electronics

MDPI

Article

A Randomized Response Framework to Achieve Differential Privacy in Medical Data

Andreas Ioannidis *, Antonios Litke and Nikolaos K. Papadakis

Infili Technologies SA, Kousidi 60, 157 72 Zografou, Greece; alitke@infili.com (A.L.); npapadakis@infili.com (N.K.P.)
* Correspondence: aioannidis@math.uoa.gr

Abstract: In recent years, differential privacy has gained substantial traction in the medical domain, where the need to balance privacy preservation with data utility is paramount. As medical data increasingly relies on cloud platforms and distributed sharing among multiple stakeholders, such as healthcare providers, researchers, and policymakers, the importance of privacy-preserving techniques has become more pronounced. Trends in the field focus on designing efficient algorithms tailored to high-dimensional medical datasets, incorporating privacy guarantees into federated learning for distributed medical devices, and addressing challenges posed by adversarial attacks. Our work lays a foundation for these emerging applications by emphasizing the role of randomized response within the broader differential privacy framework, paving the way for advancements in secure medical data sharing and analysis. In this paper, we analyze the classical concept of a randomized response and investigate how it relates to the fundamental concept of differential privacy. Our approach is both mathematical and algorithmic in nature, and our purpose is twofold. On the one hand, we provide a formal and precise definition of differential privacy within a natural and convenient probabilistic—statistical framework. On the other hand, we position a randomized response as a special yet significant instance of differential privacy, demonstrating its utility in preserving individual privacy in sensitive data scenarios. To substantiate our findings, we include key theoretical proofs and provide indicative simulations, accompanied by open-access code to facilitate reproducibility and further exploration.

Keywords: privacy and security; differential privacy; randomized response; probabilistic model

check for updates

Academic Editor: Cheonshik Kim

Received: 4 December 2024
Revised: 10 January 2025
Accepted: 13 January 2025
Published: 15 January 2025

Citation: Ioannidis, A.; Litke, A.; Papadakis, N. A Randomized Response Framework to Achieve Differential Privacy in Medical Data. *Electronics* **2025**, *14*, 326. https://doi.org/10.3390/electronics14020326

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Randomized response (RR) is an old technique for survey purposes, proposed in [1] and reconsidered in various forms thereafter, e.g., see [2–4] and the references therein. Its purpose is to provide a virtual protection for individuals in the following sense. There is often an "annoying" or "sensitive" question in a survey where a participant would prefer not be asked; this kind of question may concern the sexual orientation, a criminal activity, a substance's illegal use or health issues. In such a case, it is possible that some will elect to answer dishonestly or not answer at all. The framework of RR "shuffles" the answers and cancels potential information leakage, while maintaining the general statistical structure of the sample. As we will see, the classical RR framework is essentially an application of probability theory and is suitably modeled with Bernoulli random variables.

On the other hand, differential privacy (DP) is a relatively new concept, appeared for the first time in [5] and concerns the general and wide problem of data privacy in

computer science. The main idea behind the very definition of DP is that one cannot extract information by slightly changing the data. It is generally recognized as a standard concept in data privacy and has been a field of active research, e.g., see [6–8] and the references therein. Probabilities and random elements, especially random variables and their distributions, are also involved via the core concept of the randomized algorithm. Here, we refine the model by using the standard terminology of mathematical statistics based on a similar approach as in [9,10].

The interest for DP has recently reignited thanks to the hot topic of medical data security, e.g., [11]. DP can act as a critical cybersecurity measure within the medical domain, addressing the unique challenges of protecting sensitive patient data. As healthcare increasingly adopts digital platforms and cloud-based solutions for sharing data, preserving patient privacy becomes a primary concern. DP techniques offer a solution by allowing data to be analyzed while ensuring that individual patient information remains confidential. This approach not only upholds regulatory compliance but also facilitates secure data sharing among healthcare stakeholders, fostering innovation in medical research and practice.

This approach is followed in the SEPTON project (https://septon-project.eu/ (accessed on 12 January 2025)), which addresses the deficiency in IT network infrastructure for healthcare by introducing a comprehensive cybersecurity toolkit. This toolkit serves two critical purposes: securing networked medical devices, including wearables and implants, and ensuring safe, privacy-preserving data exchanges. To achieve the latter, SEPTON places a strong emphasis on preserving individual privacy through the utilization of DP techniques. This ensures that exchanged data remain secure and anonymous, a crucial aspect in healthcare data transactions, while, at the same time, the statistical analysis of this privacy, meaning that enhanced data can be available to researchers. The integration of these advanced methodologies fortifies both the security of medical devices and the confidentiality of patient data, responding to the evolving threat landscape in healthcare IT.

Our interest to revisit RR as a natural and straightforward implementation of DP is recently made in [12]. The purpose of the present paper is twofold:

- To further build the proper mathematical foundations for DP, in general, and RR, in particular. More precisely, we provide a conclusive definition for the concepts of randomization, randomized algorithm and randomized response in the natural context of probability theory.
- To expand the results found there so that they will cover every random variable with finite support.

This work develops the theoretical foundation of differential privacy (DP) within a probabilistic framework, emphasizing the relationship between randomized mechanisms and privacy guarantees. It introduces random elements and their randomization in a probabilistic context, forming the basis for a rigorous definition of DP and a comprehensive statistical framework for analysis. A comparison of local and global privacy supports the introduction of randomized response (RR), a key privacy-preserving mechanism. The study explores RR formulations, starting with the binary case based on Bernoulli trials, extending to variables with values in $0, 1, ..., \kappa - 1$, and generalizing to all finitely supported random variables. The generalized RR mechanism effectively handles discrete variables with finite support, preserving data integrity while adding calibrated noise to ensure privacy. This makes RR a versatile and practical tool for privacy-sensitive data collection in applications like surveys, behavioral studies, and medical research.

This paper is organized as follows. In Section 2, we present the concepts of a random element and a corresponding randomization in a purely probabilistic setting. Section 3 is a central one, since it establishes the definition of DP by using the aforementioned concepts

and setting up the statistical framework for our investigation. In Section 4, we make a comparative analysis between local and global privacy and introduce the concept of RR. The last three sections are devoted to RR. More precisely, Section 5 describes the classical RR with Bernoulli trials, i.e., binary (values 0 and 1) random variables. In Section 6, the results are generalized for random variables, assuming the values $0, 1, \ldots, \kappa - 1$. Section 7 covers all finitely supported random variables. The three appendices contain the corresponding Python code snippets, which implement the three versions of RR, where the previous is a special case of the next one.

2. Random Elements and Their Randomization

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space which describes a random experiment. Also let $(M, \mathcal{B})$ be a measurable space, where M is a metric space and $\mathcal{B} = \mathcal{B}(M)$ is the corresponding Borel σ-algebra. Consider a function $X : \Omega \to M$.

Definition 1. *X is called (an M-valued) random element (on $(\Omega, \mathcal{F}, \mathbb{P})$) if it is a measurable function, i.e.,*

$$(X \in B) := X^{-1}(B) \in \mathcal{F}$$

for every $B \in \mathcal{B}$.

The random element X induces a Borel probability measure on $(M, \mathcal{B})$ defined by

$$\mathbb{D}(X, B) := \mathbb{P}(X \in B) \, , B \in \mathcal{B}$$

Definition 2. *$\mathbb{D}(X, \cdot)$ is called the distribution of X. The smallest closed set $S(X) \in \mathcal{B}$ with the property $\mathbb{D}(X, S(X)) = 1$ is called the support of X.*

Remark 1. *The knowledge of distribution is important since it makes the underlying probability space irrelevant; the important things here are the values of the random element.*

We follow the general practice to represent random elements with capital letters, e.g., X, Y, and ..., and their values with the corresponding lowercase letters, e.g., x, y, and .... When the outcome of the experiment is $\omega \in \Omega$, the corresponding value of the random element X is, e.g., $X(\omega) = x$. Most often, we suppress the outcome by simply writing $X = x$, and x is then called a realization of X. The support, $S(X)$, of the random element X contains the possible realizations that X can admit; however, in general, it is not the same as the image of the X function.

Now consider an M-valued random element X on $(\Omega, \mathcal{F}, \mathbb{P})$, another probability space $(\Gamma, \mathcal{G}, \mathbb{Q})$ and an N-valued random element Y on the product probability space $(\Omega \times \Gamma, \mathcal{F} \otimes \mathcal{G}, \mathbb{P} \otimes \mathbb{Q})$.

