Project no.:	**PIRSES-GA-2011-295261**
Project full title:	**Mobility between Europe and Argentina applying Logics to Systems**
Project Acronym:	**MEALS**
Deliverable no.:	**3.5 / 2**
Title of Deliverable:	**Preserving differential privacy under finite-precision semantics**

Contractual Date of Delivery to the CEC:	**30 September 2015**
Actual Date of Delivery to the CEC:	**20 September 2015**
Organisation name of lead contractor for this deliverable:	**INR**
Author(s):	**Ivan Gazeau, Dale Miller Catuscia Palamidessi**
Participants(s):	**INR**
Work package contributing to the deliverable:	**WP3**
Nature:	**R**
Dissemination Level:	**Public**
Total number of pages:	**29**
Start date of project:	1 Oct. 2011 **Duration:** 48 month

Abstract:

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 adding noise to the values correlated to the private data. The typical mechanisms used to achieve differential privacy have been proved correct in the ideal case in which computations are made using infinite-precision semantics. In this paper, we analyze the situation at the implementation level, where the semantics is necessarily limited by 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, acceptable) variant of the property, under only a minor degradation of the privacy level.

This project has received funding from the European Union Seventh Framework Programme (FP7 2007-2013) under Grant Agreement Nr. 295261.

Contents

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 and privacy applications, the approximation error can became a fingerprint potentially causing the disclosure of secret or confidential information.

We investigate here the confidentiality issues caused by the use of finite precision. More specifically, we study the information leaked about the input data that can be caused by the errors in the result. The standard techniques to measure security breaches do not apply because those techniques analyze the *ideal* system—i.e. using *exact* semantics—and do 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.

While the example above is elementary, it illustrates the pervasive nature of this problem and the impact it can have on confidentiality. Clearly, the problem of working without full precision should receive adequate treatment.

In this paper, we consider a more tricky case where the agent returns a noisy answer by using random numbers. Often the addition of random noise is done on purpose to prevent access to secret values. We will see that, however, since noise is generated in finite precision, even noise contains computational errors that allow an attacker to retrieve secrets.

In order to have concrete examples and to analyze existing specifications, we will study here the particular case of using finite precision with *differential privacy*. Differential privacy is an approach to the protection of private information in the field of statistical databases. Statistical databases are databases containing individual records, aiming at supporting the discovery of aggregate information, such as plausible causes for diseases, social norms, or trends in public opinion. Of course, statistical databases have to preserve the anonymity and privacy of participants: it is likely that people will not participate in a survey if they know that their personal information will be revealed to anybody. However, as we explain in section 2.3, granting anonymity and privacy is not a trivial task.

Differential privacy was first proposed in [7, 9] as a formal approach to preserve the anonymity and privacy of the participants in a statistical database. This approach is now being used in many other domains ranging from programming languages [2, 10] to social networks [16] and geolocation [14, 12, 1].

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 of 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. Consider the simplest representation of reals: fixed-point numbers. This representation is used on low-cost processors which typically do not have a 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 $p(x) = {}^1\!/_{2b} \cdot \exp(-{}^{|x-r|}\!/_b)$ 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. The sensitivity of the query is the maximal difference in the result when one entry is removed or added. 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 $p_X(x) = {}^1\!/_2 \exp(-|x|)$. Hence, if we want to generate the random variable bX with probability distribution

$$p_{bX}(x) = {}^1\!/_{2b} \cdot \exp(-{}^{|x|}\!/_b),$$

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 $b = 2^m$ for some fixed integer m (b can be big when the sensitivity of the query and the required privacy degree i.e. ${}^1\!/_\epsilon$ are high). In this case, the multiplication by 2^m returns a number $2^m z \cdot 2^{-d}$ that, in the fixed-point representation, terminates with m zeroes. Hence, when we add this noise to the true result, we return a value whose representation has the same m 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, it is still possible to determine which was the true value between 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 $2^m - 1$. Figure 1 illustrates the situation for $m = 2$, $b = 4$, $d = 6$, $k = 3$ and $h = 2$. End of Example 1.

Another attack, based on the IEEE standard floating-point representation [13], was presented in [15]. In contrast to [15], we have chosen an example based on the fixed point representation

[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.

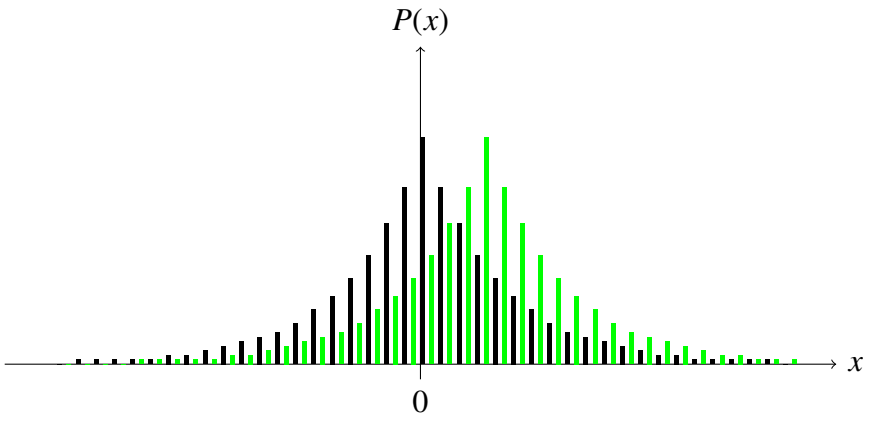

Figure 1: The probability distribution of the reported answers after the addition of Laplacian noise $P_{bX}(x)$ with $b = 4$, for the true answer $r_1 = 0$ (black) and $r_2 = 3 \cdot 2^{-4} + 2^{-5}$ (green).

because it illustrates more distinctively a problem for privacy which arises from the finite precision[2] and which is, therefore, pandemic. This is not the case for the example in [15]: 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* which allows us to reason about the approximation errors and their accumulation. In addition, 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 [1].

In [8], the more flexible notion of (ϵ, δ)-differential privacy was introduced, in order to relax the constraint on the ratio in case of small probabilities. The idea is that two negligible likelihoods should be considered equivalent, but ϵ-differential privacy is totally violated when one of them is, for example, 0. The δ is a constant that is added to the bound on the ratio, and in this way, we can ignore small divergences between small likelihoods, even when they have a significant impact on the ratio.

We will consider the effects of finite precision in the correctness of the implementation with respect to (ϵ, δ)-differential privacy and we will see that we need less strict conditions to ensure this property than to ensure ϵ-differential privacy. On the other hand, while (ϵ, δ)-differential

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

privacy is considered a sufficient privacy guarantee in many cases, it does not have the pleasant mathematical properties of ϵ-differential privacy, notably concerning compositionality. For this reason we consider important to study also how to achieve ϵ-differential privacy.

1.1 Related work

Mironov has discovered independently the problem introduced by the finite precision in the implementation of differential privacy [15]. 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 [15] proposed a method that avoids using the standard uniform random generator for floating-point (because it does not draw all representable numbers but only multiples of 2^{-52}). 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. In [15], the mechanism is given in a specific context: the numbers representation is the IEEE floating-point standard, the noise is the Laplacian noise in one dimension, the implementation is full-precision. In this paper, on the contrary, we aim at providing strong foundations to analyze the loss of differential privacy due to the (generic) finite-precision implementation. Hence, even if we propose similar techniques (truncation and rounding) to avoid the loss to be severe, we provide some general criteria to bound the loss on any domain $\mathbb{R}^m$, on any noise function and on any representation scheme for numbers. This allows, for instance, to use our results for geolocation, where the domain is the Euclidean plan. Finally, the generality of our method allows us to extend our result to the (ϵ, δ)-differential privacy case.

Another work that considers both computational error and differential privacy is [4]. 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.

There are also tools that compute directly an estimation of leakage by statistical methods [5, 6]. Such approaches are however limited by design in the number of bits of the inputs and of the pseudo random-noise. Moreover, they can only provide confidence intervals, which are enough to find likely attacks, but provide only weak guarantees of safety, especially for differential privacy, which is tuned on the worst case scenario.

1.2 Plan of the paper

This paper is organized as follows. In section 2, we recall some mathematical definitions as well as the basis of differential privacy and we introduce some notation. In section 3, we discuss how finite precision causes errors in the result of the algorithm. We modify the initial algorithm

[3] We discovered our attack independently, but [15] was published first.

in Section 3.5 to deal with some of these errors. In section 4, we use the bounds on the error found in the previous section to get a bound on the increase of the ϵ parameter due to the finite implementation. In Section 5, we compute some order of magnitude in a basic case to provide a better insight on how to choose the different parameters occuring in the modified mechanism. Section 6 concludes.

2 Preliminaries and notation

In this section, we recall some definitions and we introduce some notation that we use throughout the paper.

2.1 Geometrical notations

There are several natural definitions of distance on $\mathbb{R}^m$ [17]. 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 L_∞ norm concept is extended to functions in the following way: given $f : A \to \mathbb{R}^m$, we define $\|f\|_{p,\infty} = \max_{x\in A} \|f(x)\|_p$. When clear from the context, we will omit the parameter p and write simply $\|x\|$, $d(x, y)$ and $\|f\|_\infty$ for $\|x\|_p$, $d_p(x, y)$ and $\|f\|_{p,\infty}$ respectively.

Let $S \subseteq \mathbb{R}^m$. The *diameter of* S is defined as $\varnothing(S) = \max_{x,y\in S} d(x, y)$. For $x \in \mathbb{R}^m$, the *translations* of S by x and $-x$ are defined as $S + x = \{y + x \mid y \in S\}$ and $S - x = \{y - x \mid y \in S\}$ where $+$ and $-$ are the vectorial spaces operators.

Definition 1. We define the expansion of the set S by $r \in \mathbb{R}^+$: $\text{expand}(S, +r) = \{x \mid \exists s \in S, d(x, s) \leq r\}$ and the shrinking of the set S by $r \in \mathbb{R}^+$: $\text{expand}(S, -r) = \{x \mid \forall s \in \mathbb{R}^m, d(x, s) \leq r \implies s \in S\}$.

Note that these two definitions are related as follows: $\mathbb{R}^m \setminus \text{expand}(S, -\epsilon) = \text{expand}(\mathbb{R}^m \setminus S, +\epsilon)$.

2.2 Probability and measure theory

In this section, we recall the basic notions of measure theory.

