# Analyzing the Shuffle Model through the Lens of Quantitative Information Flow

Mireya Jurado
*Florida International University*
Miami, USA

Ramon G. Gonze
*Universidade Federal de Minas Gerais*
Belo Horizonte, Brazil
*Inria Saclay, École Polytechnique*
Palaiseau, France

Mário S. Alvim
*Universidade Federal de Minas Gerais*
Belo Horizonte, Brazil

Catuscia Palamidessi
*Inria Saclay and LIX, École Polytechnique*
Palaiseau, France

2023 IEEE 36th Computer Security Foundations Symposium (CSF) | 979-8-3503-2192-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSF57540.2023.00033

***Abstract*—Local differential privacy (LDP) is a variant of differential privacy (DP) that avoids the necessity of a trusted central curator, at the expense of a worse trade-off between privacy and utility. The shuffle model has emerged as a way to provide greater anonymity to users by randomly permuting their messages, so that the direct link between users and their reported values is lost to the data collector. By combining an LDP mechanism with a shuffler, privacy can be improved at no cost for the accuracy of operations insensitive to permutations, thereby improving utility in many analytic tasks. However, the privacy implications of shuffling are not always immediately evident, and derivations of privacy bounds are made on a case-by-case basis.**

**In this paper, we analyze the combination of LDP with shuffling in the rigorous framework of quantitative information flow (QIF), and reason about the resulting resilience to inference attacks. QIF naturally captures (combinations of) randomization mechanisms as information-theoretic channels, thus allowing for precise modeling of a variety of inference attacks in a natural way and for measuring the leakage of private information under these attacks. We exploit symmetries of $k$-RR mechanisms with the shuffle model to achieve closed formulas that express leakage exactly. We provide formulas that show how shuffling improves protection against leaks in the local model, and study how leakage behaves for various values of the privacy parameter of the LDP mechanism.**

**In contrast to the *strong adversary* from differential privacy, who knows everyone's record in a dataset but the target's, we focus on an *uninformed adversary*, who does not know the value of any individual in the dataset. This adversary is often more realistic as a consumer of statistical datasets, and indeed we show that in some situations, mechanisms that are equivalent under the strong adversary can provide different privacy guarantees under the uninformed one. Finally, we also illustrate the application of our model to the typical strong adversary from DP.**

***Keywords*-quantitative information flow, formal security models, differential privacy, shuffle model**

## I. Introduction

Differential privacy (DP), introduced by Dwork and her colleagues [1], [2], is one of the most successful frameworks for privacy protection. The original definition, which is now called *central* DP, assumes the existence of a *trusted curator* who has access to the raw user data and is in charge of receiving queries to this data and reporting the corresponding answers, suitably obfuscated so to make them essentially insensitive to any single data point. Differential privacy provides formal privacy guarantees and has various desirable properties, such as compositionality, which ensure its robustness to repeated queries. However, the fact that all user data is in the hands of one party means that there is a single point of failure: the privacy of all users depends on the integrity of the curator, and on his capability to protect them from security breaches.

As an alternative to the central model, researchers have proposed *local* differential privacy (LDP) [3], [4], which does not require a trustworthy data curator. In this model, each user applies an LDP-compliant protocol to perturb her data on the client side and sends it to an aggregator. Then, an analyst can estimate the desired query based on the collected noisy data from all users on the server side. The result is guaranteed to be private due to the postprocessing property. LDP has become very popular, partially thanks to its adoption by tech giants such as Google [5], [6], Apple [7], [8], and Microsoft [9], that have deployed LDP-compliant algorithms into their products to collect users' usage statistics.

In comparison with central DP, however, LDP suffers from a worse trade-off between privacy and utility. Namely, in order to achieve the same accuracy as in the central model, we need either to lower the level of individual protection, or to provide more data samples. Indeed, there are a number of lower bounds on the error of locally private protocols that strongly separate the local and the central model [10], [11].

The respective drawbacks of the central and the local model have stimulated the search for different architectures, and in this context, the *shuffle model* [12] has emerged as an appealing alternative. This model, in its simplest form, assumes a data collector who receives one message from each of the users, as in LDP. In contrast to the latter, however, it also assumes that a mechanism is in place to randomly permute the messages before they reach the data collector, so that any direct association between users and their reported values is lost. In this way, shuffling provides privacy amplification at no cost for the utility for the accuracy of commutative operations, i.e., operations that are insensitive to the order of data. These


DOI 10.1109/CSF57540.2023.00033

correspond to a large class of the most typical analytical tasks, such as sum, average, and histogram queries. In this way, the trade-off between privacy and utility is substantially improved for those queries, and in some cases, it even becomes very close to that of the central model [13].

Research on the shuffle model has focused on finding bounds for the level of privacy obtained after shuffling, expressed in terms of the privacy parameter (or parameters, in case of approximate LDP) of the local obfuscation mechanism used in combination with the shuffler. These bounds have been improved over the years, but the proofs are quite involved and produced in a case-by-case basis, without a unifying framework to reason about inference attacks against the shuffle model altogether.

In this paper, we take a different perspective by analyzing the shuffle model from the point of view of *quantitative information flow* (QIF) [14], which is a rigorous framework grounded on sound information– and decision-theoretic principles to reason about the leakage caused by inference attacks. QIF has been successfully applied to a variety of privacy and security analyses, including searchable encryption [15], intersection and linkage attacks against $k$-anonymity [16] and very large, longitudinal microdata collections [17], and differential privacy [18], [19].

In the QIF framework, a system is modeled as information-theoretic channel taking in some secret input and producing some observable output. The information leakage is defined as the difference between the vulnerability of the secret (i.e., the amount of useful information to perform an attack) before and after passing through the channel (prior and posterior vulnerabilities, respectively). The vulnerability, indeed, increases because the attacker can infer information about the input by observing the output. The vulnerability measure also depends on the adversary's goals and capabilities, which are captured in the state-of-the-art QIF framework of $g$-leakage [20] by using suitable *gain functions*. We model both the shuffler and the LDP mechanism as information-theoretic channels, and their composition with the operation of *channel cascading*, and we study their leakage. In this paper, we understand vulnerability as the adversary's ability to infer personal data, and can thereby be considered the inverse of privacy; i.e., the higher the vulnerability, the lower the privacy, and vice-versa.