Definition 3. *Y is called a (N-valued) randomization for X (on $(\Gamma, \mathcal{G}, \mathbb{Q})$) if for every fixed $\gamma \in \Gamma$, $Y(\cdot, \gamma)$ is $\sigma(X)$-measurable.*

Remark 2. *In the above definition, $\sigma(X)$ stands for the smallest sub-σ-algebra of $\mathcal{F}$, which makes X a random element. Formally, $\sigma(X) := \{(X \in B) : B \in \mathcal{B}\}$.*

Under some technical assumptions, the Doob–Dynkin lemma applies, e.g., see Lemma 1.13 [13], and a randomization for X depends on $\omega \in \Omega$ only via X.

Proposition 1. *Let Y be a randomization for X. Then, there exists a measurable function $\phi : M \times \Gamma \to N$ such that*

$$Y(\omega, \gamma) = \phi(X(\omega), \gamma) \tag{1}$$

The value of Y is determined by the outcomes of two independent random experiments: $\omega \in \Omega$ and $\gamma \in \Gamma$. Given the former outcome, the value $X(\omega) = x$ is known and we have actually the conditional random element $Y|X = x$ on $(\Gamma, \mathcal{G}, \mathbb{Q})$. We will denote this random element by $Y[x]$.

Definition 4. *Let Y be a randomization for X. We say that Y is uniformly supported if the support of $Y[x]$ is independent from $x \in S(X)$, or if $S(Y[x]) = S(Y)$. If, in addition, $N = M$ and $S(Y) = S(X)$, then Y is said to be a proper randomization.*

In other words, a proper randomization keeps the general outlook of X and randomly substitute a value with another value.

Proposition 2. *If Y is an N-valued randomization for X, $f : N \to N'$ is a measurable function and $Z = f(Y)$, then Z is an N'-valued randomization for X.*

Proof. Apparently, $Z : \Omega \times \Gamma \to N'$. Fix $\gamma \in \Gamma$ and let $B' \in \mathcal{B}(N')$. Then,

$$(Z(\cdot, \gamma) \in B') = (f(Y) \in B') = (Y \in f^{-1}(B'))$$

However, $f^{-1}(B') = (f \in B') \in \mathcal{B}(N)$, so $(Y \in f^{-1}(B')) \in \sigma(X)$ and therefore

$$(Z(\cdot, \gamma) \in B') \in \sigma(X)$$

□

2.1. A Convenient Construction

An efficient way to obtain a randomization, which resembles the representation (1), is the following: consider an M'-valued random element R on $(\Gamma, \mathcal{G}, \mathbb{Q})$ and a measurable function $g : M \times M' \to N$. Then, the equation $Y = g(X, R)$ defines a randomization for X. This representation has the advantage that makes no use of the underlying probability spaces and relies only on random elements. The random element R plays the role of "noise" applied on the values of X. Note that in this case, $Y[x] = g(x, R)$.

2.2. Random Variables

An $\mathbb{R}$-valued random element, X, is called a random variable. In this case, the distribution of X is fully determined by the cumulative distribution function (cdf) of X, $F_X : \mathbb{R} \to [0, 1]$, defined by

$$F_X(x) := \mathbb{P}(X \leqslant x)$$

Actually, $\mathbb{D}(X, \cdot)$ is then a Lebesgue–Stieltjes measure and

$$\mathbb{D}(X, B) = \int_B dF_X(x)$$

The support of X is given by

$$S_X = \{x \in \mathbb{R} : F_X(x + \delta) - F_X(x - \delta) > 0 \quad \forall \delta > 0\}$$

The mean value of X is defined as

$$\mathbb{E}(X) := \int_{-\infty}^{+\infty} x \, dF_X(x)$$

where the integrals are understood in the Lebesgue–Stieltjes sense. More generally, if g is a measurable real function,

$$\mathbb{E}(g(X)) := \int_{-\infty}^{+\infty} g(x) \, dF_X(x)$$

The variance of X is

$$\mathrm{Var}(X) := \int_{-\infty}^{+\infty} (x - \mathbb{E}(X)) \, dF_X(x) = \mathbb{E}\left(X^2\right) - (\mathbb{E}(X))^2$$

The distribution function is characterized by three properties, namely the right continuous, increasing, and $\lim_{x\to-\infty} F_X(x) = 0$, $\lim_{x\to+\infty} F_X(x) = 1$. Every function $\mathbb{R} \to [0,1]$ enjoying these three properties is the distribution function of some random variable. There are two kinds of distribution functions and consequent kinds of random variables that arise in applications:

- F_X is a step function. Then, $S(X)$ is a countable set (finite or infinite, with no accumulation points in the latter case), which consists of the jumps of F_X. Moreover,

  $$dF_X = \sum_{x_i \in S(X)} f_X(x_i) \delta_{x_i}$$

  where δ_x stands for the Dirac measure concentrated at x and the function f_X is defined as $f_X(x) = \mathbb{P}(X = x) = \mathbb{D}(X, \{x\})$. In this case, we say that X is a discrete random variable and f_X is called its probability mass function (pmf).
- F_X is absolutely continuous, e.g., see ([14], Section 33) for relevant definitions and results. Then, F_X is an almost everywhere differentiable function and let $F'_X = f_X$, so

  $$dF_X(x) = f_X(x)dx$$

  The support, $S(X)$, is an interval of $\mathbb{R}$, represented as $S(X) = \overline{\{x \in \mathbb{R} : 0 < F_X(x) < 1\}}$, where the overline stands for the topological closure of the set. In this case, X is said to be continuous and f_X is called its probability distribution function (pdf).

The existence of f_X simplifies the analysis in both cases. Indeed, there is a simple representation for the support

$$S(X) = \overline{\{x \in \mathbb{R} : f_X(x) > 0\}}$$

In order to have a common treatment, we will refer to f_X as the pdf for both discrete and continuous random variables. If there is no danger of confusion, we may omit subscripts.

2.3. Random Tuples and Vectors

An M^n-valued random element, $\boldsymbol{X}$, is called an M-valued random tuple. Evidently, $\boldsymbol{X}$ has a representation as $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and it turns out that $\boldsymbol{X}$ is a random tuple if and only if each coordinate, X_i, is an M-valued random element. An $\mathbb{R}$-valued random tuple is called a random vector. In other words, $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is a random vector if and only if X_i is a random variable. The distribution of $\boldsymbol{X}$ is often referred as the joint distribution o X_1, X_2,…,X_n.

Of very special interest is the case where X_i variables are independent and identically distributed (abbreviated as iid), with the latter meaning that they have the same distribution. In this case, $\boldsymbol{X}$ is called an M-valued random sample (from the random element X). An $\mathbb{R}$-random sample is simply called a random sample.

If $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is a random sample from the random variable X and X has a pdf, $f_X(x)$, we sometimes say that $\boldsymbol{X}$ is a random sample from the distribution $f_X(x)$. The joint pdf is then

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{n} f_X(x_i)$$

for every realization $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ of $\boldsymbol{X}$.

3. Differential Privacy

In this section, we introduce the terminology, the notation and the general mathematical structure for the central problem of data privacy. Our intention is to establish a mathematical model for privacy, especially differential privacy. We start with some basic definitions. Let $\boldsymbol{X}$ be an M-valued random sample, $\boldsymbol{X} = (X_1, X_2, ..., X_n)$, from the random element X.

- A *database* is a realization $\boldsymbol{x} = (x_1, x_2, ..., x_n) \in S(X)^n$ of $\boldsymbol{X}$.
- The *curator* is a trustful entity who acts as an operator for one or more databases.
- An *adverse* is a harmful entity who wants to extract sensitive information from databases.
- An *analyst* wants to apply statistical methods in databases and make inferences.
- A *query* is a function having the databases as its domain.

The procedure is as follows: An analyst asks queries to the curator, but it is generally unknown if this analyst is actually adverse or not. As said, an "innocent" analyst will use queries for statistical inference but an adverse uses queries for devious purposes. Thus, a simple but rather descriptive definition for the subject of data privacy could be that the curator must *allow the statistical analysis without compromising any of the values of* x_i*'s*.

The concept of DP works exactly in this direction. Some of the most used attacks ask queries for similar databases and recover information by comparing them. For a comparison to take place, we need a kind of metric.

Definition 5. *The Hamming distance for* $\boldsymbol{x}, \boldsymbol{x}' \in S(X)^n$ *is defined as the number of coordinates where* $\boldsymbol{x}, \boldsymbol{x}'$ *differ,*

$$h(\boldsymbol{x}, \boldsymbol{x}') = \#\{i \in \{1, 2, \ldots, n\} \,:\, x_i \neq x_i'\}$$

Two databases $\boldsymbol{x}, \boldsymbol{x}'$ *are called neighboring if* $h(\boldsymbol{x}, \boldsymbol{x}') \leqslant 1$.