Definition 2 (σ-algebra and measurable space)**.** A σ-algebra $\mathcal{T}$ for a set M is a nonempty set of subsets of M that is closed under complementation (w.r.t. to M) and (potentially empty) enumerable union. The tuple $(M, \mathcal{T})$ is called a *measurable space*.

On $\mathbb{R}^m$, we denote by $\mathcal{S}$ the Lebesgue's σ-algebra that contains all hypercubes.

Definition 3 (Probability space)**.** A probability space is a tuple $(\Omega, \mathcal{F}, P)$ where Ω is a sample space (i.e., the set of all possible outcomes), $\mathcal{F}$ is a set of events (where each event is a set of zero or more outcomes) which is also a σ-algebra on Ω, and P is a probability measure on $(\Omega, \mathcal{F})$ (i.e., a measure such that $P(\Omega) = 1$).

Definition 4 (Random variable)**.** Let $(\Omega, \mathcal{F}, P)$ be a probability space and $(E, \mathcal{E})$ be 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 $P\left(X^{-1}(B)\right)$.

Notice that given a random variable $X : \Omega \to E$ and a measurable function $f : E \to F$, $f \circ X : \Omega \to F$ is also a random variable.

Definition 5 (Support)**.** The support of a measure is the set of all x where the measure of some neighborhood of x is not null. By extension, the support of a random variable X is the support of its corresponding measure.

We will make use of the Lebesgue measure λ on $(\mathbb{R}^m, \mathcal{S})$ where $\mathcal{S}$ is the Lebesgue σ-algebra. This is the standard way of assigning a measure to subsets of $\mathbb{R}^m$.

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

Definition 6 (Laplace distribution)**.** The density function F of a Laplace distribution with scale parameter b is $F_b(x) = K(b)\exp(-b\|x\|)$ where $K(b)$ is a normalization factor which is determined by imposing $\int_S F_b(x)\,dx = 1$.

2.3 Differential privacy

In this section, we recall the definition of differential privacy on databases. We will denote by $\mathcal{D}$ the set of all possible databases. We start by formalizing the notion of presence versus non-presence of an individual.

Definition 7 (adjacent databases)**.** Given two databases D_1 and D_2 in $\mathcal{D}$, we denote by $D_1 \sim D_2$ the fact that D_1 and D_2 differ by exactly one row. Namely, D_2 is obtained from D_1 by adding or removing the data of one individual.

In order to protect the private data of an individual, the differential privacy framework uses randomization to obfuscate the answers to queries.

Definition 8 (randomized mechanism)**.** A randomized mechanism $\mathcal{K}$ is a function that takes a database D in $\mathcal{D}$ and a query q and returns a random variable X: $X = \mathcal{K}_q(D)$. We omit the q parameter when there is no ambiguity.

We can now provide the formal definition of differential privacy. The idea is that the probability of obtaining a certain answer to the query is almost the same whether an individual is present or not, hence the answer does not provide too much probabilistic information about the data of the individual.

Definition 9 (ϵ-differential privacy)**.** A randomized mechanism $\mathcal{K} : \mathcal{D} \to \mathbb{R}^m$ is ϵ-differentially private if for all databases D_1 and D_2 in $\mathcal{D}$ with $D_1 \sim D_2$, and all $S \in \mathcal{S}$ (the Lebesgue σ-algebra on $\mathbb{R}^m$), we have :

$$P[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) P[\mathcal{K}(D_2) \in S]$$

Sometime, a weaker property is used instead of differential ϵ-privacy, namely (ϵ, δ)-differential privacy [8], whose definition is the following.

Definition 10 ((ϵ, δ)-differential privacy)**.** A randomized mechanism $\mathcal{K} : \mathcal{D} \to \mathbb{R}^m$ is ϵ-differentially private if for all databases D_1 and D_2 in $\mathcal{D}$ with $D_1 \sim D_2$, and all $S \in \mathcal{S}$ (the Lebesgue σ-algebra on $\mathbb{R}^m$), we have :

$$P[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) P[\mathcal{K}(D_2) \in S] + \delta$$

Intuitively, this definition means that the ϵ-differential privacy may fail with probability δ. As we will detail, this weaker notion is interesting since it is much easier to obtain it, even for very small δ for the implemented mechanism. In the following, we show that if we are content with the more relaxed (ϵ, δ)-differential privacy framework, then the problem caused by the finite-precision implementation can be addressed in a easier way.

2.4 Standard technique to implement differential privacy

We will focus on the so-called oblivious mechanisms, which are the most used, and are defined as follows.

Definition 11 (oblivious mechanisms)**.** A mechanism is oblivious if it is obtained by applying a random function to the true answer to the query, i.e., not directly to the database. We use the notation $f_q(D)$ to denote the true answer on the database D and $f(D)$ when there is no ambiguity about q.

To calibrate the noise of an oblivious mechanism we do not need to consider the databases anymore, the only interesting parameter is the maximal deviation that one individual can generate for a given query. This leads to the following definition.

Definition 12 (sensitivity)**.** The sensitivity Δ_f of a function $f : \mathcal{D} \to \mathbb{R}^m$ is

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

We are interested in the particular case of oblivious mechanisms that only add a random value to the result. These are the most common oblivious mechanisms and we will limit our study to them.

Mechanism 1.

$$\mathcal{K}(D) = f(D) + X \tag{1}$$

Another important feature of a mechanism is its *utility*. For instance, a function that only returns a random variable independently of the input is 0-differentially private but since it never allows learning anything, as a mechanism it would be perfectly useless. In the case of an additive mechanism the utility can be defined as a function of the variance between the true answer and the real value.

A mechanism that offers a good trade-off between privacy and utility is the so-called Laplacian mechanism, which is defined as in Mechanism 1, where X is a Laplacian calibrated by ϵ and the sensitivity of the query. Namely, the density function of X is $F_b(x)$ (cf. Definition 6) with $b = {}^{\Delta_{f_q}}/_{\epsilon}$.

3 Measuring the computational error

To limit the leakage due to finite precision, we need first to ensure that the finite semantics is not too far from the exact one. In this section, we formalize this in Proposition 2 and we explain what assumptions we need on the finite-precision representation in order to satisfy it. For that, we first analyze the different sources of errors. In Section 3.1, we argue why only the added noise X' need to be studied. In Section 3.2, we briefly explain that standard techniques to generate pseudo-random values of a given distribution use first a uniform pseudo-random generator U' and then apply to the drawn value a function n'. In section 3.3, we study the influence of finite precision on U' and in Section 3.4, we study the influence of n'. From this study, we obtain an impossibility result (Theorem 1) which says that Proposition 2 cannot be achieved with Mechanism 1. Finally, we consider two options. The first one, which consists of modifying the mechanism, is described in Section 3.5. The other one, which consists of relaxing the ϵ-differential privacy to (ϵ, δ)-differential privacy is presented in Section 3.6.

Proposition 2 is only a first step to be able to bound the leakage. In Section 4, we will consider other weaknesses of a direct implementation of the mechanism, and provide our complete fix of Mechanism 1 to achieve differential privacy in the finite semantics.

3.1 Approximate computation of the true answer to the query and of the addition

Let f be the function computing the true answer of the query, and f' its implementation in finite precision.

The difference between f' and f may affect the utility of the answer, but in general it does not influence the level of differential privacy. In particular, in the case of an oblivious mechanism, the way the deviation on f influences the level of privacy is only via the features of f that are used for the computation of the noise. Typically such a feature is the sensitivity of the query, Δ_f. For example, the definition of Laplacian noise depends on the parameter b which is Δ_f.

Finally, the implementation of the addition function $(x +' y)$ occuring in $f(D) + X$ can differ from the exact addition $(x + y)$. However, since it is a simple function all actual implementations are full-precision: the returned result is just a rounding of the exact result when the exact result is not a representable number. Formally, $x +' y = rnd(x + y)$ for some rounding function rnd. We explain in Section 4 that rounding on the final result preserve differential privacy. For these reasons, we can assume that the implementation of + is exact.

3.2 Implementation of real valued random variables

Computer systems usually provide a primitive that implements a random variable U of range $[0, 1[$ with uniform distribution.[4] Other random variables and their density functions can usually be defined as functions of this U. For instance $X = n \circ U = 4U - 2$ is a uniform random variable of

[4] In general, a machine provides a pseudo-random generator U, not a "real" random one. However the question of the reliability of pseudo-randomness is out of the scope of this paper, so we will assume that U is really random.

range $[-2, 2[$. For random variables whose support (co-domain) is in $\mathbb{R}^m$, the definition can make use of a tuple U of several independent uniform random variables $U_1, \ldots, U_m$. For instance, $(X, Y) = f(U_1, U_2) = (U_2 \cos(2\pi U_1), U_2 \sin(2\pi U_1))$ is a random variable whose support is the unit disc.

In this paper, we assume that the implementation of the noise in Mechanism 1 is based on U and on a given function n. Hence, we only consider mechanisms in which the random function X of Mechanism 1 is of the form $n \circ U$:

Mechanism 2.

$$\mathcal{K}_n(D) = f(D) + n \circ U \tag{2}$$

In the following we consider the errors introduced by the emulations of those U and n in finite precision.

3.3 Error due to the initial random generator

A perfect uniform random generator would generate any value in $[0, 1[$ with uniform distribution. However, since the implementation can only return values with a finite precision, the distribution is not uniform. Here we study how this deviation influences the error in the mechanism.

We denote by U the tuple of m uniform random variables used to generate the noise in the exact semantics. In other words, U is a uniform random variable in $[0, 1[^m$. We denote by U' the pseudo-random variable in $[0, 1[^m$ generated in the finite precision semantics. To model the error, we consider that $U' = n_0 \circ U$ for some measurable function n_0 of type $[0, 1[^m \to [0, 1[^m$ that maps any $u \in [0, 1[^m$ to the finite set of possible outputs of our pseudo-random generator. In summary, U and U' are random variables of type $\Omega \to [0, 1[^m$, and $U'(\omega) = n_0(U(\omega))$.

For instance, if our precision allows representing only numbers that are multiples of 2^{-53}, then we can model U' with the function n_0 that takes a real number and truncates it after the 53^{rd} decimal bit.