We examine the *single-message shuffle model protocols*, in which each user sends a single data point to the collector, and also on the $k$-ary randomized response ($k$-RR) mechanism [21] which is one of the most popular mechanisms for LDP, and the core of Google's RAPPOR [5]. We focus on the case in which the goal of the adversary is to guess the secret value of a *single* individual (target) from the dataset, which corresponds to the typical scenario of differential privacy in which the reported answer should not allow an observer to distinguish with confidence between two adjacent datasets [22]–[25]. [1]

A key distinction of our work, however, is that we focus on an *uninformed adversary*, who does not know anyone's values before accessing a data release, and assumes a uniform prior on datasets. It is well known from the literature that the guarantees against inferences provided by differential privacy hold only w.r.t. what is usually called the *strong adversary*, who knows everyone's data except those of the target individual. However, adversaries with weaker, less informed priors, can often benefit *more* from an inference attack than the strong adversary, simply because there is more to be learned by them [22]–[25]. The fact that differential privacy's guarantees against a strong adversary do not necessarily carry over to less knowledgeable adversaries is commonly overlooked in some interpretations of the framework. [2] We refer to recent work by Tschantz et al. [22] for an excellent presentation of the issue.

Another important reason to study the uninformed adversary is that it is more realistic, as usually in DP consumers cannot access the micro-data directly. This is important if we want to compare mechanisms for privacy: it could be the case that two mechanisms $M$ and $M'$ are equivalent under the strong adversary, but not under the uninformed adversary. For example, assume for simplicity that each record contains one bit, and that $M$ outputs the last bit (i.e., the content of the last record) in the dataset, or its complement, with probabilities $e^\epsilon/1+e^\epsilon$ and $1/1+e^\epsilon$, respectively. In contrast, $M'$ computes the binary sum of the other bits, and behaves exactly as $M$ if this sum is $0$, otherwise, it outputs the same last bit or its complement, but with inverted probabilities w.r.t. what $M$ does. It is easy to see that both $M$ and $M'$ are exactly $\epsilon$-differentially private, so they are equivalent under the strong adversary (who knows all the records except the last one), but $M'$ is more private than $M$ under the uninformed adversary. More specifically, in $M'$ the reported value does not give any information on the original value of the last record (the posterior probability of it being $0$ or $1$ are the same, as in the prior), whereas in $M$ some information is gained.

In any case, once we have a QIF model for LDP and shuffle, the same techniques can be applied to derive results about their information leakage properties of other variants of adversaries.

We investigate extensively the binary case, i.e., when $k = 2$. Despite its simplicity, the binary case is quite ubiquitous, and deserves special attention. Databases often contain binary attributes, modeling the presence or absence of a given feature or storing "yes" or "no" answers to sensitive questions. Indeed, randomized response was developed as a survey technique motivated by the need to collect binary responses [26]. Successively, we extend our investigation to a generic $k$.

[1] Notice that depending on the variant of differential privacy and attack scenario considered, the adversary's prior on the secret value may vary. QIF, however, explicitly separates the adversary's goals and capabilities, modeled as a gain function, from her prior knowledge on the secret, modeled as a prior distribution. Here our comparison is focused on the former.

[2] For this reason, we believe that the term "strong adversary" is misleading, and that "informed adversary" would be preferable. However, in this paper we stick to the terminology from the literature.

### A. Contribution

Our contributions are the following:

- We study the information leakage of different combinations of $k$-RR and of the shuffler (Sec. III). To the best our knowledge, this is the first formal QIF model for the combination of LDP and shuffle mechanisms. In particular, we prove that they commute, i.e., that it is equivalent (w.r.t. leakage) to first apply $k$-RR and then the shuffler, or to do the opposite (Proposition 11). [3]
- We investigate leakage under uninformed adversaries for $k{=}2$ (Sec. IV), and derive the first exact, closed formulas for posterior vulnerability, and therefore for leakage (Theorems 17 and 19).
- We investigate leakage under uninformed adversaries for generic values $k{\geq}2$ (Sec. VI). Although we derive the first formulas for the vulnerabilities in this case (Propositions 20 and 21), they are neither closed nor computationally efficient. Nevertheless, we are able to provide novel asymptotic bounds on leakage (Theorem 20) by uncovering a surprising connection between our scenario of interest and a well-known combinatoric problem.
- We use the above formulas to study leakage as the size of the dataset increases, for various privacy parameters of $k$-RR (Sec. IV-B, VI-B and VI-C).
- We provide a brief discussion on how our QIF model, being parametric on the adversary's prior knowledge, can be used to reason about the strong adversary from differential privacy (Sec. V).
- We show that in the case of uninformed adversaries, shuffling is much more effective than noise for leakage reduction, hence the best trade-off between privacy and utility may be obtained by using the shuffling alone. In contrast, in the case of the strong adversary, noise obfuscation plays a crucial role in privacy protection, hence the best trade-off is achieved by combining noise and shuffling (Sec. IV and VI).

### B. Related Work

The shuffle model was the core idea in the Encode, Shuffle, Analyze (ESA) model by Bittau et al. [12] (see also [28] for a revised version of that work). Cheu et al. [13] formalized the shuffle model and provided the first separation result showing that the shuffle model is strictly between the central and the local models of DP. Characterizing the exact nature of this separation has been the aim of many subsequent works. Erlingsson et al. [29] showed that a trusted shuffler amplifies the privacy guarantees against an adversary who is not able to access the outputs from the local randomizers but only sees the shuffled output. Balle et al. [27] improved and generalized the results by Erlingsson et al. and provided a family of methods to analyze privacy amplification in the shuffle model. Feldman et al. [30] suggested an asymptotically optimal dependence of the privacy amplification on the privacy parameter of the local randomizer. Koskela et al. [31], [32] proposed a numerical approach to estimate tight bounds based on weak adversaries.

Other directions of research related to shuffle models address summation queries [13], [33]–[37] and histogram queries [38], [39]. Balle et al. [27] proposed a single-message protocol for messages in the interval $[0, 1]$, and Cheu et al. [13] conducted a study on bounded real-valued statistical queries using additional communication costs. Ishai et al. [35] analyzed a protocol that reduced the number of messages in the summation query under shuffle models, and Balcer et al. [38] proposed a shuffle mechanism for histogram queries.