As said, our intention here is to formalize DP, as defined, e.g., in [15], in the standard probabilistic/statistical nomenclature. One of the key notions is that of randomized algorithm, as is referred in Definition 2.2 [15], or data release mechanism, as referred to in [10]. The key object here is an N-valued tuple randomization, $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_m)$, for the aforementioned random sample $\boldsymbol{X}$.

Definition 6. *The random element* $\boldsymbol{Y}[\cdot]$*, resulting after we have the database at hand, is called a randomized algorithm.*

We are now in a position to give the formal definition of DP.

Definition 7. *Let* $\varepsilon > 0$*. The randomized algorithm,* $\boldsymbol{Y}[\cdot]$*, is called* ε*-differentially private if*

$$\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}], B) \leqslant e^{\varepsilon} \mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}'], B) \tag{2}$$

for every neighboring databases $\boldsymbol{x}, \boldsymbol{x}'$ *and Borel set* $B \in \mathcal{B}(N)$.

The inequality (2) essentially involves the distribution of the conditional N-valued random tuples $\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}$, $\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}'$. The bottom line is that when ε is small and $\boldsymbol{x}, \boldsymbol{x}'$ do not differ much, the corresponding distributions do not differ much as well. In other words, the adverse cannot obtain much information by slightly changing the database.

The definition of DP deserves a further discussion. In what follows, $\boldsymbol{Y}[\cdot]$ is an ε-DP randomized algorithm.

3.1. The Role of Support

By interchanging the role of $(\boldsymbol{x}, \boldsymbol{x}'$ in (2)), we obtain the following result.

Lemma 1. *For every neighboring databases* $\boldsymbol{x}, \boldsymbol{x}'$ *and Borel set* B,

$$e^{-\varepsilon}\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}'], B) \leqslant \mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}], B) \leqslant e^{\varepsilon}\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}'], B)$$

Therefore,

$$\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}], B) = 0 \quad \Leftrightarrow \quad \mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}'], B) = 0$$

Corollary 1. *The randomization* $\boldsymbol{Y}[\boldsymbol{x}]$ *is uniformly supported. More precisely,*

$$S(\boldsymbol{Y}[\boldsymbol{x}]) = S(\boldsymbol{Y}) \tag{3}$$

Equation (3) imposes an important necessary condition for a randomization to give differentialy private randomized algorithms. This stability of the support for $\boldsymbol{Y}[\cdot]$ is a very desirable consequence of Definition 7: a randomization algorithm whose values depend on the observed database would be useless, since information leakage could be present using this method.

3.2. The Role of ε

The usual usage of the letter ε in mathematical analysis is to represent a small positive constant. In the context of Definition 2, ε is called the privacy budget and only positivity is assumed. The privacy budget is a key parameter that quantifies the trade-off between privacy and accuracy in data analysis. A lower value of the privacy budget ε signifies stronger privacy guarantees, as it limits the amount of information an adversary can infer about an individual's private data from the output of a computation. In contrast, a larger ε allows more precise results but offers weaker privacy protection. The privacy budget controls the degree of randomness added to the data or the response to the query, ensuring that the inclusion or exclusion of any individual in the dataset has a limited impact on the output.

Selecting an appropriate ε is critical and often depends on the sensitivity of the data, the use case, and the acceptable balance between utility and privacy. However, it is evident that the smaller the ε, the closer the distributions are, and thus the stronger the privacy. To our knowledge, there is no common census of how "small" ε must be. This depends heavily on the applications where differential privacy is going to be applied. Actually, as presented later in Section 4.1, where RR algorithms are introduced, ε is a characteristic quantity of the algorithm. It is specified by the changing probabilities, so it is controllable only via them.

3.3. Procession

As said above, the role of the curator is to answer queries, and the role of analyst is to apply statistical methods. For both to work under ε-DP, they may use $\boldsymbol{Y}[\boldsymbol{x}]$ for analysis

instead $\boldsymbol{x}$ itself. The consequent procession of $\boldsymbol{Y}[\boldsymbol{x}]$ does not affect DP; this is a direct application of Proposition 2, which we reconsider below in the appropriate context.

Proposition 3. *Let the N'-valued random tuple $\boldsymbol{Z}$ be a function of $\boldsymbol{Y}$. Then, $\boldsymbol{Z}$ is an N'-valued randomization for $\boldsymbol{X}$ and $\boldsymbol{Z}[\cdot]$ is again an ε-DP randomized algorithm.*

3.4. Proximity of Distributions

The definition of DP involves the logarithmic quotient

$$\ln \frac{\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}], B)}{\mathbb{D}(\boldsymbol{Y}[\boldsymbol{x}'], B)} \tag{4}$$

and demands this to lie between $-\varepsilon$ and ε (see Lemma 1). This quotient relates with the Kullback–Leibler (KL) divergence [16], which is a statistical distance for probability distributions. To be more precise, the definition assumes a worst-case bound for the KL distance that ensures the proximity of $\boldsymbol{Y}[\boldsymbol{x}]$, $\boldsymbol{Y}[\boldsymbol{x}']$ for neighboring databases $\boldsymbol{x}$, $\boldsymbol{x}'$.

3.5. Our Framework

In order to keep things manageable and quite general, we will consider the classical estimation problem. In the sequel, $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ will always be a random sample from a distribution with pdf $f_X(\cdot;\theta)$, where θ is an unknown parameter which takes values in the so-called parametric space $\Theta \subseteq \mathbb{R}$. The joint pdf is then

$$f_{\boldsymbol{X}}(\boldsymbol{x};\theta) = \prod_{i=1}^{n} f_X(x_i;\theta)$$

If we surpass the realization of the sample, $\boldsymbol{x}$, and consider this only as a function of $\theta \in \Theta$, we speak about the likelihood function, denoted by $L(\theta)$. Sometimes, it is more convenient to use the logarithmic likelihood function, $\ell(\theta) := \ln L(\theta)$.

The intention is to construct an estimator for θ, i.e., a statistical function (simply, a statistic) with values on Θ. The term statistic is used for a function, T, of the random sample, $\boldsymbol{X}$, which does not depend on θ. Therefore, $T = T(\boldsymbol{X})$. There are many criteria for the goodness of an estimator, e.g., see Chapter 12 [17] or Chapter 7 [18], but the two most important notions are the following:

- T is called an unbiased estimator of θ if $\mathbb{E}(T) = \theta$. The unbiased estimator which attains the minimum variance among all such estimators is called the MVUE (minimum variance unbiased estimator). The most convenient way to construct the MVUE is by the aim of the Lehmann-Sheffé Theorem: an unbiased estimator which is a function of a complete and sufficient statistic is the MVUE.
- T is called an MLE (maximum likelihood estimator) if $\hat{\theta} := T(\boldsymbol{x})$ maximizes the likelihood function. This means that $\hat{\theta}$ satisfies the necessary condition $L'(\hat{\theta}) = 0$ and may satisfy the sufficient condition $L''(\hat{\theta}) < 0$.

  The quality of the estimation is checked via the mean square error, defined as

$$\mathrm{MSE}(T) := \mathbb{E}\Big((T-\theta)^2\Big)$$

It turns out that we have the representation

$$\mathrm{MSE}(T) = \mathrm{Var}(T) + b^2(T)$$

where $b(T) := \mathbb{E}(T) - \theta$ is the bias of T. For unbiased estimators, the MSE coincides with the variance, so the MVUE minimizes the MSE as well.

The construction of the estimator, in any case, has to be completed under DP. We will assume that we have randomized algorithms $\boldsymbol{Y}[\cdot]$ where the random vector $\boldsymbol{Y}[\boldsymbol{x}]$, which is the conditional random vector $\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}$. Moreover, it is described by a conditional pdf, $f(\boldsymbol{y}|\boldsymbol{x})$. Then, there is a substantial simplification since, instead of using the cumbersome quotient (4), it suffices to check the boundness of

$$\ln \frac{f(\boldsymbol{y}|\boldsymbol{x})}{f(\boldsymbol{y}|\boldsymbol{x}')}$$

Observe, incidentally, that DP must work only with $f(\boldsymbol{y}|\boldsymbol{x})$. The distribution of X is irrelevant and enters into the discussion only due to the estimation problem of θ. To conclude, the crucial task here is to describe the randomized algorithm $\boldsymbol{Y}[\cdot]$, i.e., to define $f(\boldsymbol{y}|\boldsymbol{x})$ in an efficient way. Since $\boldsymbol{Y}[\boldsymbol{x}]$ emerges after $\boldsymbol{X}$ is observed, $f(\boldsymbol{y}|\boldsymbol{x})$ should not depend on θ.