We will use ξ_0 to denote the error due to this initial random generator:

Definition 13. The initial error $\xi_0 \in \mathbb{R}^+$ is the maximal distance between $n_0(u)$ and u for any $u \in [0, 1[^m$:

$$\xi_0 = \|n_0 - \mathrm{Id}\|_\infty.$$

where Id is the identity function.

For instance, in case U' generates numbers that are multiples of ξ, we have $\xi_0 = \xi$. In our previous example we have $\xi = 2^{53}$.

3.4 Errors due to the transformation function

We now consider the error in the implementation of the function n which is used to transform the initial uniform U into a noise $n \circ U$ with the desired distribution. We denote by n' the actual function resulting by implementing n in finite-precision and by $X' = n' \circ U' = n' \circ n_0 \circ U$ the random variable actually generated.

In order to bound the maximal error between the result generated by the implementation and the exact one, we use the notion of closeness that was defined in [11].

Definition 14 ((k,ξ)-closeness, [11])**.** 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 $k, \xi \in \mathbb{R}^+$. We say that n' is (k,ξ)-close to n if

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

The closeness property is related to the one of being k-Lipschitz, as shown in [11]:

Proposition 1 ([11])**.** *If n is k-Lipschitz and $\|n - n'\|_\infty \leq \xi$ then n and n' are (k,ξ)-close.*

Corollary 1. *n_0 and Id are $(1,\xi_0)$-close.*

Proof From Definition 13, the fact Id is 1-Lipschitz, and Proposition 1. □

In the following, our goal is to find suitable conditions to achieve the property that the implementation is (k,ξ)-close to the exact mechanism:

Proposition 2. *Let $n_{impl.}$ be the function corresponding to the implementation of n, there exists some $\xi_t \in \mathbb{R}^+$ such that $\forall u \in [0,1[^m, \quad d(f(D) + n(u), f(D) + n_{impl.}(u)) \leq \xi_t$.*

If, in addition to the closeness of n_0 and Id, n and n' were also (k,ξ)-close for some k and ξ, then, by compositionality we could derive that n and $n' \circ n_0$ are $(k, k\xi_0+\xi)$-close as well, and which would prove Proposition 2 for $\xi_t = k\xi_0+\xi$. Unfortunately, we prove in the following that whatever is the implementation n', n and n' cannot be (k,ξ)-close. Our solutions then consists of reasoning in an *idealized* implementation which can return numbers not finitely representable. In the first solution which modifies the mecanism and we provide conditions that ensure that the difference between the *idealized* implementation and the actual implementation are not observable. In the second solution which relax differential privacy, we bound the probability that the *idealized* implementation differs from the actual implementation.

To show the impossibility result, we start with the following lemma which states that in a differentially private mechanism, if the noise is additive, then its amplitude is not bounded.

Lemma 1. *If a mechanism of the form of Mechanism 1 is differentially private, and there exist at least two adjacent databases where the query has two different answers, then the support of X has infinite diameter.*

Proof Let r_1 and r_2 be different answers on two adjacent databases. Let $M \in \mathcal{S}$ (the set of measurable sets) such that $\|M\|$ is bounded and $P(X \in M) > 0$. By definition of translations (cf. Section 2.1), we also have $P(r_1 + X \in M + r_1) > 0$. Since $\mathcal{K}$ is ϵ-differentially private, we must have $P(r_2 + X \in M + r_1) > 0$ as well, otherwise the ratio between the two probabilities would be infinite. Hence we have shown that $P(X \in M) > 0 \implies P(X \in M + r_1 - r_2) > 0$. Since this is valid for any M, we have in particular that for any $h \in \mathbb{N}$, $P(X \in M + h(r_1 - r_2)) > 0 \implies P(X \in M + (h+1)(r_1 - r_2)) > 0$. From the assumption $P(X \in M) > 0$ and the last implication, we conclude, by induction, that for all $h \in \mathbb{N}$, $P(X \in M + h(r_1 - r_2)) > 0$.

Finally, since M is bounded, for any $r \in \mathbb{R}$ there exists a h such that

$$\{x \in \mathbb{R}^m \mid d(0,x) \leq r\} \cap M + h(r_1 - r_2) = \emptyset \tag{3}$$

where 0 is the origin. (Note that $\{x \in [0,1[^m | \ d(0,x) \leq r\}$ is the ball centered in 0 of radius r.) This means that the set $M + h(r_1 - r_2)$ does not have any element x with $d(0,x) \leq r$. In summary, we have exhibited sets $(M + h(r_1 - r_2))$ with non null probability which are arbitrarily far from the origin. We can conclude that, for all $s \in \mathbb{R}$, $P(X > s) > 0$. □

Finally, we show that indeed the deviation caused by the finite precision, in the implementation of the noise, is unbounded:

Theorem 1. *If a mechanism of the form of Mechanism 2 is differentially private, then there is no* $\xi \in \mathbb{R}$ *such that*

$$\forall u \in [0,1[^m,\ d((n' \circ n_0)(u), n(u)) \leq \xi$$

Proof Since there is only a finite set of possible values $n_0(u)$ for $u \in [0,1[^m$, there exists a maximum value $x_M = \max\{\|n'(n_0(u))\| \mid u \in [0,1[^m\}$. From Lemma 1, we have that, for every $\xi \in \mathbb{R}$, there exists $u \in [0,1[^m$ such that $\|n(u)\| > \xi + x_M$, and therefore $d(n(u), n'(n_0(u))) > k$. □

In Figure 2, we illustrate the problem pointed out by Theorem 1, and the consequent impossibility for the implementation to achieve differential privacy on the whole domain of a mechanism of the form of Mechanism 2. We have considered a noise n defined so that the ideal mechanism would be 0.2-differentially private, and we have simulated the distribution of the corresponding noise generated in fixed precision (where errors have been overplayed). The black and the green curves represent the right branch of the distributions $f(D_1) + n' \circ n_0 \circ U$ and $f(D_2) + n' \circ n_0 \circ U$ respectively, where $f(D_1) = 0$ and $f(D_2) = 1$. They are step functions: each step represents the probability for a given representable value. As we can see, while the ratio between the two distributions is (almost) $\exp(0.25)$ around 0, when the mechanism returns a larger noise the ratio becomes much higher, and for the value x_M defined in the proof of Theorem 1 we have $P(f(D_1) + n' \circ n_0 \circ U > x_M) = 0$ while $P(f(D_2) + n' \circ n_0 \circ U > x_M) \neq 0$, which means that the ratio is infinite.

To get around this impossibility result, we have two options. The first one consists of truncating the (final) result: because high values are truncated, the observable behavior of the implementation can remain safe. The other one consists of considering the more relaxed notion of (ϵ, δ)-differential privacy property: because the problem appears in with small probability and it might be acceptable to tolerate differential privacy breaches with small probability. The quantitative analysis of these two strategy are strongly related, in the two next sections, we present what is specific to each of them to get Proposition 2 satisfied, then Section 4 concerns both options except for the final results.

3.5 Truncating the result

In this section we discuss how to modify the mechanism so as to be able to bound the difference between n and $n' \circ n_0$. One way of doing this is by truncating the result.

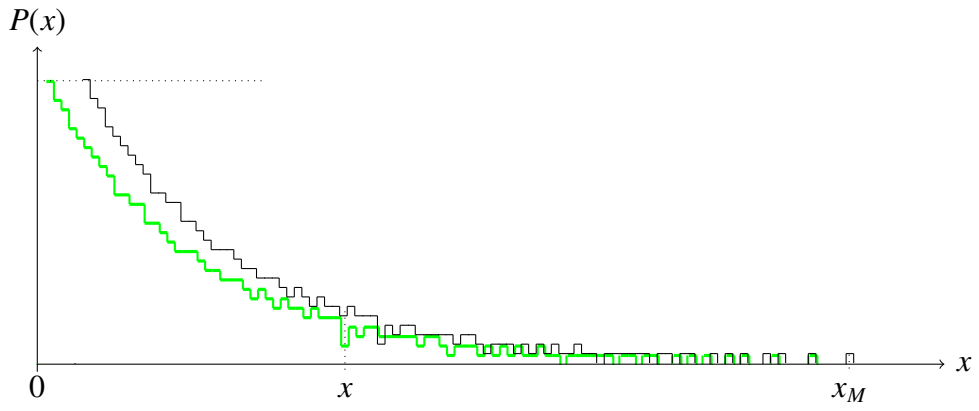

Figure 2: The effect of the finiteness of $n_0 \circ U$ on the implementation of the noise

Some differentially private mechanisms already use a truncation procedure. The idea is the following: choose a subset $\mathbb{M}_r \subset \mathbb{R}^m$ (domain of interest) and, whenever the reported answer x is outside $\mathbb{M}_r$, return the closest point to x in $\mathbb{M}_r$. In the exact semantics the truncation is safe, in the sense that it has been proved that it does not alter differential privacy. However, the fact that n and n' are not close means that we cannot expect the properties of n to be transferred to n'. Indeed, there are cases in which this method is not safe in the finite-precision semantics, unless we make strong assumptions on the computational errors. Since we are trying to be as general as possible, we cannot use this kind of truncation.[5] Hence, we adopt the following simple approach: Whenever the result is outside the intended subset $\mathbb{M}_r$ of $\mathbb{R}^m$, we return an exception.

This truncation method is quite drastic in the sense that all values outside $\mathbb{M}_r$ are collapsed in the same value (the "exception"). The advantage is that under relatively mild assumptions we can prove that this method is safe in all cases. On the other hand, the loss of utility is higher than in the standard method mentioned above. Better mechanisms can be found, but they are specific to some algorithms or to some dimensions. For instance, in the uni-dimensional case ($m = 1$) it is possible to return the lowest or the highest value instead of an undifferentiated value for both cases.

We denote by ∞ the exception, i.e., the value returned by the mechanism when $f(D)+n\circ U \notin \mathbb{M}_r$. Hence, the truncated mechanism $\mathcal{K}_{\mathbb{M}_r}$ is defined as:

Mechanism 3.

$$\mathcal{K}_{\mathbb{M}_r}(D) = \begin{cases} \mathcal{K}_n(D) & \text{if } \mathcal{K}_n(D) \in \mathbb{M}_r \\ \infty & \text{otherwise} \end{cases}$$

We now present the conditions that ensure the above mechanism is safe. The first one states that the implementation should preserve the ordering.

[5]Note that redrawing a new random value, when the result is outside $\mathbb{M}_r$, would not be a solution, because it would change the distribution in a way that may cause the loss of differential privacy.