Further relevant work involves robust shuffle differential privacy [40]–[42]. Shuffle models can provide a targeted level of privacy protection only with at least a specific number of users participating in the shuffle. If the number of data providers does not reach a certain quantity, the level of privacy protection degrades. Thus, studies on robust shuffle differential privacy have gained recent attention in the community. Balcer et al. [40] explores robust shuffle private protocols and suggests a relationship between robust shuffle privacy and pan-privacy.

Connections between QIF and differential privacy have been explored in the literature. Barthe and Köpf [43] relate centralized differential privacy and information-theoretic notions of leakage, establishing bounds on the former in terms of the value of $\epsilon$. In a similar line of work, Alvim et al. [18] employ QIF to analyze leakage and utility in oblivious, centralized differentially-private mechanisms. They improve some of the bounds by Barthe and Köpf, and provide a mechanism that, if some constraints are satisfied, maximizes utility for a given level of differential privacy. Both of these works, however, focus on the centralized model of differential privacy (rather than on the local model, as we do), and do not consider shuffling. Chatzikokolakis et al. [44] investigate leakage orderings induced by different differential privacy mechanisms, but, again, do not consider shuffling. To the best of our knowledge, our work is the first to provide a QIF analysis of the combination of locally differentially-privacy and shuffle mechanisms. Moreover, we provide closed formulas (rather than bounds) to compute leakage in practical scenarios.

### C. Plan of the paper

In Sec. II we review fundamentals from QIF, LDP, $k$-RR, and the shuffle model. In Sec. III we provide a QIF model for $k$-RR and the shuffler as channels and investigate different combinations of such channels. In Sec. IV we study leakage under an adversary with a uniform prior over datasets focusing on a single target, for $k{=}2$. In Sec. V we briefly consider the typical strong adversary from differential privacy also for $k{=}2$. In Sec. VI we extend our investigation from Sec. IV to generic values of $k{\geq}2$. In Sec. VII we present consequences of our findings, and in Sec. VIII we conclude. The appendix contains further technical details, and full proofs can be found in a corresponding technical report [45].

[3] In Balle et al. [27], the LDP mechanism and the shuffler do not commute, but that is because they use a notion of attacker that, in addition to knowing the true values of all records but one, can also observe whether users report their true values or not (except for the user under attack).

## II. Preliminaries

Firstly, we review the analytical framework of *quantitative information flow*. While we introduce fundamental notation and vocabulary, the definitive resource on quantitative information flow with definitions, theorems, and proofs can be found in [14]. Secondly, we review $k$-RR local differential privacy along with the shuffle model.

### A. QIF

The quantitative information flow (QIF) framework captures the adversary's knowledge, goals, and capabilities, and from that, quantifies the leakage of information caused by a corresponding optimal inference attack. The framework is grounded on sound information– and decision-theoretic principles enabling the rigorous assessment of how much information leakage a system allows *in principle*, and independently from the adversary's computational power [14], [46]–[48]. Hence, QIF guarantees hold no matter the particular tactic or algorithm the adversary employs to execute the attack, as what is measured is exactly how much sensitive information is leaked by the best possible such tactic or algorithm.

QIF separates (1) the adversary's knowledge (modeled as a prior distribution on secret values) from (2) her intentions and capabilities (modeled as a gain function), and that is separated from (3) the description of the system being run (modeled as an information-theoretic channel). We describe these components in more detail now.

QIF assumes the input has a probability distribution $\pi$ that is known to an adversary. We denote the random variables associated with the channel's input and output as $X$ and $Y$, respectively. The system is modeled as a channel matrix $\mathsf{C}$, where $\mathsf{C}_{x,y}$ contains the conditional probability $Pr(y \mid x)$. We assume the adversary knows how the channel works as well as the entries in $\mathsf{C}_{x,y}$. Knowing the distribution on secrets and the channel matrix, she can update her knowledge about $X$ to a posterior distribution $Pr(x \mid y)$. Since each output $y$ also has a probability $Pr(y)$, the channel matrix $\mathsf{C}$ provides a mapping from any prior $\pi$ to *distributions on posterior distributions*, which we call a hyper-distribution and denote $[\pi \triangleright \mathsf{C}]$.

There is no one "right" way to measure how a system affects a secret since the adversary's probability of success depends on their goals and operational context. To address this, QIF uses the $g$-leakage framework, introduced by [20]. This framework defines the *vulnerability* of secret $X$ with respect to specific operational scenarios. An adversary is given a set $\mathcal{W}$ of actions (or guesses) that she can make about the secret, and given a *gain function* $g(w, x)$ ranging over non-negative reals which defines the gain of selecting the action $w$ when the real secret is $x$. [4] An optimal adversary will choose an action that maximizes her expected gain with respect to $\pi$. The gain function then determines a secret's vulnerability.

Given a prior distribution on secrets $\pi$, prior $g$-vulnerability, denoted $V_g(\pi)$, represents the adversary's *expected gain* of her optimal action based only on $\pi$, i.e., before observing the channel output.

**Definition 1** (Prior vulnerability)**.** *Given a prior $\pi$ and a gain function $g$, the corresponding* prior vulnerability *is given by*

$$V_g(\pi) := \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x \cdot g(w, x)\,. \tag{1}$$

In the posterior case, the adversary observes the output of the system which allows her to improve her action and consequent expected gain.

**Definition 2** (Posterior vulnerability)**.** *Given a prior $\pi$, a gain function $g$, and channel matrix $\mathsf{C}$ from $\mathcal{X}$ to $\mathcal{Y}$, the corresponding* posterior vulnerability *is given by*

$$V_g[\pi \triangleright \mathsf{C}] := \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x \cdot \mathsf{C}_{xy} \cdot g(w, x)\,. \tag{2}$$

The choice of the gain function $g$ covers a vast variety of adversarial scenarios [20]. Indeed, *any* vulnerability function satisfying a set of basic information-theoretic axioms can be expressed as $V_g$ for a properly constructed $g$ [49].