4. Local vs. Global Privacy

Definition 7 is quite general to include different aspects of DP. There are various discrimination and categorization attributes for privacy, in general, and DP, in particular; see Chapter 2 [7] for a thorough and comprehensive discussion. Here, we focus on the most fundamental aspect of the matter, i.e., how and where the noise is applied. There are two "schools of thought" in this direction. Roughly speaking, one can add noise separately to each datum and then calculate the estimator (local privacy) or first calculate the estimator from the original data and then apply the noise (global privacy). In the sequel, $\boldsymbol{X}=(X_1, X_2, \ldots, X_n)$ is a random sample from the random variable X, which has a pdf $f_X(x;\theta)$, and $T=T(\boldsymbol{X})$ is an estimator of θ.

4.1. Local Privacy

Let Y be a randomization for the random variable X. Then, $Y[x]$, $x \in S(X)$, is the conditional random variable $Y|X=x$. Then, let $f(y|x)$ be its pdf. Note that $f(y|x)$ is interpreted as the changing probability from the value x to the value y. For each random variable, X_i, in the sample, we consider a version Y_i of this randomization. With these at hand, the random vector $\boldsymbol{Y}=(Y_1, Y_2, \ldots, Y_n)$ is again a random sample and provides a randomization for $\boldsymbol{X}$. Indeed, given the database $\boldsymbol{x}$,

$$\boldsymbol{Y}[\boldsymbol{x}]=(Y_1[x_1], Y_2[x_2], ..., Y_n[x_n]) \tag{5}$$

is a random sample as well, and it has pdf

$$f(\boldsymbol{y}|\boldsymbol{x})=\prod_{i=1}^{n} f(y_i|x_i)$$

Remark 3. *By y_i, we denote the realization of the random variable $Y[x_i]$.*

Now take two neighboring databases $\boldsymbol{x}, \boldsymbol{x}'$, which differ in the j-th entry. Then, obviously,

$$\frac{f(\boldsymbol{y}|\boldsymbol{x})}{f(\boldsymbol{y}|\boldsymbol{x}')}=\frac{f(y_j|x_j)}{f(y_j|x_j')}$$

As a consequence, it suffices to check only simple random variables. Thus, we obtain the following important result:

Proposition 4. *The randomized algorithm defined in* (5) *is ε-DP if and only if it is uniformly supported and*

$$\ln \frac{f(y|x)}{f(y|x')} \leqslant \varepsilon$$

for every $y \in S(Y)$, $x, x' \in S(X)$, $x \neq x'$.

Definition 8. *If the randomized algorithm* (5) *originates from a proper randomization, then it is called a randomized response (RR).*

Corollary 2. *If X has finite support and* (5) *is an RR, then it is ε-DP, with*

$$\varepsilon = \max \ln \frac{f(y|x)}{f(y|x')}$$

where the maximum is taken among all $x, x', y \in S(X)$, $x \neq x'$.

The analyst has access not to the original random sample $\boldsymbol{X}$ but to $\boldsymbol{Y}$, on which any analysis can be performed. The naive approach, to consider $T(\boldsymbol{Y})$ as the estimator for θ, will generally not work efficiently; to obtain the desired estimator, further transformations may be needed. As a recap, all the steps of the procedure are summarized in Figure 1.

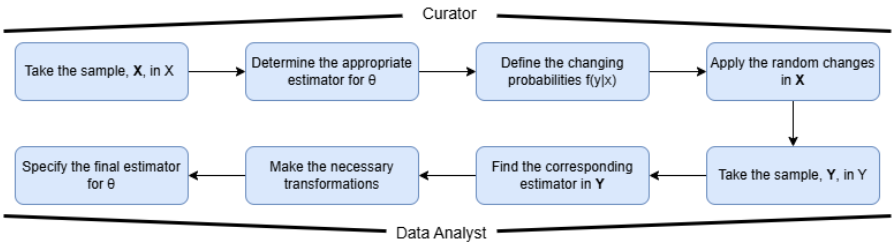


Figure 1. Flow diagram of the RR framework.

4.2. Global Privacy

In this approach, randomization is carried out directly on the estimator, which is on the random variable $T = T(\boldsymbol{X})$. The usual and simpler methodology to achieve global privacy is by using a "mechanism", i.e., to add noise into the outcome. More precisely, consider the randomization $Y = T(\boldsymbol{X}) + R$, where R is a random variable independent from X. A choice that seems to be natural ([15], Section 3.3) is to take $R \sim \text{Laplace}\left(\frac{\ell_T}{\varepsilon}\right)$. The parameter ℓ_T is defined as

$$\ell_T := \sup_{\substack{\boldsymbol{x},\boldsymbol{x}' \in S(X)^n \\ h(\boldsymbol{x},\boldsymbol{x}')=1}} |T(\boldsymbol{x}) - T(\boldsymbol{x}')|$$

provided this is a finite number, which applies when X has a finite support. It also applies when T is a Lipschitz function, namely when there exists $M \geqslant 0$ such that

$$|T(\boldsymbol{x}) - T(\boldsymbol{x}')| \leqslant M\, h(\boldsymbol{x}, \boldsymbol{x}')$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in S(X)^n$. Under these assumptions, $Y[\cdot]$ is an ε-DP randomized algorithm.

In this paper, we will pursue only the local version of privacy, especially the concept of RR, and we will not touch the global one. The main reason for this is because the present paper is devoted to RR. Another reason is that the global version of privacy is extensively studied by Cynthia Dwork and her pioneering team (see the relevant references [5,6,15]).

5. RR for Bernoulli Trials

In the present section, we describe the original version of RR as presented in [1]. Suppose that there is a survey where participants are asked if they have ever done a "bad" thing and a person can either have "done" or "not done" the thing. To assure some guarantee to the participants that their answer is safe, the interviewee uses a spinner wheel that is not visible to the interviewer. This wheel has two regions, one marked with "done" and the other marked with "not done" (see Figure 2). The interviewee replies "yes" to the question if the spinner outcome matches their action and "no" otherwise. This way, even if someone somehow figures out the answer of a specific person, they cannot tell for sure if the person has done the thing or not.

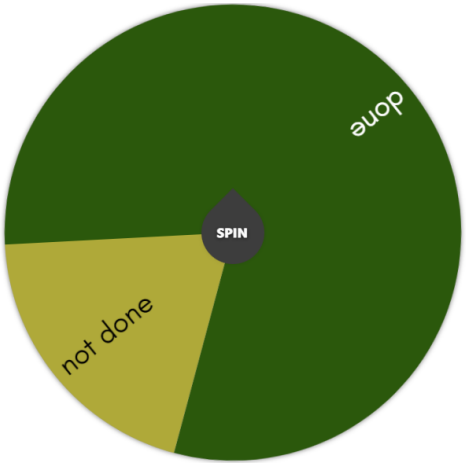


Figure 2. The spinner wheel.

Now, let X be the binary random variable that indicates the action: $X = 1$ for "done" and $X = 0$ for "not done". Then, $X \sim \text{Bernoulli}(p)$, where the true probability, p, is to be estimated. Also let Y be the binary random variable that indicates the answer: $Y = 1$ for "yes" and $Y = 0$ for "no". In order to make things precise and fit the procedure into our framework, let α be the probability for the spinner to show "not done", so α is the percentage of the area of the corresponding region in the whole circle. The conditional pdf $f(y|x)$ is presented in Table 1. As it is directly seen in this table, α has an interesting interpretation: it expresses the probability that the interviewee answers dishonestly or that they switch their answer. In other words, instead of "done" and "not done", the spinner could have the commands "stay" (truth) and "switch" (lie).

Table 1. Conditional probabilities $f(y|x) = \mathbb{P}(Y = y|X = x)$.