Condition 1. *The implementation respects the order. Namely, for every function $g : \mathbb{R}^j \to \mathbb{R}^k$ and its implementation g', for all $x \in \mathbb{R}^j$ and $y \in \mathbb{R}^j$, $g(x)_i \leq g(y)_i$ implies $g'(x)_i \leq g'(y)_i$ where, for each $i \in [1, k]$, z_i is the projection of z on the i^{th} component.*

The second one states that for a suitable $\mathbb{M}_r \subset \mathbb{R}^m$ there exists a domain $U_r \subset [0, 1[^m$ such that, inside this domain, n and n' are close in U_r, while, outside this domain, U_r the result determined by n falls out of (the expansion of) $\mathbb{M}_r$, and hence gets truncated.

Condition 2. *For some k and ξ_n in $\mathbb{R}^+$, and sets $U_r \subset [0, 1[^m$ and $\mathbb{M}_r \subset \mathbb{R}^m$*

(i) for all $x \in \mathbb{M}_r$, n and n' are (k, ξ_n)-close on U_r

(ii) $\forall u \notin U_r,\ f(D) + n(u) \notin \text{expand}(\mathbb{M}_r, +k\xi_0 + \xi_n)$

The above condition may look strong, but in fact it is a quite reasonable assumption. For instance, it is granted when $\|f\|_\infty < \infty$ and n is absolutely continuous. In such cases, indeed, we can construct k, ξ_n, U_r and $\mathbb{M}_r$ in the following way.

Construction 3.1. *First choose U_r to be a closed subset of $[0, 1[^m$.* [6] *Then set k to be the maximal value for the derivative of n in U_r, whose existence is ensured by the absolute continuity of n, and set ξ_n to be a bound for $\|n - n'\|_\infty$ in U_r.* [7] *Finally, choose $\mathbb{M}_r$ to be the largest set that satisfies $U_r \subseteq n^{-1}(\text{expand}(\mathbb{M}_r, +\|f\|_\infty + k\xi_0 + \xi_n))$.*

Proposition 3. *k, ξ_n, U_r and $\mathbb{M}_r$ defined in Construction 3.1 satisfy Condition 2.*

Proof The first part of Condition 2 follows from Proposition 1. For the second part, let $f(D) + n(u) \in \text{expand}(\mathbb{M}_r, +k\xi_0+\xi_n)$. Then $n(u) \in \text{expand}(\mathbb{M}_r, +\|f\|_\infty+k\xi_0+\xi_n)$ and therefore $u \in \mathbb{M}_r$. □

Condition 2 ensures that the implementation of the noise behaves well inside U_r, and that outside U_r we do not need to worry about the value of n because the result (in the ideal semantics) gets truncated. However, we still need to worry about the result of n' outside U_r, because nobody ensures that the result gets truncated also in the implementation. More precisely, it could be that for some value far outside U_r, n' returns a value inside $\mathbb{M}_r$. To avoid this, we require the implementation to be monotonic with respect to the ideal semantics (Condition 1).

Another way to understand the necessity of the monotonicity condition is the following: If we do not require this property, we could have an implementation of n' such that for all $u \notin \mathbb{M}_r$, $n'(u) = 0$. Such an implementation would be valid with respect to Condition 2 because we do not ask for closeness of n and n' outside of U_r. Then, for all $u \notin U_r$, $\mathcal{K}'_{\mathbb{M}_r}(D)(u) = f'(D)$ will not be truncated if the domain of f' is a subset of $\mathbb{M}_r$. Therefore the probability that the returned answer by $\mathcal{K}'_{\mathbb{M}_r}$ is just the true answer becomes much higher than expected.

Finally, the algorithm tests whether the result is in $\mathbb{M}_r$. Let t be the test function, i.e., $t(x) = 1$ if $x \in \mathbb{M}_r$ and $t(x) = 0$ otherwise. The implementation t' in general does not coincide with t,

[6] The choice of U_r influences the utility and the optimal set could be obtained by a fix-point construction, but this is out of the scope of this paper.

[7] When n is a well-known function, often there exist implementations which are full precision, in the sense that the error is only due to the fact the result has been rounded to the nearest representable number. In such cases computing ξ_n is easy.

and therefore it accepts a set $\mathbb{M}_r{}' = t'^{-1}(\{1\})$ different from $\mathbb{M}_r$. Hence, we need to ensure that the implementation of the test is safe, namely that in the implementation we do the truncation every time that we would do it in the ideal semantics. This can be ensured with the following condition.

Condition 3. *The test t' is such that for all $x \in \mathbb{R}^m$, $t(x) = 0$ implies $t'(x) = 0$ i.e. $\mathbb{M}_r{}' \subseteq \mathbb{M}_r$.*

We are now able to provide a bound on the maximal error of the implementation of Mechanism 3. For stating the precise property, we first define the notion of idealized implementations $\widehat{g}$ of any function g, which coincides with g outside of the domain U_r and which coincides with g' inside of the domain U_r:

$$\widehat{g}(u) = \begin{cases} g'(u) \text{ if } u \in U_r \\ g(u) \text{otherwise} \end{cases} .$$

$\mathcal{K}'_{\mathbb{M}_r{}'}$ and $\widehat{\mathcal{K}_n}$ differ mainly because the condition on $\mathcal{K}'_{\mathbb{M}_r{}'}$ is about the result of the noise function while the condition in $\widehat{\mathcal{K}_n}$ is about u i.e. the domain of the noise function. In the following proposition, however, we relate $\mathcal{K}'_{\mathbb{M}_r{}'}$ to $\widehat{\mathcal{K}_n}$ which allow us to finally prove Proposition 2.

Proposition 4. *When Conditions 1, 2 and 3 hold, the implementation $\mathcal{K}'_{\mathbb{M}_r{}'}$ of Mechanism 3 satisfies the following:*

$$\mathcal{K}'_{\mathbb{M}_r{}'}(D) = \begin{cases} \widehat{\mathcal{K}_n}(D) & \text{if } \widehat{\mathcal{K}_n}(D) \in \mathbb{M}_r{}' \\ \infty & \text{otherwise} \end{cases}$$

Moreover, $\mathcal{K}'_{\mathbb{M}_r{}'}(D)$ and $\mathcal{K}_n$ satisfy Proposition 2 for $\xi_t = k\xi_0 + \xi_n$.

Proof By Condition 2 (ii) we have $\forall D, \forall u \notin U_r, f(D) + n(u) \notin \text{expand}(\mathbb{M}_r, +k\xi_0 + \xi_n)$. Then by Condition 2 (i) and Condition 3 we have $\mathbb{M}_r{}' \subseteq f(D) + n' \circ n_0(U_r)$. By Condition 1 we derive $\forall u \notin U_r, f(D) + n' \circ n_0(u) \notin \mathbb{M}_r{}'$. Since $n' \circ n_0(u)$ and $\widehat{n'} \circ \widehat{n_0}$ coincide on U_r, we have that for all $u \in U_r, t'(f(D) + n' \circ n_0(u)) = t'(f(D) + \widehat{n'} \circ \widehat{n_0})$. Moreover $\widehat{\mathcal{K}_n}(D)(u) \in \mathbb{M}_r{}'$ implies $u \in U_r$ and therefore $f'(D) + \widehat{n'} \circ \widehat{n_0} \circ U = f'(D) + n' \circ n_0(U)$. So the first part of the proposition is proven.

For the second part, we recall that $\widehat{n}$ and n are (k, ξ_n)-close on U_r. Therefore by composition of closeness with n_0, and since $\widehat{n'} \circ \widehat{n_0} = n' \circ n_0$ on U_r, we have the property on U_r. Then, since $\widehat{n'} \circ \widehat{n_0} = n$ on $[0, 1[^m \backslash U_r$, we have the property on the whole domain $[0, 1[^m$.

□

3.6 Relaxing the differential privacy property

We now consider the case of the weaker notion of (ϵ, δ)-differential privacy. We will see that, in this case, the only condition we need is the closeness of the implementation in some subdomain U_r.

Condition 4. *For some k and ξ_n in $\mathbb{R}^+$, and some set $U_r \subset [0, 1[^m$ n and n' are (k, ξ_n)-close on U_r.*

This Condition implies Proposition 2 for $\xi_t = k\xi_0 + \xi_n$ of the idealized implementation of n since, by definition, outside U_r the idealized implementation coincides with the exact function. The next section is mainly focused on the first option (Conditions 1-3 and ϵ-differential privacy on the truncated mechanism) but all propositions apply also for this second option (Condition 4 and (ϵ, δ)-differential privacy on the (untruncated) mechanism). In Section 4.4, Theorem 2 states ϵ'-differential privacy for the finite-precision of the truncated mechanism. Proposition 11 states that the idealized implementation of the mechanism is also ϵ'-differentially private. From the latter, we then explain how to prove (ϵ', δ)-differential privacy for the mechanism in the actual implementation.

4 Quantification of the loss of differential privacy due to computational errors

In the previous section we provided a bound ξ_t on the computational error for Mechanism 3 and $\widehat{\mathcal{K}}_n$. In this section, we use this bound to quantify the loss of differential privacy. In Section 4.1, we explain, in a general way, how to obtain bounds on probability distributions from bounds on computational errors. In Section 4.2, we show that, unfortunately, these bounds are too loose when considering the probability of a precise answer. We then propose a natural solution: rounding the result which prevents accessing to a precise answer. Then, Section 4.3 addresses a last minor issue that might slightly influence the result when the distribution used to generate the noise is not smooth. Finally, we provide our main result: a bound on the increase of the ϵ, on Mechanism 4, due to the implementation error. We also prove, as a side result, that Mechanism 5 (the one without truncation) is (ϵ', δ)-differentially private.

4.1 Modeling the error as a distance between distributions

Given that we are in a probabilistic setting, it is not so useful to measure the errors due to the finite representation in terms of numerical difference. We should rather measure them in terms of distance between the theoretical distribution and the actual distribution. In this way, we will be able to establish a link with differential privacy, which is indeed, basically, a notion of distance between distributions.