We measure the channel leakage by comparing the prior and posterior $g$-vulnerability, which quantifies how much a specific channel $\mathsf{C}$ *increases* the vulnerability of the system. This comparison can be done additively or multiplicatively. Channel matrices can be large and difficult to evaluate, necessitating simplification. It is interesting, hence, to consider the concept of channel equivalence w.r.t. information leakage properties. Two channels $\mathsf{C}$ and $\mathsf{C}'$ defined on the same input set (but possibly different output sets) are considered *equivalent*, denoted by $\mathsf{C} \equiv \mathsf{C}'$, iff, for all priors $\pi$ on their input set and gain function $g$, their posterior vulnerabilities are the same, i.e., $V_g[\pi \triangleright \mathsf{C}] = V_g[\pi \triangleright \mathsf{C}']$. (Notice that if the posterior vulnerabilities for both channels are same, then so will be their corresponding multiplicative and additive leakages, since the channels share the same prior vulnerability.) One way to simplify a channel matrix into an equivalent one is to adjust extraneous structure by deleting output labels and adding similar columns together (columns that are scalar multiples of each other), since output labels do not affect leakage.

Importantly for this work, channel matrices can compose in *cascades* such that the output of one channel becomes the input for another. As defined in [14],

**Definition 3** (Channel cascade)**.** *Given channel matrices $\mathsf{C} : \mathcal{X} \to \mathcal{Y}$ and $\mathsf{D} : \mathcal{Y} \to \mathcal{Z}$, the* cascade *of $\mathsf{C}$ and $\mathsf{D}$ is the channel matrix $\mathsf{CD}$ of type $\mathcal{X} \to \mathcal{Z}$, where $\mathsf{CD}$ is given by ordinary matrix multiplication.*

Cascades model sequential operations, where the adversary observes the final output. In the same way that $\mathsf{C}_{x,y}$ specifies $Pr(y \mid x)$, a cascade $\mathsf{CD} : \mathcal{X} \to \mathcal{Z}$ specifies $Pr(z \mid x)$. From the perspective of information leakage, cascades have an

[4] In principle, the return value of a gain function may be negative, as long as the corresponding vulnerability for all priors is non-negative. This restriction is imposed so the ratio between posterior and prior vulnerabilities remains meaningful. However, the restriction can be met by introducing an action in the gain function having gain value of 0 for all secrets, representing, e.g., the action of not performing an attack. Another solution is to shift the gain function by adding a suitable constant positive real number to its return value.

important operational significance: $\mathsf{D}$ here acts as a sanitization policy, suppressing the release of $Y$.

In fact, by the data-processing inequality for the $g$-leakage framework (expressed in Theorem 4 below), for any $\mathsf{D}$ in cascade $\mathsf{CD}$, we know that leakage can never be increased. To understand this property, let us define the *refinement relation* among channels of the same input space as $\mathsf{C} \sqsubseteq \mathsf{C}'$, meaning that $\mathsf{C}$ is refined by $\mathsf{C}'$ (or, equivalently, that $\mathsf{C}'$ refines $\mathsf{C}$) iff, for all priors $\pi$ on their input set and gain function $g$, their posterior vulnerabilities satisfy $V_g[\pi \triangleright \mathsf{C}] \geq V_g[\pi \triangleright \mathsf{C}']$. Then the data-processing inequality for the $g$-leakage framework is established as follows [14].

**Theorem 4** (Data-processing inequality)**.** *Given channel matrices $\mathsf{C} : \mathcal{X} \to \mathcal{Y}$ and $\mathsf{C}' : \mathcal{X} \to \mathcal{Z}$, we have that $\mathsf{C} \sqsubseteq \mathsf{C}'$ iff there exists a channel matrix $\mathsf{D} : \mathcal{Y} \to \mathcal{Z}$ such that $\mathsf{CD} = \mathsf{C}'$.*

### B. Local Differential Privacy (LDP)

Let $\mathcal{K}$ be the set of values for a single individual's sensitive attribute, and $k$ be the size of $\mathcal{K}$. We say that a mechanism $\mathcal{R}$ is $\epsilon$-LDP if, for all $x, x', y \in \mathcal{K}$, we have

$$Pr(\mathcal{R}(x) = y) \leq e^{\epsilon}\ Pr(\mathcal{R}(x') = y)\,,$$

where $Pr(a)$ represents the probability of the event $a$ and $e$ is the exponentiation operator.

### C. $k$-ary Randomized Response ($k$-RR)

The $k$-ary Randomized Response with privacy parameter $\epsilon$ is a mechanism $\mathcal{R}$ defined as:

$$Pr(\mathcal{R}(x) = y) = \begin{cases} e^{\epsilon}/(k - 1 + e^{\epsilon}), & \text{if } y = x, \\ 1/(k - 1 + e^{\epsilon}), & \text{otherwise.} \end{cases} \tag{3}$$

It is easy to prove that such a mechanism satisfies $\epsilon$-DP [50].

### D. Shuffler, simplified version

We consider only the single-message shuffler model. We can define the shuffler as a mechanism $\mathcal{R}$ that takes as input a tuple of elements of $\mathcal{K}$ of some fixed length $n$, and produces a random permutation of it, with uniform probability. Namely, for all $(x_0, x_1, \ldots, x_{n-1}) \in \mathcal{K}^n$ and all permutations $(y_0, y_1, \ldots, y_{n-1})$ of $(x_0, x_1, \ldots, x_{n-1})$, we have

$$Pr(\mathcal{R}(x_0, x_1, \ldots, x_{n-1}) = (y_0, y_1, \ldots, y_{n-1})) \;=\; \frac{1}{n!}\,.$$

## III. QIF model for local differential privacy and shuffling

In this section, we provide a formal model based on the QIF framework for the application of LDP mechanisms and of shuffle mechanisms to datasets containing sensitive values. The model will be instrumental in the next sections, where we investigate information leakage properties of these mechanisms and of their various compositions in different configurations of the adversary's prior knowledge.

### A. Sensitive values, datasets, and the general scenario

We begin by formalizing the general scenario we consider.