$y \backslash x$	**0**	**1**
0	$1 - \alpha$	α
1	α	$1 - \alpha$

Let R be the binary random variable that indicates the spinner outcome, $R = 1$ for "switch" and $R = 0$ for "stay". Then, $R \sim \text{Bernoulli}(\alpha)$, and we apparently have the representation

$$Y = (X + R) \mod 2 \tag{6}$$

It turns out that a much more convenient formula applies,

$$Y = X + R - 2XR \tag{7}$$

Proposition 5. *As defined in (7), Y is an RR for X, with $Y[0] = R$, $Y[1] = 1 - R$. Additionally, it provides an ε-DP randomized algorithm with*

$$\varepsilon = \max\left\{\ln\frac{\alpha}{1-\alpha}, \ln\frac{1-\alpha}{\alpha}\right\}$$

Proof. We have $S(X) = S(Y) = \{0, 1\}$, so Y is a proper randomization and thus an RR. Moreover, X has finite support, so $Y[\cdot]$ is ε-DP, with

$$\varepsilon = \max\left\{\ln\frac{f(0|0)}{f(0|1)}, \ln\frac{f(0|1)}{f(0|0)}, \ln\frac{f(1|0)}{f(1|1)}, \ln\frac{f(1|1)}{f(1|0)}\right\}$$

thanks to Corollary 2. Now,

$$\frac{f(0|0)}{f(0|1)} = \frac{f(1|1)}{f(1|0)} = \frac{1-\alpha}{\alpha}$$

and, similarly,

$$\frac{f(0|1)}{f(0|0)} = \frac{f(1|0)}{f(1|1)} = \frac{\alpha}{1-\alpha}$$

□

Remark 4. *The closer α is to $\frac{1}{2}$, the smaller the ε. For the standard RR, which corresponds to $\alpha = \frac{1}{4}$, we find $\varepsilon = \ln 3$.*

We have that $\mathbb{E}(X) = p$, $\mathbb{E}(Y) = \alpha$ and since X, R are independent,

$$\mathbb{E}(Y) = p + \alpha - 2p\alpha = (1 - 2\alpha)p + \alpha = p + (1 - 2p)\alpha \tag{8}$$

Therefore, $Y \sim \text{Bernoulli}(p + \alpha - 2p\alpha)$. We also have

$$\text{Var}(Y) = [p + (1 - 2p)\alpha][1 - p - (1 - 2p)\alpha] = p(1 - p) + (1 - 2p)^2\alpha(1 - \alpha)$$

Observe that we can write $\mathbb{E}(Y) = (1 - 2\alpha)\mathbb{E}(X) + \alpha$; thus,

$$p = \frac{\mathbb{E}(Y) - \alpha}{1 - 2\alpha} \tag{9}$$

Additionally, by using $\text{Var}(X) = p(1 - p)$, $\text{Var}(R) = \alpha(1 - \alpha)$, we find

$$\text{Var}(Y) = \text{Var}(X) + (1 - 2p)^2\,\text{Var}(R) \tag{10}$$

Let us now start the discussion about the estimation of p. The random sample $\boldsymbol{X} = (X_1, X_2, ..., X_n)$ gives the sample mean, $\overline{X}$, as the MVUE and the MLE. In order to create the corresponding sample of the randomized responses, we can just sample R as $\boldsymbol{R} = (R_1, R_2, ..., R_n)$ and reproduce $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{R} - 2\boldsymbol{X} \odot \boldsymbol{R} \tag{11}$$

where $\odot$ denotes the Hadamard (entry-wise) product. Thanks to (8),

$$\mathbb{E}\left(\overline{Y}\right) = p + (1 - 2p)\alpha;$$

therefore, $\overline{Y}$ is a biased estimator of p with bias $b(\overline{Y}) = (1-2p)\alpha$. Moreover, (10) gives

$$\mathrm{Var}(\overline{Y}) = \mathrm{Var}(\overline{X}) + (1-2p)^2\,\mathrm{Var}(\overline{R}) = \mathrm{MSE}(\overline{X}) + (1-2p)^2\,\mathrm{Var}(\overline{R})$$

and

$$\mathrm{MSE}(\overline{Y}) = \mathrm{Var}(\overline{Y}) + b^2(\overline{Y}) = \mathrm{MSE}(\overline{X}) + (1-2p)^2(\mathrm{Var}(\overline{R}) + \alpha^2)$$

When $\alpha \neq \frac{1}{2}$, the estimator $T = T(\boldsymbol{Y}) = T(\boldsymbol{X}, \boldsymbol{R})$ where

$$T = \frac{\overline{Y} - \alpha}{1-2\alpha} \tag{12}$$

is then unbiased with

$$\mathrm{MSE}(T) = \frac{1}{(1-2\alpha)^2}\,\mathrm{Var}(\overline{Y}) = \frac{1}{(1-2\alpha)^2}\,\mathrm{MSE}(\overline{X}) + \frac{(1-2p)^2}{(1-2\alpha)^2}\,\mathrm{Var}(\overline{R})$$

Substitute to find

$$\mathrm{MSE}(T) = \frac{p(1-p) + (1-2p)^2\alpha(1-\alpha)}{(1-2\alpha)^2 n} \tag{13}$$

It is worth to discuss the three extreme values of α:

(a) When $\alpha = 0$, i.e., there is no chance for switching. We have $R = 0$ and $Y = X$. Then, $T = \overline{Y} = \overline{X}$ and the statistical analysis is the usual one. In other words, for $a \to 0$, the randomization becomes negligible and we rely more and more on the honest answer of the participants.

(b) When $\alpha = 1$, i.e., switching always occurs. We have $R = 1$ and $Y = 1 - X$. Then, $\overline{Y} = 1 - \overline{X}$ and $T = \overline{X}$. That is, for $\alpha \to 1$, there is too much switching, which is useless as well.

(c) When $\alpha = \frac{1}{2}$, the switching becomes a coin toss. Thus, there is "pure" randomization. In this case, $Y \sim \mathrm{Bernoulli}(\frac{1}{2})$, so the distribution of Y has nothing to do with p, and every attempt for statistical analysis is meaningless. As representation (12) dictates, T is not even defined when $\alpha = \frac{1}{2}$ and more precisely, for $\alpha \to \frac{1}{2}$, T becomes infinite. In other words, too much randomization is not advisable. Nevertheless, for values of α are quite close to $\frac{1}{2}$. Thus, T is unbiased with a large MSE (see (13)). Therefore, we have the desired properties.

Formula (11) is very convenient to use and is easily implemented in an algorithmic setting. The corresponding code snippet in the Python language that implements the RR can be seen in Appendix A.1. As an application, we run a simulation for actual values of p in $(0, 1)$, we sample for X and R, and we study the behavior of $\overline{X}$ and T estimators for various values of the sample size, n, as well as the switching probability, α. The results are presented in Figure 3. As we can see there, T gives estimations that generally follow those of the sample mean. For large samples, they are practically indistinguishable. As expected, the convergence is faster for values of α closer to 0 (less switching) and slower for values closer to $\frac{1}{2}$ (more switching).