In the literature one can find many definitions of distance between distributions. Here we consider the ∞-Wassertein distance [3] since, as we will see, it has a direct relation with differential privacy. We will use $\Gamma(\mu, \nu)$ to denote the set of all measures on $\mathbb{R}^m \times \mathbb{R}^m$ with marginal μ and ν respectively. Namely, every $\gamma \in \Gamma(\mu, \nu)$ is a measure such that

$$\gamma(S \times \mathbb{R}^m) = \mu(S) \text{ (left marginal)} \quad \text{and} \quad \gamma(\mathbb{R}^m \times S) = \nu(S) \text{ (right marginal)}$$

We also denote by $\text{supp}(\gamma)$, the set of points where γ is non zero.

Definition 15 (∞-Wassertein distance [3])**.** Let μ, ν be two probability measures on $(\mathbb{R}^m, \mathcal{S})$ for which there exists a compact C such that $\mu(C) = \nu(C) = 1$ (i.e, the support of μ and ν is bounded).

The ∞-Wassertein distance between μ and ν is defined as:

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

We extend this definition to any pair of measures μ and ν that differs only on a compact ($\mathbb{M}_r$ in our case) by restricting $\Gamma(\mu,\nu)$ to be null on all sets $X\times Y$ such that $X\cap Y=\emptyset$, $X\subset\mathbb{R}^m\setminus\mathbb{M}_r$ and $Y\subset\mathbb{R}^m\setminus\mathbb{M}_r$.

We can now quantify the distance between two random variables:

Proposition 5. *Let X and X' be two random variables $\Omega\to\mathbb{R}^m$ with distribution μ and ν respectively. We have that $\|X-X'\|_\infty\le\xi$ implies $W_\infty(\mu,\nu)\le\xi$.*

Proof Let A, B be two measurable sets distant by more than ξ, namely: $\forall a\in A, b\in B, d(a,b)>\xi$. Then $\|X-X'\|_\infty\le\xi$ implies that there is no $\omega\in\Omega$ such that $X(\omega)\in A$ and $X'(\omega)\in B$. Hence $P(X\in A\wedge X'\in B)$ is null which means that the support of (μ,ν) does not include any such pair of sets (A,B), and therefore we can construct a $\gamma\in\Gamma(\mu,\nu)$ such that $\sup_{(x,y)\in\mathrm{supp}(\gamma)} d(x,y)\le\xi$. □

The following proposition allows us to use the ∞-Wassertein distance between μ and ν to establish a bound on μ in terms of ν.

Proposition 6. *Let μ,ν be two probability measures on $\mathbb{R}^m$. Then:*

$$W_\infty(\mu,\nu)\le\xi \implies \forall S\in\mathcal{S}, \mu(\mathrm{expand}(S,-\xi))\le\nu(S)\le\mu(\mathrm{expand}(S,+\xi))$$

Proof Let $\gamma\in\Gamma(\mu,\nu)$. Since ν is the right marginal of γ, $\nu(S)=\int_{\mathbb{R}^m\times S} d\gamma(x,y)$. Since $\gamma(x,y)=0$ if $d(x,y)>\xi$, we derive $\nu(S)=\int_{\mathrm{expand}(S,+\xi)\times S} d\gamma(x,y)$. Therefore, $\nu(S)\le\int_{\mathrm{expand}(S,+\xi)\times\mathbb{R}^m} d\gamma(x,y)$. The last expression is the left marginal of γ in $\mathrm{expand}(S,+\xi)$, hence by definition: $\nu(S)\le\mu(\mathrm{expand}(S,+\xi))$.

The other inequality is obtained by replacing S with $\mathbb{R}^m\setminus S$. □

We can now provide upper and lower bounds on the probabilities of the implemented noise in terms of those in the exact semantics and the computational error ξ_t:

Corollary 2. *Let μ,ν be the probability measures of $n\circ U$, $\widehat{n'}\circ\widehat{n_0}\circ U$, respectively. Namely, for all $S\in\mathcal{S}$ (where we recall that $\mathcal{S}$ is the set of measurable sets in $[0,1[^m$), $\mu(S)=P[n\circ U\in S]$ and $\nu(S)=P[\widehat{n'}\circ\widehat{n_0}\circ U\in S]$. Then, for all $S\in\mathcal{S}$:*

$$\mu(\mathrm{expand}(S,-\xi_t))\le\nu(S)\le\mu(\mathrm{expand}(S,+\xi_t))$$

Proof By Proposition 2 and Proposition 5 it follows that $W_\infty(\mu,\nu)\le\xi_t$. Then apply Proposition 6. □

However, used carelessly, the lower bound provided by the previous corollary is not accurate enough. There is in fact a problem when S is too small: $\mathrm{expand}(S,-\xi_t)$ can be much smaller than S or even be the empty set. Consequently, the probability of the returned answer to be in $\mathrm{expand}(S,-\xi_t)$ will be close to 0, or even 0. Since differential privacy is about ratio, this case would not have a suitable bound. Note that this is exactly the problem described in Example 1

4.2 Rounding the answer

In the last paragraph of the previous section we pointed out a problem for differential privacy that may arise when the analyst measures sets that are too small. The most natural way to solve this problem consists of rounding the answer, which has the effect of collapsing all values from a neighborhood into a unique value. In this approach, once the computation of $\mathcal{K}_{\mathbb{M}_r}(D)$ is achieved, we do not return the answer but a "rounding" of the answer through a function *rnd*, whose implementation *rnd'* is defined as any measurable function that maps all values of $\mathbb{M}_r$ into some finite subset F of $\mathbb{M}_r$, such that $\forall x \in F, rnd'(x) = x$ and such that $rnd'(\infty) = \infty$.

If two values close to the boundary of $\mathbb{M}_r$ are rounded to the same one but one is truncated while not the other one, some problematic behaviors can happen. To avoid that we add an additional condition about *rnd'* and $\mathbb{M}_r{}'$.

Condition 5. *The function rnd' is such that* $rnd'^{-1}(F) = \mathbb{M}_r{}'$.

Remark 1. To achieve this property we assume we already get $\mathbb{M}_{r0}$ that has all other properties (it may have been obtained with the method in Construction 3.1) and that *rnd'* is a rounding function for $\mathbb{M}_{r0}$ which maps to F_0. Then we build $\mathbb{M}_r \subseteq \mathbb{M}_{r0}$ that has Condition 5. We define $L = \max_{x \in F_0} \varnothing(rnd'^{-1}(x))$. Then Condition 5 holds for any set $\mathbb{M}_r{}' = rnd'^{-1}(\mathbb{M}_{r1})$ where $\mathbb{M}_{r1} \subseteq \text{expand}(\mathbb{M}_{r0}', -L))$.

Proof We have at first *rnd'* such that all values of $\mathbb{M}_{r0}$ maps to F_0, Since $\mathbb{M}_{r1} \subseteq \text{expand}(\mathbb{M}_{r0}', -L))$ we have $rnd'^{-1}(\mathbb{M}_{r1}) \subseteq \mathbb{M}_{r0}$ $\mathbb{M}_r{}' \subseteq \mathbb{M}_{r0}$ and so $rnd'^{-1}(\mathbb{M}_r{}') \subseteq \mathbb{M}_{r0}$. We consider $F = rnd'(\mathbb{M}_{r1})$, since $\mathbb{M}_{r1} \subseteq \mathbb{M}_{r0}$, *rnd'* satisfies $\forall x \in F, rnd'(x) = x$ and all values of $\mathbb{M}_r{}'$ map to $\mathbb{M}_{r1}$, *rnd'* satisfies all other conditions.

On the other hand, Condition 5 can be rewritten as: $rnd'^{-1}(rnd(\mathbb{M}_{r1})) = rnd'^{'-1}(\mathbb{M}_{r1})$. Since $rnd'(\mathbb{M}_{r1}) \subseteq \mathbb{M}_{r1}$ we have $rnd'^{-1}(rnd(\mathbb{M}_{r1})) \subseteq rnd'^{'-1}(\mathbb{M}_{r1})$. For the other inclusion, if $x \in rnd'^{'-1}(\mathbb{M}_{r1})$, $rnd'(rnd'(x)) \in \mathbb{M}_{r1}$ since *rnd'* is idempotent on $\mathbb{M}_{r0}$ and $rnd'^{-1}(\mathbb{M}_{r1}) \subseteq \mathbb{M}_{r0}$. This means that $x \in rnd'^{-1}(rnd'(\mathbb{M}_{r1}))$. □

Mechanism 4. *Our final mechanism is* $rnd(\mathcal{K}_{\mathbb{M}_r})$*, where rnd is a rounding function.*

In Figure 3, we represent the probability distribution of the final mechanism with the same settings as Figure 2. Now, large values like x_M are truncated and the discrepancy that caused a violation of differential privacy at x is now softened due to rounding.

Although the sets corresponding to the elements of F form a finite collection, it will be convenient, for the sake of uniformity, to consider them as the generators of a σ-algebra. Hence, we define $\mathcal{S}'_0 = \{rnd'^{-1}(x) | x \in F\}$, and $\mathcal{S}'$ as the σ-algebra generated by $\mathcal{S}'_0$. Our mechanism allows us to restrict our analysis of probabilities only to $\mathcal{S}'$.

Proposition 7. $\forall S \in \mathcal{S}', P[\mathcal{K}'_{\mathbb{M}_r{}'}(D_1) \in S] \leq \exp(\epsilon')P[\mathcal{K}'_{\mathbb{M}_r{}'}(D_2) \in S]$ *implies* $rnd'(\mathcal{K}'_{\mathbb{M}_r{}'})$ *is* ϵ'*-differentially private :*
$\forall S \in \mathcal{S}, P[rnd'(\mathcal{K}'_{\mathbb{M}_r{}'})(D_1) \in S] \leq \exp(\epsilon')P[rnd'(\mathcal{K}'_{\mathbb{M}_r{}'})(D_2) \in S]$.

