variables. For instance, when generating a one-dimensional random variable, it may have some advantage to pick more values from the uniform random generator, instead than just one (we recall that the standard method

is to draw one uniformly distributed value in $]0,1]$ and then apply the inverse of the cumulative function). For instance, $u_1 + u_2$ has a density function with a triangular shape and cost only one addition. The other advantage is due to the finite representation: if the uniform random generator can pick N different values then two calls of it generate N^2 possibilities, which enlarge considerably the number of possibilities, and therefore reduce the "holes" in the distribution.

- Considering more relaxed versions of differential privacy, for instance the (ε, δ)-differential privacy allows for a (small) additive shift δ between the two likelihoods in Definition 3.1 and it is therefore more tolerant to the implementation error. It would be worth investigating for what values of δ (if any) the standard implementation of differential privacy is safe.

- Enlarging the scope of this study to the more general area of quantitative information flow. There are various notions of information leakage that have been considered in the computer security literature; the one considered in differential privacy is just one particular case. Without the pretense of being exhaustive, we mention the information-theoretic approaches based on Shannon entropy [8, 16, 6] and those based on Rényi min-entropy [20, 4] and the more recent approach based on decision theory [1]. The main difference between differential privacy and these other notions of leakage is that in the former any violation of the bound in the likelihood ratio is considered catastrophic, while the latter focuses on the average amount of leakage, and it is therefore less sensitive to the individual violations. However, even though the problem of the implementation error may be attenuated in general by the averaging, we expect that there are cases in which it may still represent a serious problem.

References

[1] Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *Proc. of CSF*, pages 265–279, 2012.

[2] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. Technical report, 2012. Av. at `arXiv:1212.1984`.

[3] Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Béguelin. Probabilistic relational reasoning for differential privacy. In *Proc. of POPL*. ACM, 2012.

[4] Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proc. of MFPS*, volume 249 of *ENTCS*, pages 75–91. Elsevier, 2009.

[5] T. Champion, L. De Pascale, and P. Juutinen. The ∞-wasserstein distance: Local solutions and existence of optimal transport maps. *SIAM*, 40(1):1–20, 2008.

[6] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Inf. and Comp.*, 206(2–4):378–401, 2008.

[7] Swarat Chaudhuri, Sumit Gulwani, Roberto Lublinerman, and Sara NavidPour. Proving programs robust. In *Proc. of ESEC-13*, pages 102–112. ACM, 2011.

[8] David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantified interference for a while language. In *Proc. of QAPL*, volume 112 of *ENTCS*, pages 149–166. Elsevier, 2005.

[9] Cynthia Dwork. Differential privacy. In *Proc. of ICALP*, volume 4052 of *LNCS*, pages 1–12. Springer, 2006.

[10] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 265–284. Springer, 2006.

[11] Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Ravi Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *Proc. of POPL 2013.*, 2013.

[12] Ivan Gazeau, Dale Miller, and Catuscia Palamidessi. A non-local method for robustness analysis of floating point programs. Technical report, INRIA, June 2012. Available at `http://hal.inria.fr/hal-00665995`.

[13] Shen-Shyang Ho and Shuhua Ruan. Differential privacy for location pattern mining. In *Proc. of SPRINGL*, pages 17–24. ACM, 2011.

[14] IEEE Task P754. *IEEE 754-2008, Standard for Floating-Point Arithmetic*. IEEE, pub-IEEE-STD:adr, August 2008.

[15] Ashwin Machanavajjhala, Daniel Kifer, John M. Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *Proc. of ICDE*, pages 277–286. IEEE, 2008.

[16] Pasquale Malacaria. Assessing security threats of looping constructs. In *Proc. of POPL*, pages 225–235. ACM, 2007.

[17] Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proc. of*, CCS '12, pages 650–661, New York, NY, USA, 2012. ACM.

[18] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proc. of S&P*, pages 173–187. IEEE, 2009.

[19] Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, 3rd edition, 1986.

[20] Geoffrey Smith. On the foundations of quantitative information flow. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 288–302. Springer, 2009.