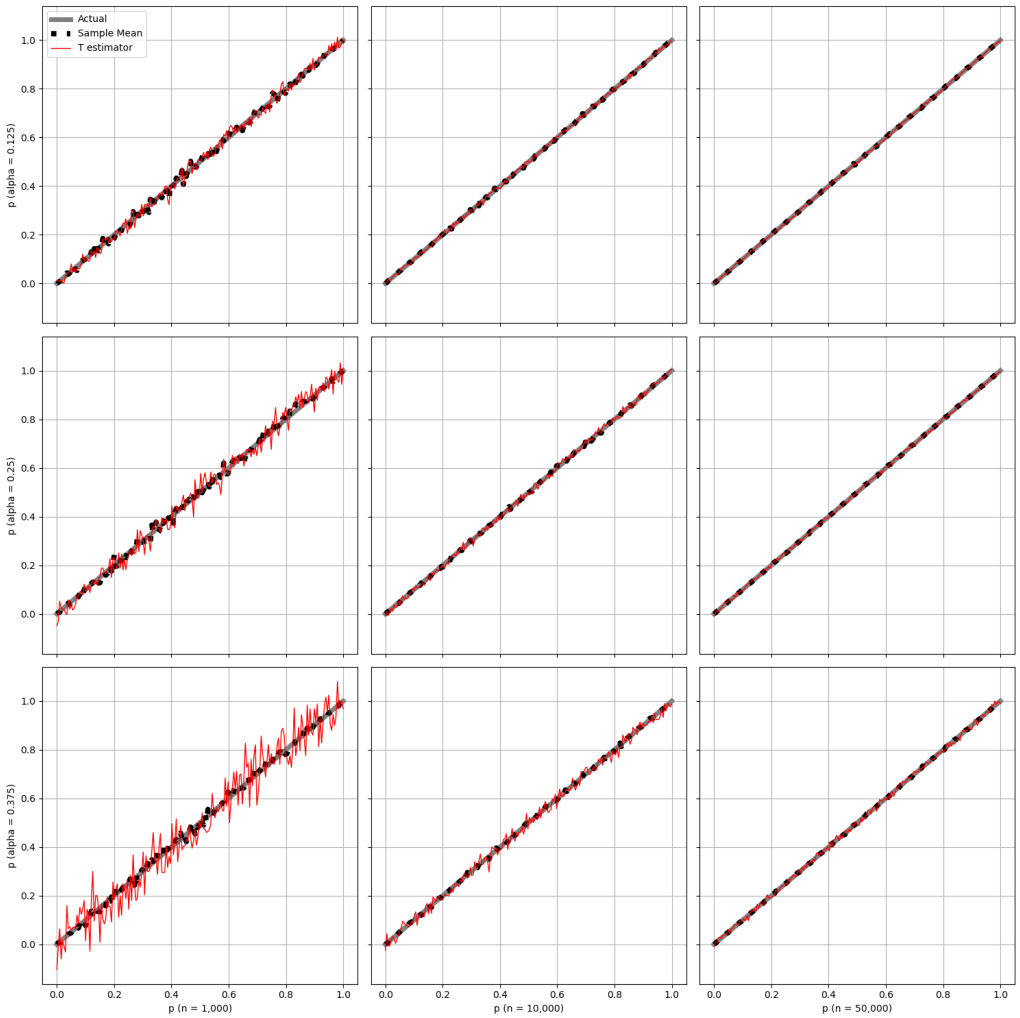


Figure 3. Simulation for various sizes of α and n.

6. RR for Random Variables with Finite Support

Now, we proceed to more general discrete variables with finite support. Let X be a random variable with κ possible values; initially, we assume that $S(X) = \{0, 1, \ldots, \kappa - 1\}$, and we will elaborate the more arbitrary values later. Since we want to generalize the concept of RR for binary random variables, we consider the random variable R with $S(R) = S(X)$ and define the randomization, in accordance with (6), as follows:

$$Y = (X + R) \mod \kappa = \begin{cases} X + R \text{ , if } X + R < \kappa \\ X + R - \kappa \text{ , if } X + R \geqslant \kappa \end{cases}$$

For $\kappa > 2$, it is difficult, if not possible at all, to find a representation similar to (7). However, the following one always applies:

$$Y = X + R - \kappa I_{(X+R \geqslant \kappa)}, \tag{14}$$

where we denote the indicator random variable of the event A by I_A.

The distribution of R is now chosen in such a way so the probability of changing a value is again α and is equally distributed among the values $1, 2, \ldots, \kappa - 1$, namely

$$f_R(r) := \begin{cases} 1 - \alpha \text{ , for } r = 0 \\ \dfrac{\alpha}{\kappa - 1} \text{ , for } r = 1, 2, \ldots, \kappa - 1 \end{cases}$$

Straightforward calculations show tha:

Lemma 2. *(a)* $\mathbb{E}(R) = \dfrac{\alpha\kappa}{2}$.

(b) $\mathrm{Var}(R) = \dfrac{\alpha\kappa(4\kappa - 2 - 3\alpha\kappa)}{12}$.

It is also evident that $Y[x] = (x + R) \mod \kappa$, which has pdf

$$f(y|x) = \begin{cases} 1 - \alpha \text{ , if } y = x, \\ \dfrac{\alpha}{\kappa - 1} \text{ , otherwise} \end{cases} \tag{15}$$

Proposition 6. *The randomization defined in* (14) *is an RR for X and provides an ε-DP randomized algorithm with*

$$\varepsilon = \max\left\{ \ln \frac{\alpha}{(\kappa - 1)(1 - \alpha)}, \ln \frac{(\kappa - 1)(1 - \alpha)}{\alpha} \right\}$$

Proof. The possible values for the ratio

$$\frac{f(y|x)}{f(y|x')}$$

are

$$1, \frac{\alpha}{(\kappa - 1)(1 - \alpha)}, \frac{(\kappa - 1)(1 - \alpha)}{\alpha}$$

□

As is already noted, we have not made any assumption on the distribution of X. The intention here is to estimate the mean value, $\mathbb{E}(X) = \mu$.

Proposition 7. *We have that*

$$\mathbb{E}(Y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right)\mathbb{E}(X) + \frac{\alpha\kappa}{2}$$

Therefore, for $\alpha \neq \frac{\kappa - 1}{\kappa}$,

$$\mu = \frac{\kappa - 1}{2} \frac{2\mathbb{E}(Y) - \alpha\kappa}{\kappa - 1 - \alpha\kappa} \tag{16}$$

Proof. It is direct from (14) and the fact that

$$\mathbb{E}\left(I_{(X+R \geqslant \kappa)}\right) = \mathbb{P}(X + R \geqslant \kappa) = \sum_{x=1}^{\kappa-1} \sum_{r=\kappa-x}^{\kappa-1} \mathbb{P}(X = x, R = r) =$$
$$= \sum_{x=1}^{\kappa-1} \sum_{r=\kappa-x}^{\kappa-1} f_X(x) f_R(r) = \frac{\alpha}{\kappa - 1} \sum_{x=1}^{\kappa-1} x f_X(x) = \frac{\alpha}{\kappa - 1}\mathbb{E}(X)$$

□

Remark 5. *Observe that the following representation follows:*

$$\mathbb{E}(Y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right)\mathbb{E}(X) + \mathbb{E}(R)$$

Actually, there is another approach to calculate the mean value and variance of Y by finding its distribution, as follows:

Proposition 8. *For* $y = 0, 1, \ldots, \kappa - 1$,

$$f_Y(y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right) f_X(y) + \frac{\alpha}{\kappa - 1} \tag{17}$$

Proof. By using the total probability formula,

$$f_Y(y) = \sum_{x=0}^{\kappa-1} f_X(x) f(y|x) = f_X(y) f(y|y) + \sum_{\substack{x=0 \\ x \neq y}}^{\kappa-1} f_X(x) f(y|x) =$$

$$= (1 - \alpha) f_X(y) + \frac{\alpha}{\kappa - 1} \sum_{\substack{x=0 \\ x \neq y}}^{\kappa-1} f_X(x) = (1 - \alpha) f_X(y) + \frac{\alpha}{\kappa - 1}(1 - f_X(y))$$

□

Remark 6. *We have that, for* $y > 0$,

$$f_Y(y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right) f_X(y) + f_R(y)$$

Proposition 9. *We have that*

$$\mathrm{Var}(Y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right)\left(\mathrm{Var}(X) + (\mathbb{E}(X))^2\right) + \frac{\kappa(2\kappa - 1)}{6}\alpha - (\mathbb{E}(Y))^2$$

Proof. Multiply (17) by y^2 and sum across $y = 0, 1, \ldots, \kappa - 1$,

$$\sum_{y=0}^{\kappa-1} y^2 f_Y(y) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right) \sum_{y=0}^{\kappa-1} y^2 f_X(y) + \frac{\alpha}{\kappa - 1} \sum_{y=0}^{\kappa-1} y^2$$

Therefore,

$$\mathbb{E}\left(Y^2\right) = \left(1 - \frac{\kappa}{\kappa - 1}\alpha\right)\mathbb{E}\left(X^2\right) + \frac{\alpha}{\kappa - 1}\frac{(\kappa - 1)\kappa(2\kappa - 1)}{6}$$

Now use the fact that $\mathbb{E}\left(Y^2\right) = \mathrm{Var}(Y) + (\mathbb{E}(Y))^2$ (and the analogous statement for $\mathbb{E}\left(X^2\right)$), and we are done. □

Along with the original random sample in X, by taking the random sample in R and the corresponding in Y, we consider

$$T = \frac{\kappa - 1}{2}\frac{2\overline{Y} - \alpha\kappa}{\kappa - 1 - \alpha\kappa} \tag{18}$$

which is an unbiased estimator of μ, thanks to (16). The corresponding code snippet for the implementation of this version of the RR can be found in Appendix A.2. In accordance with the Bernoulli case, we run a simulation for the binomial distribution. Therefore, $X \sim \text{Binomial}(\kappa - 1, p)$ and $\mathbb{E}(X) = \mu = (\kappa - 1)p$. Now we let κ vary,and the switching

probability is set to the half-way point of the "prohibited value", i.e., to $\alpha = \frac{\kappa-1}{2\kappa}$. The results are depicted in Figure 4, and the outcomes are quite similar to that of Figure 3.