Proof The image of $\mathcal{K}'_{\mathbb{M}_r{}'}$ is $\mathbb{M}_r{}' \cup \infty$ so the image of $rnd'(\mathcal{K}'_{\mathbb{M}_r{}'})$ is $F \cup \{\infty\}$. Therefore, for any set $S \in \mathcal{S}$, $P(rnd'(\mathcal{K}'_{\mathbb{M}_r{}'}) \in S) = P(rnd'(\mathcal{K}'_{\mathbb{M}_r{}'}) \in S \cap (F \cup \{\infty\}))$ which is equivalent to

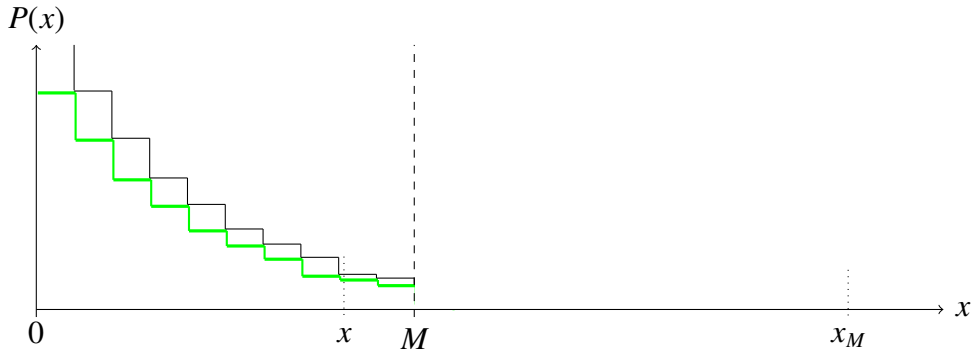

Figure 3: The probability distribution of Mechanism 4. The larger steps represent the rounding (formally each step should be a Dirac in the center of each step domain, but it would be less readable). Here, the truncation is done outside $\mathbb{M}_r = [-M, M]$ but the probability $P(\infty)$ that the result is truncated is not represented here.

$P(rnd'(\mathcal{K}'_{\mathbb{M}_r,'}) \in S) = P(rnd'(\mathcal{K}'_{\mathbb{M}_r,'}) \in (S \cap F) \cup (S \cap \{\infty\}))$. This can be rewritten as $P(rnd'(\mathcal{K}'_{\mathbb{M}_r,'}) \in S) = P(\mathcal{K}'_{\mathbb{M}_r,'} \in rnd'^{-1}(S \cap F) \cup rnd'^{-1}(S \cap \{\infty\}))$. Since Condition 5 ensures that $rnd'^{-1}(F) \subseteq \mathbb{M}_r{}'$, for all $S \subseteq F$, $rnd'(\mathcal{K}'_{\mathbb{M}_r,'})^{-1}(rnd^{-1}(S)) = \mathcal{K}'^{-1}_{\mathbb{M}_r,'}(rnd^{-1}(S))$. Then since Condition 5 ensures that $rnd'^{-1}(F) \supseteq \mathbb{M}_r{}'$, $rnd'(\mathcal{K}'_{\mathbb{M}_r,'})^{-1}(\{\infty\}) = \mathcal{K}'^{-1}_{\mathbb{M}_r,'}(\{\infty\})$. From these two equalities, we get $P(rnd'(\mathcal{K}'_{\mathbb{M}_r,'}) \in S) = P(\mathcal{K}'_{\mathbb{M}_r,'} \in rnd'^{-1}(S \cap F) \cup rnd'^{-1}(S \cap \{\infty\}))$. This can be rewritten as: $P(rnd'(\mathcal{K}'_{\mathbb{M}_r,'}) \in S) = P(\mathcal{K}'_{\mathbb{M}_r,'} \in rnd'^{-1}(S \cap (F \cup \{\infty\})))$. Since for all $S \in \mathcal{S}$, $rnd'^{-1}(S \cap (F \cup \{\infty\})) \in \mathcal{S}'$, we get the implication. □

We now quantify the efficiency of the rounding function with respect to helping to establish bounds on ratios between the probability measures of the implementation. For this purpose, we introduce the notion of ratio between the bounds established in Corollary 2. This notion will be used in Theorem 2.

Definition 16 (Rounding ratio)**.** We define R to be the maximal ratio between the areas $\text{expand}(S, +\xi_t)$ and $\text{expand}(S, -\xi_t)$ over all values that can be returned:

$$R = \max_{S \in \mathcal{S}'_0 \setminus \emptyset} \frac{\lambda(\text{expand}(S, \xi_t) \setminus \text{expand}(S, -\xi_t))}{\lambda(\text{expand}(S, -\xi_t))} \tag{4}$$

Where λ is the Lebesgue measure on $\mathbb{R}^m$.

The R defined by (4) may look quite hard to compute, however, when rnd is particularly regular, we can bound R from above:

Proposition 8. *Assume the rounding function* $\mathrm{rnd}_l : \mathbb{R}^m \to \mathbb{R}^m$ *is defined as* $\mathrm{rnd}_l(x_1, \ldots, x_m) = (\lfloor lx_1 \rfloor/l, \ldots, \lfloor lx_m \rfloor/l)$. *Then,*

$$R \leq \left(\frac{l + 2\xi_t}{l - 2\xi_t}\right)^m - 1$$

Proof With such a rounding, the space is split in hypercubes C of uniform size l. Therefore $\mathrm{expand}(C, \xi_t)$ is included in a hypercube of size $l + 2\xi_t$ and $\mathrm{expand}(C, -\xi_t)$ is exactly a hypercube of size $l - 2\xi_t$. The Lebesgue measure of an hypercube of size s is by definition s^m. Hence the measure of $\mathrm{expand}(C, -\xi_t)$ is $(l - 2\xi_t)^m$ and the measure of $\mathrm{expand}(C, \xi_t) \setminus \mathrm{expand}(C, -\xi_t)$ is $(l + 2\xi_t)^m - (l - 2\xi_t)^m$. □

4.3 Strengthening the differential privacy property

The last issue, in order to be able to quantify the loss due to the finite precision, comes from the definition of differential privacy. This definition relies on the definition of sensitivity in the exact semantics while here we need the sensitivity in the finite precision semantics. More precisely, if the noise has a density function p, then differential privacy can be expressed by the following inequality.

$$\forall x, y \in \mathbb{R}^m, d(x, y) \leq \Delta_f \implies p(x) \leq \exp(\epsilon) p(y) \tag{5}$$

In the past sections we have been able to establish a relation between the implemented noise p' and the ideal noise p, but we do not know anything about the relation between the implemented sensitivity $\Delta_{f'}$ and the ideal one, Δ_f, so we cannot use (5) directly to derive conclusions about the level of privacy satisfied by the implementation.

There are several ways to solve this problem. Here, we choose to constrain the distribution of X. Indeed, when using a Laplace distribution (which is the most used), adding this constraint allows us to keep the bound as accurate as if there was no problem about the distribution.

Condition 6. *Given a mechanism* $\mathcal{K}(D) = f(D) + X$ *we say that* $\mathcal{K}$ *satisfies Condition 6 with parameter* ϵ *(the desired parameter of differential privacy) if the random variable X has a probability distribution which is absolutely continuous according to the Lebesgue measure, and*

$$\forall S \in \mathcal{S}, r_1, r_2 \in \mathbb{R}^m, P[r_1 + X \in S] \leq \exp(\epsilon d(r_1, r_2)/\Delta_f) P[r_2 + X \in S]$$

This property is actually stronger than differential privacy as we state in the following proposition.

Proposition 9. *Condition 6 implies that the mechanism* $\mathcal{K}(D) = f(D) + X$ *is* ϵ*-differentially private.*

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 $\exp(\epsilon^{d(r_1,r_2)/\Delta_f}) \leq \exp(\epsilon)$. Hence, $P[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) P[\mathcal{K}(D_2) \in S]$. □

We chose this Condition because in case the noise has a Laplacian distribution then the converse of Proposition 9 holds and therefore in this case Condition 6 and differential privacy are equivalent.

Proposition 10. *Let $\mathcal{K}(D) = f(D) + X$ be a mechanism, and assume that X is Laplacian. If $\mathcal{K}$ is ϵ-differentially private (w.r.t. f), then Condition 6 holds.*

Proof First, we show that if $\mathcal{K}$ is ϵ-differentially private then $b \leq \epsilon/\Delta_f$ 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 $S \in \mathcal{S}$: $P[f(D_1) + X \in S] \leq \exp(\epsilon)P[f(D_2) + X \in S]$. From the density function of the Laplace noise (Definition 6), we derive: $K(n, d)d\lambda \leq \exp(\epsilon)K(n, d)\exp(-b\Delta_f)d\lambda$. Hence,

$$b \leq \epsilon/\Delta_f. \tag{6}$$

Now, by definition of the density function, we have:
$P[r_2 + X \in S] = \int_{x\in S} K(n, d)\exp(-bd(x, r_2))d\lambda$. From the triangular inequality, we derive: $P[r_2 + X \in S] \geq \int_{x\in S} K(n, d)\exp(-b(d(x, r_1) + d(r_1, r_2)))d\lambda$.
Hence, $P[r_2 + X \in S] \geq \exp(-bd(r_2, r_1)) \int_{x\in S} \exp(-bd(r_1, x))d\lambda$. From inequality (6), we derive: $P[r_2 + X \in S] \geq \exp(-\epsilon d(r_2,r_1)/\Delta_f) \int_{x\in S} \exp(-bd(r_1, x))d\lambda$. Finally, $P[r_2 + X \in S] \geq \exp(-\epsilon d(r_2,r_1)/\Delta_f)P[r_1 + X \in S]$. □

4.4 Preserving differential privacy

Now we have established all necessary conditions and we can finally state our main theorem.

Theorem 2. *If mechanism $rnd(\mathcal{K}_{\mathbb{M}_r})$ (Mechanism 4) has an implementation $rnd'(\mathcal{K}'_{\mathbb{M}_{r'}})$ such that Conditions 1, 2, 3 and 5 hold and the density function of $n(U)$ satisfies Condition 6, then the implemented mechanism is ϵ'-differentially private, namely:*
$\forall S \in \mathcal{S}, P[rnd(\mathcal{K}_{\mathbb{M}_r})'(D_1) \in S] \leq \exp(\epsilon')P[rnd(\mathcal{K}_{\mathbb{M}_r})'(D_2) \in S]$ *where*

$$\epsilon' = \epsilon + \ln\left(1 + R\exp\left(\epsilon\left(1 + \frac{L + \xi_t}{\Delta_{f'}}\right)\right)\right)$$

with R corresponding to Definition 16, $L = \max_{S\in\mathcal{S}'_0} \varnothing S$ and ξ_t as defined in Proposition 4.