**Definition 5** (Sensitive values and datasets)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$* individuals of interest*, and $\mathcal{K}$ be a set of $k \geq 2$ possible values for each individual's* sensitive attribute*. A* dataset *$x$ is a tuple $(x_0, x_1, \ldots, x_{n-1})$ in which each element $x_i \in \mathcal{K}$ is the value of the sensitive attribute for individual $i \in \mathcal{N}$. The domain of all possible datasets is, hence, $\mathcal{K}^n$, and it has size $k^n$.*

Moreover, we adopt the following notation and terminology. Given a set $\mathcal{K} = \{\kappa_0, \kappa_1, \ldots, \kappa_{k-1}\}$ of values for the sensitive attribute, and a dataset $x \in \mathcal{K}^n$, we let

- the tuple $x = (x_0, x_1, \ldots, x_{n-1})$ be denoted by the sequence $x_0 x_1 \ldots x_{n-1}$ of its elements;
- $n_{\kappa_j}(x)$ represent the the number of individuals in the dataset $x$ having sensitive value $\kappa_j \in \mathcal{K}$;
- $h(x) = (\kappa_0{:}n_{\kappa_0}(x),\ \kappa_1{:}n_{\kappa_1}(x),\ \ldots,\ \kappa_{k-1}{:}n_{\kappa_{k-1}}(x))$ be the the histogram of dataset $x$, containing a list of each possible value $\kappa_j \in \mathcal{K}$ followed by the count $n_{\kappa_j}(x)$ of individuals in $x$ with that value; and
- $\#h(x) = |\{x' \in \mathcal{K}^n \mid h(x') = h(x)\}|$ be the number of distinct datasets in $\mathcal{K}^n$ having the same histogram as $x$.

We consider a dataset $x \in \mathcal{K}^n$ containing the real value of the sensitive attribute for all individuals in $\mathcal{N}$. We also assume that a data analyst wants to infer some statistical information about the original dataset (e.g., an average or count), but the dataset's exact contents –in particular, the link between individuals and their sensitive attribute– are considered secret.

**Example 6** (Running example)**.** *Consider a scenario in which there are $n = 3$ individuals and a sensitive binary attribute with possible values in $\mathcal{K} = \{\mathtt{a}, \mathtt{b}\}$. The set of all possible datasets is then $\{\mathtt{a}, \mathtt{b}\}^3 = \{\mathtt{aaa}, \mathtt{aab}, \mathtt{aba}, \mathtt{abb}, \mathtt{baa}, \mathtt{bab}, \mathtt{bba}, \mathtt{bbb}\}$.*

### B. Sanitization mechanisms as channels

We consider two sanitization mechanisms which can be employed either individually or in combination. The $k$-RR mechanism is applied to each individual's sensitive value to introduce uncertainty, while the shuffle mechanism permutes dataset entries to obfuscate the link between each sensitive value and its owner.

*a) The shuffle channel:* The full channels of these mechanisms are quite unwieldy. For example, the shuffle mechanism, with the input dataset `aab`, can output `abb`, `aba`, or `baa`, each with probability $1/3$. Formally, the shuffle randomly permutes the entries to obfuscate the link between each sensitive value and its owner. Abstracting from implementation details and assuming perfect shuffling, each input dataset $x$ has an equal probability $1/\#h(x)$ of mapping to any of the $\#h(x)$ possible output datasets with the same histogram.

**Definition 7** (Full shuffle channel)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ sensitive values. A* full shuffle channel *$\mathsf{S}$ is a channel from $\mathcal{K}^n$ to $\mathcal{K}^n$*

*s.t., for all $x \in \mathcal{K}^n$ and $y \in \mathcal{K}^n$,*

$$\mathsf{S}_{x,y} = \begin{cases} 1/\#h(x), & \text{if } h(y) = h(x), \\ 0, & \text{otherwise}. \end{cases}$$

Table Ib contains the full channel $\mathsf{S}$ representing the application of shuffling to the scenario of Example 6.

Conveniently, there are symmetries here that we can exploit to simplify the shuffle channel into something more tractable. Notice that once an input dataset $x$ is shuffled into an output dataset $y$, the position of each element in $y$ no longer identifies its owner, and the only information preserved is $x$'s histogram. For instance, in Table Ib, the columns corresponding to datasets `aab`, `aba`, and `baa` are identical and, from the point of view of the adversary, convey the same information. Hence, these columns can be all merged into a new column that represents only the histogram `a`:2, `b`:1 that these datasets share. This motivates the definition of a reduced shuffle channel below, whose output is simply the input dataset's histogram.

**Definition 8** (Reduced shuffle channel)**.** *Let $\mathsf{S}$ be a full shuffle channel as per Def. 8. A* reduced shuffle channel $\mathsf{S}^r$ *corresponding to $\mathsf{S}$ is a channel from $\mathcal{K}^n$ to histograms on $\mathcal{K}$ s.t., for all $x \in \mathcal{K}^n$ and histogram $z$,*

$$\mathsf{S}^r_{x,z} \;=\; \sum_{y \in \mathcal{K}^n : h(y) = z} \mathsf{S}_{x,y} \;=\; \begin{cases} 1, & \text{if } h(x) = z, \\ 0, & \text{otherwise}. \end{cases}$$

Table Ic contains the channel $\mathsf{S}^r$ representing the application of reduced shuffle to the scenario of Example 6.

We can be confident that this simplification does not alter our analysis. It is known from QIF literature that merging columns that are multiples of (or identical to) each other does not alter the leakage properties of a channel [14]. The original and resulting channels are *equivalent*, in the sense that, for every $g$-vulnerability measure and prior distribution on secret values, both yield the same quantification of information leakage. Since $\mathsf{S}^r$ is obtained from $\mathsf{S}$ by merging similar columns, these channels are equivalent.