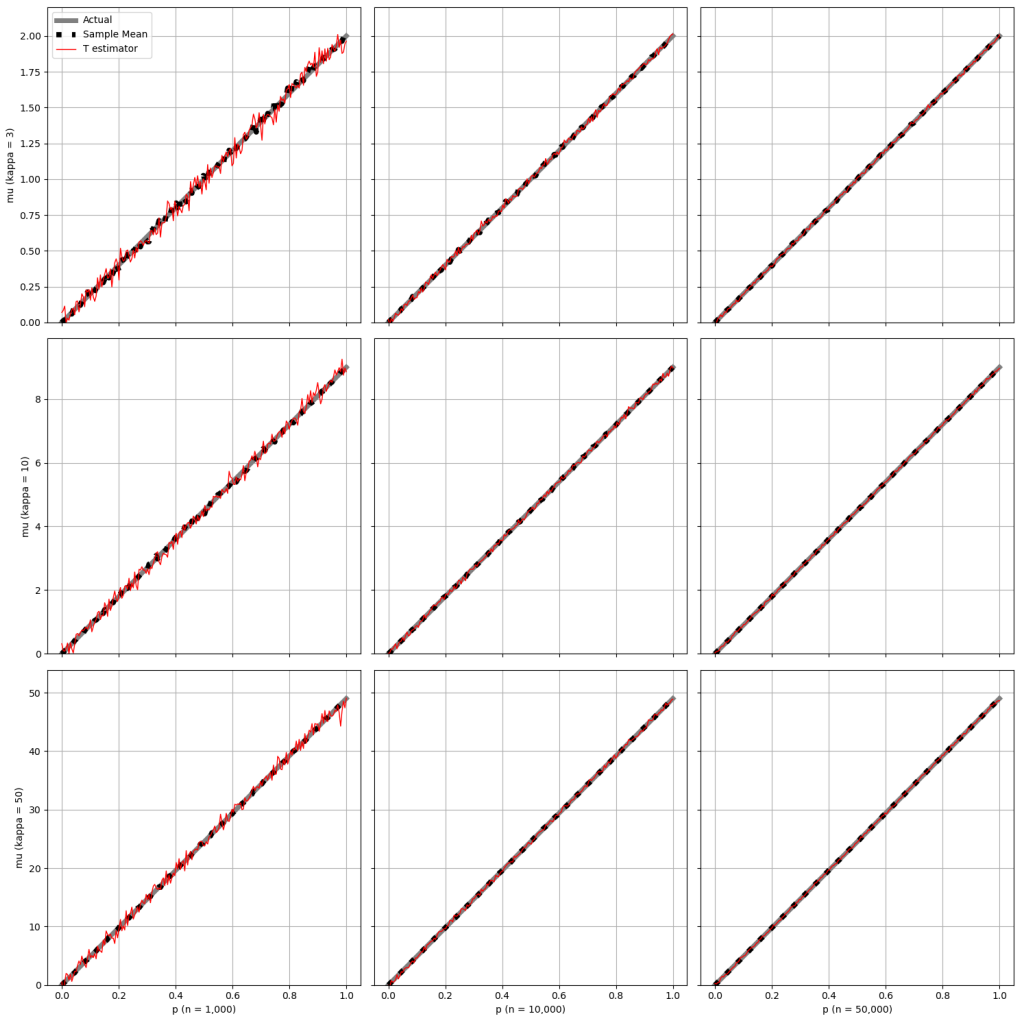


Figure 4. Simulation of binomial distribution various sizes of κ and n.

The experimental setup for this work was designed as a simulation framework, implemented using the Python programming language. Random numbers were utilized to generate synthetic datasets, model parameters, and other relevant variables, ensuring a controlled and reproducible environment for testing and analysis.

7. The General RR Algorithm

We now consider a discrete random variable, X, with generic finite support, $S(X)$, having κ values. We would like to generalize our approach and construct an RR, Y, for X, which will be used for the estimation of the mean value $\mathbb{E}(X) = \mu$. In order to see how things could work, consider the following total mean value formula:

$$\mathbb{E}(Y) = \mathbb{E}(\mathbb{E}(Y|X)) = \sum_{x \in S(X)} f_X(x)\mathbb{E}(Y|X = x) = \sum_{x \in S(X)} f_X(x)\mathbb{E}(Y[x]) \tag{19}$$

In the sequel, we have

$$\mathbb{E}(Y[x]) = \sum_{y \in S(X)} yf(y|x) = xf(x|x) + \sum_{\substack{y \in S(X) \\ y \neq x}} yf(y|x)$$

Substitute into (19) to obtain

$$\mathbb{E}(Y) = \sum_{x \in S(X)} f(x|x) x f_X(x) + \sum_{x \in S(X)} f_X(x) \sum_{\substack{y \in S(X) \\ y \neq x}} y f(y|x)$$

The right-hand side of the above equality becomes manageable if we assume that $f(x|x)$, $f(y|x)$ are constant, i.e., independent from x, y. This remark makes (15) a natural choice and, with this at hand,

$$\begin{aligned} \mathbb{E}(Y) &= (1-\alpha) \sum_{x \in S(X)} x f_X(x) + \frac{\alpha}{\kappa - 1} \sum_{x \in S(X)} f_X(x) \sum_{\substack{y \in S(X) \\ y \neq x}} y \\ &= (1-\alpha) \sum_{x \in S(X)} x f_X(x) + \frac{\alpha}{\kappa - 1} \sum_{x \in S(X)} f_X(x) \left(\sum_{y \in S(X)} y - x \right) \end{aligned}$$

By using the facts that

$$\sum_{x \in S(X)} f_X(x) = 1 \quad , \quad \mathbb{E}(X) = \sum_{x \in S(X)} x f_X(x)$$

and letting

$$s := \sum_{y \in S(X)} y,$$

we obtain the key formula

$$\mathbb{E}(Y) = \left(1 - \frac{\kappa}{\kappa - 1} \alpha \right) \mathbb{E}(X) + \frac{\alpha}{\kappa - 1} s$$

We can gain further insights from the formula above by considering the average of the values of X, $\tilde{x} := \frac{s}{\kappa}$. Then,

$$\mathbb{E}(Y) = \left(1 - \frac{\kappa}{\kappa - 1} \alpha \right) \mu + \frac{\kappa}{\kappa - 1} \alpha \tilde{x} = \mu + \frac{\kappa}{\kappa - 1} \alpha (\tilde{x} - \mu) \tag{20}$$

Note that Proposition 8 and (17) still apply for $y \in S(X)$. Actually, Equations (20) and (17) are inherently similar and have very interesting implications.

1. First of all, (20) is actually a corollary of (17). We can even generalize the formula as follows, for an arbitrary function g:

$$\begin{aligned} \mathbb{E}(g(Y)) &= \left(1 - \frac{\kappa}{\kappa - 1} \alpha \right) \mathbb{E}(g(X)) + \frac{\kappa}{\kappa - 1} \alpha \widetilde{g(x)} = \\ &= \mathbb{E}(g(X)) + \frac{\kappa}{\kappa - 1} \alpha \left(\widetilde{g(x)} - \mathbb{E}(g(X)) \right) \end{aligned} \tag{21}$$

   where

$$\widetilde{g(x)} = \frac{1}{\kappa} \sum_{x \in S(X)} g(x)$$

   is the average of the values of $g(X)$.
2. Equation (20) (and the more general one, (21)) says that $\mathbb{E}(Y)$ is equal to μ plus an "error", which becomes larger while μ moves away from the average. In other words, the proximity of $\mathbb{E}(Y)$ to μ depends on how close is the distribution of X to the uniform distribution. Actually, $\mathbb{E}(Y) = \mu$, if and only if $\mu = \tilde{x}$. This means that X is uniform. On the other hand, when $\alpha = \frac{\kappa - 1}{\kappa}$, Y becomes uniform and $\mathbb{E}(Y) = \tilde{x}$.

3. According to Equation (20), $\mathbb{E}(Y)$ is a linear function of μ. It is exactly this linear relation that gives the opportunity for an ε-DP estimation. Actually, we have

$$\mu = \frac{(\kappa - 1)\mathbb{E}(Y) - \alpha\kappa\tilde{x}}{\kappa - 1 - \alpha\kappa}$$

The above representation has the following consequence: by having the original sample in X and obtaining the corresponding sample in Y, the statistic

$$T = \frac{(\kappa - 1)\overline{Y} - \alpha\kappa\tilde{x}}{\kappa - 1 - \alpha\kappa}$$

is an unbiased estimator for μ, provided that $\alpha \neq \frac{\kappa-1}{\kappa}$. Note that $\overline{Y}$ itself is an unbiased estimator for μ only when X is uniformly distributed.