Proof According to Proposition 7, we only have to consider Mechanism 2 on the sigma algebra $\mathcal{S}'$. Moreover since $\mathcal{S}'$ is generated by a finite number of disjoint sets, we only have to prove the property for all $S \in \mathcal{S}'_0$. In fact we will prove the property only for all $S \in \mathcal{S}'_0 \setminus \{\infty\}$. Indeed, once this has been proved the fact that $P(rnd(\mathcal{K}_{\mathbb{M}_r})'(D_i) \in \{\infty\}) = 1 - P(rnd(\mathcal{K}_{\mathbb{M}_r})'(D_i) \in \bigcup_{S\in\{\mathcal{S}'_0\setminus\{\infty\}\}} S)$ implies the property for $\{\infty\}$.

So, we consider $S \in \mathcal{S}'_0 \setminus \{\infty\}$. In that case, $\mathcal{K}'_{\mathbb{M}_{r'}} = \widehat{\mathcal{K}_n}$. We define $P_1 = P[\widehat{\mathcal{K}_n}(D_1) \in S]$ and $P_2 = P[\widehat{\mathcal{K}_n}(D_2) \in S]$. Since ν is the measure associated to $\widehat{n'} \circ \widehat{n_0} \circ U$, we have $P_i = \nu(S' - f'(D_i))$.

With Proposition 4 and Proposition 5, we get $d(\nu, \mu) \leq \xi_t$.

From Proposition 6 we derive $P_1 \leq \mu(\text{expand}(S, \xi_t) - f'(D_1))$ and $P_2 \geq \mu(S_2)$ where $S_2 = \text{expand}(S, -\xi_t) - f'(D_2)$. In the following, we denote by S_1 the set $(\text{expand}(S, \xi_t) \setminus \text{expand}(S, -\xi_t)) - f'(D_1)$. The additivity property of measures grants us $\mu(\text{expand}(S, \xi_t) - f'(D_1)) = \mu(\text{expand}(S, -\xi_t) - f'(D_1)) + \mu(S_1)$. Condition 6 can be expressed in terms of the measure as: $\forall S \in \mathcal{S}, r \in$

$\mathbb{R}^m, \mu(S) \leq \exp\left(\epsilon^{\|r\|}/_{\Delta_{f'}}\right)\mu(S-r)$. From this inequality and the one of P_1, we can derive, since $\|f'(D_2) - f'(D_1)\| \leq \Delta_{f'}$: $P_1 \leq \exp(\epsilon)P_2 + \mu(S_1)$.

Since the probability is absolutely continuous according to the Lebesgue measure (Condition 6), we can express the probability with a density function p: $\forall S \in \mathcal{S}, \mu(S) = \int_S p(x)d\lambda$. We derive: $\forall S \in \mathcal{S}, \min_{x\in S} p(x) \leq {}^{\mu(S)}/_{\lambda(S)}$. By applying this property on S_2, we get: $\min_{x\in S_2} p(x) \leq {}^{\mu(S_2)}/_{\lambda(S_2)}$. Since the set is compact, the minimum is reached : $\exists x_0 \in S_2, p(x_0) \leq {}^{P_2}/_{\lambda(S_2)}$. By a triangular inequality, we get : $\forall x \in S_1, d(x, x_0) \leq \Delta_{f'} + L + \xi_t$. Hence, from Condition 6 we derive: $\forall x \in S_1, p(x) \leq \exp\left(\epsilon({}^{\Delta_{f'}+L+\xi_t}/_{\Delta_{f'}})\right)p(x_0)$. Then since $\mu(S_1) = \int_{S_1} p(x)d\lambda$: $\mu(S_1) \leq \exp({}^{\epsilon(\Delta_{f'}+L+\xi_t)}/_{\Delta_{f'}})^{\lambda(S_1)}/_{\lambda(S_2)}P_2$. Because the Lebesgue measure is invariant by translation, we can rewrite this inequality with R from Definition 16: $\mu(S_1) \leq \exp({}^{\epsilon(\Delta_{f'}+L+\xi_t)}/_{\Delta_{f'}})RP_2$. Finally, since we already get $P_1 \leq \exp(\epsilon)P_2 + \mu(S_1)$, we obtain : $P_1 \leq (1 + R\exp({}^{\epsilon(\Delta_{f'}+L+\xi_t)}/_{\Delta_{f'}}))\exp(\epsilon)P_2$.

□

4.5 Results for (ϵ', δ)-differential privacy

We now consider the case of (ϵ', δ)-differential privacy for the (untruncated) mechanism, under milder conditions.

Mechanism 5. *The mechanism we consider in this section is rnd($\mathcal{K}$).*

Because there is no truncation on the result, Condition 5 is vacuously true and therefore can be removed from the hypothesis.

Before stating this result, we adapt the former Theorem 2 to Mechanism 5.

Proposition 11. *If Mechanism 5 has an ideal implementation rnd($\widehat{\mathcal{K}_n}$) such that Condition 4 holds for the set U_r and the density function of $n(U)$ satisfies Condition 6, then the ideally implemented mechanism is ϵ'-differentially private, namely:*
$\forall S \in \mathcal{S}, P[rnd(\widehat{\mathcal{K}_n}(D_1)) \in S] \leq \exp(\epsilon')P[rnd(\widehat{\mathcal{K}_n})(D_2) \in S]$ *where* $\epsilon' = \epsilon + \ln\left(1 + R\exp\left(\epsilon\left(1 + {}^{L+\xi_t}/_{\Delta_{f'}}\right)\right)\right)$
with R corresponding to Definition 16, $L = \max_{S\in\mathcal{S}'_0} \varnothing S$ *and ξ_t as defined in Proposition 4.*

Proof Condition 4 implies Proposition 2 for $(\widehat{\mathcal{K}_n}, \mathcal{K}_n)$. Otherwise the proof is similar of the one of Theorem 2 except that the finite precision semantics is now the idealized semantics. □

Theorem 3. *If mechanism rnd($f(D_1) + n(U)$) has an implementation rnd'($f(D_1) + \widehat{n'} \circ \widehat{n_0} \circ U$) such that Condition 4 holds and the density function of $n(U)$ satisfies Condition 6, then the implemented mechanism is ϵ', γ-differentially private, namely:*

$$\forall S \in \mathcal{S}, P[rnd'(f(D_1) + n' \circ n_0 \circ U) \in S] \leq \exp(\epsilon')P[rnd'(f(D_2) + n' \circ n_0 \circ U) \in S] + \gamma$$

where ϵ' and R are the same as in Theorem 2, $\gamma = (1 + \exp(\epsilon'))(1 - \lambda(U_r))$ and U_r is the set for which Conditions 4 is satisfied.

The proof consists of bounding the probability that the idealized implementation differs from the actual one.

Proof We have, by set decomposition: $P[rnd'(f(D_1) + \widehat{n'} \circ \widehat{n_0} \circ U) \in S] = P[rnd'(\mathcal{K}'(D_1)) \in S \cap U \in U_r] + P[rnd'(\mathcal{K}'(D_1)) \in S \cap U \notin U_r]$. Moreover, due to set inclusion, we have: $P[rnd'(\mathcal{K}'(D_1)) \in S \cap U \in U_r] \leq P[rnd'(\mathcal{K}'(D_1)) \in S]$ and $P[rnd'(\mathcal{K}'(D_1)) \in S \cap U \notin U_r] \leq P[U \notin U_r]$. By applying Proposition 11, we have: $P[rnd'(\widehat{\mathcal{K}_n}(D_1)) \in S] \leq \epsilon' P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S]$. Here, we decompose the probability: $P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S] = P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S \cap U \in U_r] + P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S \cap U \notin U_r]$. Because implemented mechanism and idealized implementation coincide on U_r, we have: $P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S \cap U \in U_r] \leq P[rnd'(f(D_2) + \widehat{n'} \circ \widehat{n_0} \circ U) \in S]$ and due to set inclusion we have $P[rnd'(\widehat{\mathcal{K}_n}(D_2)) \in S \cap U \notin U_r] \leq P[U \notin U_r]$. Finally, by putting together the above equalities and inequalities, we get: $P[rnd'(f(D_1) + \widehat{n'} \circ \widehat{n_0} \circ U) \in S] \leq \epsilon'(P[rnd'(f(D_2) + \widehat{n'} \circ \widehat{n_0} \circ U) \in S] + P[U \notin U_r]) + P[U \notin U_r]$. The final result follows from the fact that, since U is uniform, we have $P[U \notin U_r] = \lambda([0, 1[^m \backslash U_r)$. □

5 Case study: the Laplacian noise in one dimension

In this section, we aim at providing a bound to the degradation of privacy in a typical case: we consider a query that sums all entries of a database where each entry is between 0 and 1, which implies that Δ_f is 1. We assume that the database has 2^N entries. We assume that the added noise is a Laplacian noise of parameter ϵ, so that in the exact semantics Mechanism 4 achieves ϵ-differential privacy. To implement this noise, we choose $n(u, s) = \epsilon^{-1} s \ln(u)$ where u is uniform in $]0, 1]$ and s is uniform in $\{-1, 1\}$. We define the utility of the answer to be σ, the standard deviation of the noise divided by the number of entries: $\sigma = 2^{-N} \sqrt{\int_{u \in [0,1]} n^2(u)} = 2^{-N} \epsilon^{-1}$. The value $\log_2(\sigma)$ then corresponds to the number d of significant bits that analysts can rely on when they want to get an average on all entries (we assume they already know the total number of entries). In the following, we express ϵ as a function of d: $\epsilon = 2^{N-d}$. Since ϵ has to be small, we consider $\epsilon \leq 1$ and so $d \leq N$.

About the implementation, we assume that we use a fixed-point representation which has p bits for the fractional parts of the number. I.e., representable numbers are in the set $\{n2^{-p} | n \in \mathbb{Z}\}$. We assume also that the function ln is implemented in full precision: the error due to the function is at most 2^{-p}. [8] We define the loss due to the finite implementation as a relative error with a parameter x: $\epsilon' = (1 + 2^{-x})\epsilon$. Since a small x means that privacy would be must worst than it should be, we assume $x \geq 2$ (i.e. a maximum 25% loss due to implementation). Finally, given that the standard deviation of the answer is 2^{N-d}, rounding the result to a multiple of 2^{N-d} seems a reasonable choice. With this choice, the parameter of the Definition 16 is $L = 2^{N-d}$.

The goal of this section is to find a relation between p and the other parameters N, d and x. Then, we study the corresponding scenario for (ϵ, δ)-differential privacy.