**Proposition 9** (Equivalence between full and reduced shuffle)**.** *Let $\mathsf{S}$ be a full shuffle channel as per Def. 7, and $\mathsf{S}^r$ be the reduced shuffle channel obtained from $\mathsf{S}$ as per Def. 8. Then*

$$\mathsf{S} \equiv \mathsf{S}^r.$$

*b) The $k$-RR channel:* The $k$-RR mechanism independently randomizes the sensitive value of each entry of the dataset, and outputs the resulting dataset. This process does not break the link between any individual and their sensitive value, but creates uncertainty about whether the reported value for each individual is accurate. To simplify notation, we will use $p$ to denote the probability representing that an individual's sensitive value is reported accurately, i.e., $p = e^\epsilon/k-1+e^\epsilon$ (cf. Equation 3), and $\overline{p}$ to represent $1-p$, i.e., the probability that the true value is swapped with some other one.

| N | aaa | aab | aba | baa | abb | bab | bba | bbb |
|---|---|---|---|---|---|---|---|---|
| aaa | $p^3$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $\overline{p}^3$ |
| aab | $p^2\overline{p}$ | $p^3$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $\overline{p}^3$ | $p\overline{p}^2$ |
| aba | $p^2\overline{p}$ | $p\overline{p}^2$ | $p^3$ | $p\overline{p}^2$ | $p^2\overline{p}$ | $\overline{p}^3$ | $p^2\overline{p}$ | $p\overline{p}^2$ |
| baa | $p^2\overline{p}$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^3$ | $\overline{p}^3$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p\overline{p}^2$ |
| abb | $p\overline{p}^2$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $\overline{p}^3$ | $p^3$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^2\overline{p}$ |
| bab | $p\overline{p}^2$ | $p^2\overline{p}$ | $\overline{p}^3$ | $p^2\overline{p}$ | $p\overline{p}^2$ | $p^3$ | $p\overline{p}^2$ | $p^2\overline{p}$ |
| bba | $p\overline{p}^2$ | $\overline{p}^3$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^3$ | $p^2\overline{p}$ |
| bbb | $\overline{p}^3$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^3$ |

(a) The $k$-RR channel $\mathsf{N}$ where $p$ is the probability a user responds with their true value, and $\overline{p} = 1 - p$. Cells are shaded according to the entries' values when $p = 0.75$.

| S | aaa | aab | aba | abb | baa | bab | bba | bbb |
|---|---|---|---|---|---|---|---|---|
| aaa | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| aab | 0 | 1/3 | 1/3 | 0 | 1/3 | 0 | 0 | 0 |
| aba | 0 | 1/3 | 1/3 | 0 | 1/3 | 0 | 0 | 0 |
| abb | 0 | 0 | 0 | 1/3 | 0 | 1/3 | 1/3 | 0 |
| baa | 0 | 1/3 | 1/3 | 0 | 1/3 | 0 | 0 | 0 |
| bab | 0 | 0 | 0 | 1/3 | 0 | 1/3 | 1/3 | 0 |
| bba | 0 | 0 | 0 | 1/3 | 0 | 1/3 | 1/3 | 0 |
| bbb | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

(b) Full shuffle channel $\mathsf{S}$ which receives datasets as input and produces a datasets as output.

| $\mathsf{S}^r$ | a:3, b:0 | a:2, b:1 | a:1, b:2 | a:0, b:3 |
|---|---|---|---|---|
| aaa | 1 | 0 | 0 | 0 |
| aab | 0 | 1 | 0 | 0 |
| aba | 0 | 1 | 0 | 0 |
| abb | 0 | 0 | 1 | 0 |
| baa | 0 | 1 | 0 | 0 |
| bab | 0 | 0 | 1 | 0 |
| bba | 0 | 0 | 1 | 0 |
| bbb | 0 | 0 | 0 | 1 |

(c) Reduced shuffle channel $\mathsf{S}^r$ which receives datasets as inputs and produces histograms as outputs.

TABLE I: Examples of channels for the $k$-RR and shuffling mechanisms, with $k = 2$ possible sensitive values and $n = 3$ individuals.

The $k$-RR mechanism can be modeled as a channel that probabilistically maps each input dataset $x \in \mathcal{K}^n$ to an output dataset $y \in \mathcal{K}^n$ as follows.

**Definition 10** ($k$-RR channel)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ sensitive values. A* $k$-RR channel $\mathsf{N}$ *(for "noise") with parameter $p \in [1/k, 1]$ is a channel from $\mathcal{K}^n$ to $\mathcal{K}^n$ s.t., for all $x \in \mathcal{K}^n$ and $y \in \mathcal{K}^n$,*

$$\mathsf{N}_{x,y} = \prod_{i=0}^{n-1} Pr(y_i \mid x_i),$$

*where*

$$Pr(y_i \mid x_i) = \begin{cases} p, & \text{if } y_i = x_i, \\ \overline{p}/(k-1), & \text{if } y_i \neq x_i \end{cases}$$

Notice that $p$ is at least $1/k$ so $p \geq \overline{p}/(k-1)$, indicating that the real sensitive value never has a lower probability to be reported than any other value. Table Ia contains the full channel $\mathsf{N}$ representing the application of a $k$-RR mechanism to the scenario of Example 6.

### C. The combination of sanitization mechanisms

The sanitization mechanisms presented in the previous section can be applied to an input dataset either in isolation or in combination. In the QIF framework, the effect of this combination is captured by channel cascading (Def. 3). We shall now cover the various possibilities.

We begin by analyzing the typical pipeline used in the literature, which first applies the noisy $k$-RR mechanism to the original dataset, and then applies a shuffle mechanism to the result of the first sanitization [12]. The channel cascade $\mathsf{NS}$ mirrors this process: the datasets are processed by $\mathsf{N}$, producing a randomized intermediate dataset, which are taken as input into $\mathsf{S}$ which shuffles the entries and produces a final output dataset. The matrix itself can be easily generated by matrix multiplication.

Interestingly, the order of application of the full mechanisms $\mathsf{N}$ and $\mathsf{S}$ is irrelevant, in the sense that the resulting channels will be equivalent w.r.t. the information leakage they cause. This commutativity property is formalized as follows:

**Proposition 11** (Commutativity of $k$-RR and shuffle: Full case)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ values for the sensitive attribute. Let also $\mathsf{N}$ be a $k$-RR channel as per Definition 10, and $\mathsf{S}$ be a full shuffle channel per Definition 7. Then*

$$\mathsf{NS} \equiv \mathsf{SN}.$$

Consider again the scenario of Example 6. The channel $\mathsf{NS}$ and the channel $\mathsf{SN}$ representing the application of $k$-RR followed by full shuffle or vice-versa are identical, as represented in Table II.

Finally, we notice that the equivalence between the full $\mathsf{S}$ and reduced $\mathsf{S}^r$ versions of shuffle channel (Prop. 9) is carried over to the their composition with the $k$-RR mechanism. This property will facilitate some proofs of our technical results, and is formalized below.