4. It is important to repeat that the distribution of X does not enter in the analysis at any point, so there is no need to be known in advance.

For the algorithmic implementation of this general version of the RR, we assume that $S(X) = \{v_0, v_1, \ldots, v_{k-1}\}$. Therefore, X can be represented as v_J, where the random variable J has support $S(J) = \{0, 1, \ldots, k-1\}$. This allows us to recap the situation in the precious section. Moreover, the randomized response can be defined as $Y := v_{J+R}$. This idea is used in the code snippet found in Appendix A.3.

8. Discussion

Although we provide a very general definition of differential privacy (DP), our focus is restricted to the (relatively limited) case of finitely supported random variables. The reason for this choice is that it serves as the natural framework for the randomized response (RR), which is the main subject of this paper. Our results establish and confirm well-known facts, primarily for the classical binary RR, while also expanding the theory in new and meaningful directions. To enhance accessibility and reproducibility, we provide implementations of our RR schemes in simple and concise Python code.

One of the notable benefits of the general randomized response mechanism lies in its ability to handle discrete random variables effectively, especially those with finite support. This makes the RR particularly well suited for applications involving categorical data, such as binary or multi-class variables, which are prevalent in domains like surveys, behavioral studies, and medical research. By directly operating within the space of discrete outcomes, the RR avoids unnecessary transformations or approximations, preserving the integrity of the original data while adding controlled noise to ensure privacy. This property makes the RR a versatile and practical tool for privacy-preserving data collection and processing.

A possible avenue for future research involves extending our approach to encompass random variables with infinite support, whether discrete or continuous. While such extensions are theoretically interesting, their practical utility warrants further investigation. The key question is whether these extensions would address real-world challenges or remain as purely mathematical exercises. For instance, in scenarios involving high-dimensional or continuous data (e.g., sensor readings in IoT systems or medical imaging), adapting the RR to handle infinite support could enable privacy-preserving mechanisms tailored to complex datasets. Such work would bridge the gap between theoretical exploration and practical application, pushing the boundaries of privacy-preserving data analysis to encompass a wider variety of use cases. A possible extension of the integration of differential privacy (DP) and randomized response (RR) into federated learning (FL) frameworks such as in [19] can address critical privacy concerns associated with the handling of sensitive medical data. DP ensures that client-to-server communications, such as gradients or model updates, are safeguarded by adding controlled noise, preventing the inference of individual patient

data while enabling global knowledge sharing. Similarly, the RR enhances client-side operations by perturbing local data or model parameters ensuring strong privacy guarantees without compromising the statistical properties needed to address class imbalance. By incorporating these privacy-preserving techniques, the proposed retrogress-resilient FL framework not only protects sensitive data but also builds trust among stakeholders, fostering a broader adoption of FL in real-world healthcare applications.

Author Contributions: Conceptualization, A.I., A.L. and N.K.P.; methodology, A.I., A.L. and N.K.P.; software, A.I.; validation, A.I., A.L. and N.K.P.; formal analysis, A.I., A.L. and N.K.P.; investigation, A.I., A.L. and N.K.P.; writing—original draft preparation, A.I., A.L. and N.K.P.; writing—review and editing, A.I., A.L. and N.K.P.; visualization, A.I. All authors have read and agreed to the published version of the manuscript.

Funding: The present work was funded under the SEPTON project, which received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101094901.

Data Availability Statement: All data used in this paper were randomly generated.

Conflicts of Interest: Authors Andreas Ioannidis, Antonios Litke and Nikolaos Papadakis were employed by the company Infili Technologies SA. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RR	Randomized response
DP	Differential privacy
MVUE	Minimum variance unbiased estimator
MLE	Maximum likelihood estimator
MSE	Mean square error

Appendix A. Python Code Snippets

Appendix A.1. RR for Bernoulli Trials

```
import numpy as np

'''
The arguments of the function are:
x -> an array, the realization of the random sample in X
alpha -> a number between 0 and 1, the switching probability
The functions outputs:
y -> an array, the sample of randomized responses
'''

def rr_binary(x, alpha):

    n = len(x)   # size of the sample

    r = np.random.binomial(n = 1, p = alpha, size = n)  # random sample in R
    y = x+r-2*x*r

    return y
```

Appendix A.2. RR for Values $0, 1, \ldots, \kappa - 1$

```
'''
The arguments of the function are:
kappa -> an integer, the size of the support of X
x -> an array, the realization of the random sample in X
alpha -> a number between 0 and 1, the switching probability
The functions outputs:
y -> an array, the sample of randomized responses
'''

def rr_finite(kappa, x, alpha):

    n = len(x)   # size of the sample
    sx = np.arange(kappa)   # support of X
    fr = np.array([1-alpha]+[alpha/(kappa-1)]*(kappa-1)) # distribution of R

    r = np.random.choice(a = sx, size = n, p = fr)   # random sample in R
    y = (x+r) % kappa

    return y
```

Appendix A.3. The General RR Algorithm for Finite Values

```
'''
The arguments of the function are:
sx -> an array [v_0,v_1,...,v_{k-1}], the support of X
x -> an array, the realization of the random sample in X
alpha -> a number between 0 and 1, the switching probability
The functions outputs:
y -> an array, the sample of randomized responses
'''

def rr_general(sx, x, alpha):

    k = len(sx)  # size of the support

    # dictionary of the indices
    ind = {}
    for i in range(k):
        ind[sx[i]]=i

    xi = np.array([ ind[v] for v in x ])  # sample into indices
    yi = rr_finite(kappa = k, x = xi, alpha = alpha)  # rr on the indices
    y = np.array([ sx[i] for i in yi ])  # back to x

    return y
```

References

1. Warner, S.L. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.* **1965**, *60*, 63–69. [CrossRef] [PubMed]
2. van den Hout, A.; van der Heijden, P.G. Randomized response, statistical disclosure control and misclassification: A review. *Int. Stat. Rev.* **2002**, *70*, 269–288. [CrossRef]
3. Wang, Y.; Wu, X.; Hu, D. Using randomized response for differential privacy preserving data collection. In Proceedings of the EDBT/ICDT Workshops, Bordeaux, France, 15–18 March 2016; Volume 1558, pp. 0090–6778.
4. Chaudhuri, A.; Mukerjee, R. *Randomized Response: Theory and Techniques*; Routledge: New York, NY, USA, 2020.
5. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the TCC'06: Proceedings of the Third Conference on Theory of Cryptography, New York, NY, USA, 4–7 March 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.

6. Dwork, C. Differential privacy: A survey of results. In Proceedings of the International Conference on Theory and Applications of Models of Computation, Xi'an, China, 25–29 April 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–19.
7. Bönisch, F. Differential Privacy: General Survey and Analysis of Practicability in the Context of Machine Learning. Master's Thesis, Freie Universität Berlin, Berlin, Germany, 2019.
8. Aitsam, M. Differential privacy made easy. In Proceedings of the 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE), Lahore, Pakistan, 2–4 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7.
9. Smith, A. Efficient, differentially private point estimators. *arXiv* **2008**, arXiv:0809.4794.
10. Wasserman, L.; Zhou, S. A statistical framework for differential privacy. *J. Am. Stat. Assoc.* **2010**, *105*, 375–389. [CrossRef]
11. Liu, W.; Zhang, Y.; Yang, H.; Meng, Q. A survey on differential privacy for medical data analysis. *Ann. Data Sci.* **2024**, *11*, 733–747. [CrossRef] [PubMed]
12. Ioannidis, A.; Litke, A.; Papadakis, N. Randomized Response and Differential Privacy. In Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments, Crete, Greece, 26–28 June 2024; pp. 600–605.
13. Kallenberg, O. *Foundations of Modern Probability*; Springer: New York, NY, USA, 1997.
14. Kolmogorov, A.N.; Fomin, S.V. *Introductory Real Analysis*; Dover: New York, NY, USA, 1975.
15. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
16. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
17. Roussas, G.G. *A Course in Mathematical Statistics*, 2nd ed.; Academic Press: San Diego, CA, USA, 1997.
18. Hogg, R.V.; McKean, J.W.; Craig, A.T. *Introduction to Mathematical Statistics*, 8th ed.; Pearson: Boston, MA, USA, 2019.
19. Chen, Z.; Yang, C.; Zhu, M.; Peng, Z.; Yuan, Y. Personalized Retrogress-Resilient Federated Learning Toward Imbalanced Medical Data. *IEEE Trans. Med. Imaging* **2022**, *41*, 3663–3674. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.