5.1 Truncation to achieve ϵ-differential privacy

Since the sum of the entries is in the range $[0, 2^N]$, it is natural to truncate any result which is outside this range, i.e., $\mathbb{M}_r = [0, 2^N]$. By doing so, we obtain the following bound:

[8] All standard libraries implementing ln are full-precision so this seems a reasonable assumption.

Proposition 12. *When $p \geq x - d + N + \ln(2)^{-1}(3 + 2^d) - 3$, the implementation of Mechanism 4 is $(1 + 2^{-x})2^{N-d}$-differentially private.*

Proof We only consider the case of $s = 1$ since the case $s = -1$ is similar. To satisfy Condition 2, we have to find ξ_t such that n and n' are (k, ξ_n)-close on $[-2^N, 2^N]$ but we can restrict to $[-2^N, 0]$ since we are considering $s = 1$. The derivative of the function n is $\delta n(u)/\delta u = 2^{N-d}u^{-1}$. Its maximum is reached when $n(u) = 2^N$, i.e., for $u = \exp(-2^d)$. Since a function whose the absolute value of the derivative is bounded by k is k-Lipschitz, we can set: $k = 2^{N-d}\exp(2^d)$. Since ln is full precision and multiplication by a constant multiplies the error by this constant, we have $\xi_t = 2^{-p+N-d}\exp(2^d)$.

According to Theorem 2, since $\ln(1 + x) \leq x$ for $x > 0$, we have $\epsilon' = \epsilon + R\exp(\epsilon(1 + L + \xi_t))$. Then since we fixed $\epsilon \leq 1$, $\epsilon L = 1$, and since ξ_t has to be less then L, or otherwise the ratio R of Definition 16 would be infinite, we have: $R \leq 2^{-x+d-N}\exp(-3)$. By replacing R with its definition ($R = {}^{4\xi_t}/_{L-2\xi_t}$), we get: ${}^{4\xi_t}/_{L-2\xi_t} \leq 2^{-x+d-N}\exp(-3)$. This can be rewritten as: $4\xi_t \leq (L - 2\xi_t)2^{-x+d-N}\exp(-3)$. Hence we get: $\xi_t(4 + 2^{-x+d-N+1}\exp(-3)) \leq 2^{-x+d-N}\exp(-3)L$. Since $L = 2^{N-d}$ and $\xi_t = 2^{-p+N-d}\exp(2^{d+1})$, we get: $2^{-p+N-d}\exp(2^d)(4 + 2^{-x+d-N+1}\exp(-3)) \leq 2^{-x+d-N}\exp(-3)2^{N-d}$. A simple rewriting leads to: $2^{-p} \leq {}^{(2^{-x+d-N}\exp(-3-2^d))}/_{(4+2^{-x+d-N+1}\exp(-3))}$. Finally, since $x \geq 2$ we have $2^{-x+d-N+1}\exp(-3) < 4$, which concludes our proof. Note the final switch of the inequality: we proved that the required number of bits is less than the expression, so if a greater number is provided, differential privacy is necessarily granted. □

This result has some over approximations, but the dependency between p and the heaviest term $\ln(2)^{-1}2^d$ is not due to approximations. It is due to the fact that the noise should have an amplitude of at least 2^N to avoid some answer to have probability 0 when queried on extreme databases (all inputs are 1 or 0). Note now that $-2^N = \epsilon^{-1}\ln(u)$ implies $u = \exp(2^d)$, which should be a representable number.

In the case of the Laplacian noise, to avoid having numbers with thousand of bits (which should be the case when $d = 10$), we can use the fact that $\ln(m2^{-E}) = \ln(m) - E\ln(2)$. For that, we draw uniform random variables in $\{0, 1\}$ either p times or $E+P$ where E is the numbers of 0 which has been drawn before a 1 occurs and P the actual precision of the fixed-point implementation which is used. And we consider $u = \ln(m2^{-E})$ where m is made of the P bits occurring after the first sequence of zeros.

Incidentally, we note that there is a small problem in one of the results of [15]. That paper uses the floating-point representation, and the uniform distribution is generated by independently sampling an exponent (from the geometric distribution with parameter $1/2$) and a significand (by drawing a uniform string from $\{0, 1\}^{52}$). In that approach, the value of the exponent corresponds to E and the value of the mantissa to m. However, the paper does not take into account the fact that in the floating point representation the exponent has only 10 bits: this implies that $p = E + P = 1076$ (the mantissa has 52 bits) and from Proposition 12, we get $d \leq 10$ (while what is claimed in [15] corresponds, in our notations, to $d = 47$).

5.2 (ϵ, δ)-differential privacy

We consider now the case of (ϵ, δ)-differential privacy in the same context (sum query, Laplacian noise).

Proposition 13. *When $p \geq x + N + D - d + 10$, the implementation of the mechanism is $(1 + 2^{-x})2^{N-d}, 2^{-D}$-differentially private.*

Proof We fix the set of Condition 4 to be $U_r = [2^{-D-2}, 1]$. On this set, the maximum of the derivative of n, 2^{N+D-d}, is reached when $u = 2^{-D-2}$. Hence, on this set, n is $2^{N+D-d-2}$-Lipschitz. Because of the full-precision of ln, we have $\xi_t = 2^{N+D-d-p+2}$. According to Theorem 3, and since $\ln(1+x) \leq x$ for $x > 0$, we have $\epsilon' = \epsilon + R \exp \epsilon(1 + L + \xi_t)$ and $\delta = 2^{-D-2}(1+(1+2^{-x}) \exp(2^{d-N}))$. Since $(1 + (1 + 2^{-x}) \exp(2^{d-N})) \leq 4$ when $x \geq 2$, we have $\delta \leq 2^{-D}$.

From the above equality about ϵ' and following the lines of previous proof, we get: $\xi_t(4 + 2^{-x+d-N+1} \exp(-3)) \leq 2^{-x+d-N} \exp(-3)L$ and therefore $2^{N+D+2-d-p}(4 + 2^{-x+d-N+1} \exp(-3)) \leq 2^{-x} \exp(-3)$. □

This result shows that, when (ϵ, δ)-differential privacy can be considered a sufficient privacy guarrantee, then it is better to implement (ϵ, δ)–differential privacy instead of ϵ-differential privacy.

6 Conclusion

This paper concerns the influence of rounding errors in mechanisms achieving differential privacy: a security protocol that relies on real-valued random number generation. Existing studies on differential privacy focus either only on the mathematical properties or on technical problems which are particular to a given implementation. Here, we have presented a method to quantify the loss of privacy induced by finite precision. It should be possible to apply this method to most kinds of implementation and computer architecture that achieve differential privacy.

With this model, we have been able to prove that the additive Mechanism 1 which is safe from a theoretical point of view breaks the differential privacy once implemented in a finite precision architecture. This loss has two origins: first, extreme values can cuase significant errors, and secondly, locally, last digits can provide fingerprints of the secret. Our solution solves these two problems: the last digits leakage has been fixed by a rounding procedure and the extreme perturbations by raising an error when the result is outside some range of values.

While proposing these fixes, we have done a quantitative analysis to measure the loss of privacy induced by finite-precision representation. We have proved that when the fixes on the mechanism are implemented, the differential privacy parameter is just increased by some additive factor.

7 Acknowledgements

This work was partially supported by the MSR-INRIA joint lab, by the European Union 7th FP project MEALS, by the project ANR-12-IS02-001 PACE, and by the INRIA Large Scale Initiative CAPPRIS.

References

Bibliography

[1] M. Andrés, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 901–914, 2013.

[2] G. Barthe, B. Köpf, F. Olmedo, and S. Z. Béguelin. Probabilistic relational reasoning for differential privacy. In *Proceedings of the 39th Annual ACM Symposium on Principles of Programming Languages (POPL)*. ACM, 2012.

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

[4] S. Chaudhuri, S. Gulwani, R. Lublinerman, and S. NavidPour. Proving programs robust. In T. Gyimóthy and A. Zeller, editors, *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13rd European Software Engineering Conference (ESEC-13), Szeged, Hungary, September 5-9, 2011*, pages 102–112. ACM, 2011.

[5] T. Chothia, Y. Kawamoto, and C. Novakovic. A tool for estimating information leakage. In N. Sharygina and H. Veith, editors, *Computer Aided Verification*, volume 8044 of *Lecture Notes in Computer Science*, pages 690–695. Springer Berlin Heidelberg, 2013.

[6] T. Chothia, Y. Kawamoto, and C. Novakovic. Leakwatch: Estimating information leakage from java programs. In M. Kutylowski and J. Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II*, volume 8713 of *Lecture Notes in Computer Science*, pages 219–236. Springer, 2014.

[7] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In S. Vaudenay, editor, *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[9] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *In Proceedings of the Third Theory of Cryptography Conference (TCC)*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[10] M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan, and B. C. Pierce. Linear dependent types for differential privacy. In R. Giacobazzi and R. Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'13)*, pages 357–370. ACM, 2013.

[11] I. Gazeau, D. Miller, and C. Palamidessi. A non-local method for robustness analysis of floating point programs. In H. Wiklicky and M. Massink, editors, *Proceedings of the 10th Workshop on Quantitative Aspects of Programming Languages and Systems (QAPL 2012)*, volume 85 of *EPTCS*, pages 63–76, 2012.

[12] S.-S. Ho and S. Ruan. Differential privacy for location pattern mining. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL)*, pages 17–24. ACM, 2011.

[13] IEEE Task P754. *IEEE 754-2008, Standard for Floating-Point Arithmetic*. IEEE, Aug. 2008.

[14] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In G. Alonso, J. A. Blakeley, and A. L. P. Chen, editors, *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 277–286. IEEE, 2008.

[15] I. Mironov. On significance of the least significant bits for differential privacy. In T. Yu, G. Danezis, and V. D. Gligor, editors, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 650–661. ACM, 2012.

[16] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pages 173–187. IEEE Computer Society, 2009.

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

MEALS Partner Abbreviations

SAU: Saarland University, D

RWT: RWTH Aachen University, D

TUD: Technische Universität Dresden, D

INR: Institut National de Recherche en Informatique et en Automatique, FR

IMP: Imperial College of Science, Technology and Medicine, UK

ULEIC: University of Leicester, UK

TUE: Technische Universiteit Eindhoven, NL

UNC: Universidad Nacional de Córdoba, AR

UBA: Universidad de Buenos Aires, AR

UNR: Universidad Nacional de Río Cuarto, AR

ITBA: Instituto Técnológico Buenos Aires, AR