**Proposition 12** (Equivalence of full and reduced compositions)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ values for the sensitive attribute. Let also $\mathsf{N}$ be a full $k$-RR channel as per Definition 10, $\mathsf{S}$ be a full shuffle channel as per Definition 7, and $\mathsf{S}^r$ be a reduced shuffle channel obtained from $\mathsf{S}$ as per Definition 8. Then:*

$$\mathsf{NS} \equiv \mathsf{NS}^r \tag{4}$$

Continuing our example, the cascade $\mathsf{NS}^r$ representing the application of $k$-RR followed by the reduced shuffle is represented in Table III.

The relationship among channels is depicted in Fig. 1. Notice, however, that commutativity does not hold between $\mathsf{N}$ and $\mathsf{S}^r$, as the composition $\mathsf{S}^r\mathsf{N}$ is not mathematically consistent (the inner dimensions of the matrices do not match so multiplication is impossible). Commutativity can be recovered, however, if we define a suitable reduced counterpart to the $k$-RR channel. This possibility is explored in the Appendix.

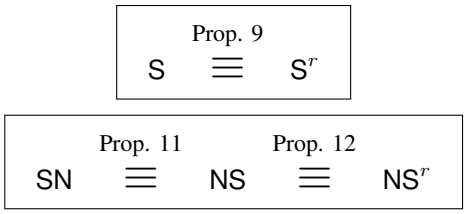


Fig. 1: Relationship among compositions of full and reduced $k$-RR and shuffle. Here ($\equiv$) denotes equivalence w.r.t. information leakage.

### D. Gain functions and priors on secrets

**On the choice of gain functions.** In this work, we focus on an adversary who has one try to correctly guess the value of a single individual chosen as a target, and has maximum benefit from a correct guess, and no benefit from an incorrect one. The gain functions associated with this type of adversary yield what is usually called *Bayes vulnerability* [14]. Given the intuitive operational interpretations and convenient mathematical properties, Bayes vulnerability and its variants have been used in many works on privacy and security [15], [17], [18], [20], [47], [51], and our work is aligned with them.

**On the choice of priors.** In this work, we focus on what we call an *uninformed adversary*. Besides the motivations given in the introduction, we remark that in QIF, uniform priors are closely linked to the maximum possible leakage that can be caused by a channel (over all priors and gain functions) [14], [52], hence, the study of uniform priors is of particular relevance for inference attacks. However, to illustrate the generality of our approach, we also show how to apply it to the strong adversary of differential privacy and we derive the leakage on some examples.

## IV. Single-target leakage of a binary value against an uninformed adversary

In this section, we study the information leakage caused by different combinations of $k$-RR and shuffling. We focus on an attack scenario with two central characteristics: (i) the adversary is attempting to guess the secret value of a single individual from $\mathcal{N}$ chosen as the target in the dataset; and (ii) the sensitive value can take binary values only, so the size $\mathcal{K}$ of the set of sensitive attributes is $k = 2$. Moreover, we focus on an uninformed adversary, with a uniform prior on datasets.

We derive exact formulas to compute prior and posterior vulnerabilities, and therefore leakage, in this attack scenario. Although these formulas are combinatorial in nature, they turn out to have computationally efficient equivalent formulations. Finally, we analyze the behavior of leakage as the size of the dataset grows and other parameters vary. Sec. V discusses a variation to this single-target adversary, while Sec. VI expands our investigation to the case of general size $k$ of the set of

| NS / SN | aaa | aab | aba | baa | abb | bab | bba | bbb |
|---|---|---|---|---|---|---|---|---|
| aaa | $p^3$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $\overline{p}^3$ |
| aab | $p^2\overline{p}$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $p\overline{p}^2$ |
| aba | $p^2\overline{p}$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $p\overline{p}^2$ |
| baa | $p^2\overline{p}$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $p\overline{p}^2$ |
| abb | $p\overline{p}^2$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $p^2\overline{p}$ |
| bab | $p\overline{p}^2$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $p^2\overline{p}$ |
| bba | $p\overline{p}^2$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(2p^2\overline{p}+\overline{p}^3)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $1/3(p^3+2p\overline{p}^2)$ | $p^2\overline{p}$ |
| bbb | $\overline{p}^3$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p\overline{p}^2$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^2\overline{p}$ | $p^3$ |

TABLE II: Channel representing both the cascade NS of $k$-RR followed by full shuffle and the cascade SN of full shuffle followed by $k$-RR, with $k = 2$ possible sensitive values and $n = 3$ individuals. Here $p$ is the probability a user responds with their true value and $\overline{p} = 1 - p$. Cells are shaded according to the entries' values when $p = 0.75$ and $\overline{p} = 0.25$.

| $\mathsf{NS}^r$ | (a:3,b:0) | (a:2,b:1) | (a:1,b:2) | (a:0,b:3) |
|---|---|---|---|---|
| aaa | $p^3$ | $3p^2\overline{p}$ | $3p\overline{p}^2$ | $\overline{p}^3$ |
| aab | $p^2\overline{p}$ | $p^3+2p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p\overline{p}^2$ |
| aba | $p^2\overline{p}$ | $p^3+2p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p\overline{p}^2$ |
| baa | $p^2\overline{p}$ | $p^3+2p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p\overline{p}^2$ |
| abb | $p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p^3+2p\overline{p}^2$ | $p^2\overline{p}$ |
| bab | $p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p^3+2p\overline{p}^2$ | $p^2\overline{p}$ |
| bba | $p\overline{p}^2$ | $2p^2\overline{p}+\overline{p}^3$ | $p^3+2p\overline{p}^2$ | $p^2\overline{p}$ |
| bbb | $\overline{p}^3$ | $3p\overline{p}^2$ | $3p^2\overline{p}$ | $p^3$ |

TABLE III: Channel $\mathsf{NS}^r$ representing $k$-RR followed by reduced shuffling, with $k = 2$ possible sensitive values and $n = 3$ individuals. Cells are shaded according to the entries' values when the probability $p$ a user responds with their true value is $0.75$ and $\overline{p} = 0.25$. Here $p^3 + 2p\overline{p}^2$ represents the largest value at approx. 0.5156 while $\overline{p}^3$ is the smallest at approx. 0.0156. Interestingly, this ordering can change with $p$.

sensitive attributes.

### A. Attack scenario and leakage formulas

We start by providing an intuitive overview of the attack scenario. Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be the set of $n \geq 1$ individuals of interest and $\mathcal{K}$ be the set of $k$ possible sensitive values. The adversary starts off by knowing $\mathcal{N}$ and $\mathcal{K}$, but does not have access to the original dataset or to its sanitized and published version. Moreover, we assume the adversary knows that all possible datasets are equally likely a priori, so her prior knowledge is captured by the uniform distribution defined, for every possible dataset $x \in \mathcal{K}^n$, as

$$\pi_x = 1/k^n. \tag{5}$$

Without loss of generality, we assume the selected target to be the first participant in the dataset, i.e., individual 0.

The adversary's action consists in picking a value $\kappa \in \mathcal{K}$ as her guess for the target's value $x_0$, obtaining some gain if her guess is correct, and no gain otherwise. The adversary's prior $\pi$ on uniform datasets implies that the adversary's a priori knowledge about the target's secret value is a uniform distribution on $\kappa \in \mathcal{K}$. The goals and capabilities of this adversary are formalized as the following gain function.

**Definition 13** (Single-target gain function)**.** *Let $\mathcal{X} = \mathcal{K}^n$ be the set of all possible (secret) datasets, and $\mathcal{W} = \mathcal{K}$ be the set of actions available to the adversary, consisting in all possible guesses for the target individual's sensitive value. The* single target gain function $g_{\mathrm{T}} : \mathcal{W} \times \mathcal{X} \to [0, 1]$ *is defined, for every action $w \in \mathcal{W}$ and secret dataset $x \in \mathcal{X}$, as*

$$g_{\mathrm{T}}(w, x) := \begin{cases} 1, & \text{if } x_0 = w, \\ 0, & \text{otherwise}, \end{cases}$$

*where $x_0$ is the sensitive value of the first individual in $x$.*

The prior vulnerability is exactly $1/k$, representing exactly the adversary's probability of correctly guessing the target individual's sensitive value in one try given only her prior knowledge about the secret.

**Proposition 14** (Prior single-target vulnerability)**.** *Given the uniform prior $\pi$ on the set $\mathcal{X} = \mathcal{K}^n$ of all possible datasets, and the single-target gain function $g_{\mathrm{T}}$ per Def. 13, the corresponding prior vulnerability is given by*

$$V_{g_{\mathrm{T}}}(\pi) = \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{K}^n} \pi_x g_{\mathrm{T}}(w, x) = 1/k.$$

We now turn our attention to the vulnerability of the secret after a sanitized version of the original dataset is released and becomes visible to the adversary. We consider the sanitization process as consisting of the application of the full $k$-RR channel $\mathsf{N}$, the shuffle channel $\mathsf{S}$, or the cascade $\mathsf{NS}$ representing the combination of both mechanisms.

First, let us consider posterior vulnerability when only the $k$-RR mechanism is used. Clearly, an adversary's optimal action is always to guess the target's reported value since $p \geq \overline{p}/k-1$, indicating it is always at least equally likely the reported value is the correct one. Therefore, the adversary's probability of correctly guessing the target's value is $p$.

**Proposition 15** (Posterior single-target vulnerability under $k$-RR)**.** *Given the uniform prior $\pi$ on the set $\mathcal{K}^n$ of all possible datasets, the single-target gain function $g_{\mathrm{T}}$ per Def. 13, and a full $k$-RR channel $\mathsf{N}$ with parameter $p \in [1/k, 1]$ per Def. 10, the corresponding posterior vulnerability is given by*

$$V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{N}] = \sum_{y \in \mathcal{K}^n} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{K}^n} \pi_x \mathsf{N}_{x,y} g_{\mathrm{T}}(w, x) = p.$$

Now let us consider the posterior vulnerability corresponding to the shuffle mechanism alone. Recall that, from Def. 8, the reduced shuffle channel $\mathsf{S}^r$ simply maps each input dataset to its histogram. In this way, the correspondence between individuals and their sensitive value is lost, but the adversary

can still observe the counts of people with each attribute. Hence, intuitively, after observing any given histogram, an adversary's optimal action is always to guess the most common value in the histogram as the target individual's value.

**Proposition 16** (Posterior single-target vulnerability under shuffle and a binary sensitive attribute)**.** *Given the uniform prior $\pi$ on the set $\mathcal{K}^n$ of all possible datasets over a binary attribute set $\mathcal{K}$, the single-target gain function $g_{\mathrm{T}}$ per Def. 13, and a full shuffle channel* $\mathsf{S}$ *per Def. 7, the corresponding posterior vulnerability is given by*

$$V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{S}] = V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{S}^r] = \frac{1}{2^n} \sum_{i=0}^{n} \binom{n}{i} \frac{\max(i, n-i)}{n}, \quad (6)$$

*where* $\mathsf{S}^r$ *is the reduced channel equivalent to* $\mathsf{S}$*, per Def. 8.*

For intuition, consider that the binary set of sensitive values is $\mathcal{K} = \{\mathtt{a}, \mathtt{b}\}$. The formula for (6) iterates through all histograms of size $n$ containing exactly $i$ individuals with value $\mathtt{a}$ (and, hence, $n-i$ individuals with value $\mathtt{b}$). Clearly, there are $\binom{n}{i}$ of such histograms. The adversary picks $\mathtt{a}$ as her guess if the number $i$ of $\mathtt{a}$'s in the histogram is at least as large as the number $n-i$ of $\mathtt{b}$'s, and picks $\mathtt{b}$ as her guess if $i < n-i$. Her probability of guessing correctly is the ratio between the maximum among $i$ and $n-i$, normalized by the total $n$ of individuals in the histogram. To obtain the final vulnerability, we take the expected success over all possible datasets, and under a uniform prior each dataset has probability $1/2^n$.

It is not trivial to immediately grasp the behavior of (6) as the size $n$ of the dataset grows. Moreover, the formula is not computationally efficient, as the number of binomials needed grows linearly with $n$. While computational efficiency could be improved by taking advantage of the symmetry of binomial coefficients, a remarkably simple equivalent formulation for (6) exists, using only a single binomial coefficient.

**Theorem 17** (Posterior single-target vulnerability under shuffle and a binary sensitive attribute: Fast formula)**.** *Given the same hypotheses as Prop. 16,*

$$V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{S}] = V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{S}^r] = \frac{1}{2} + \frac{1}{2^n} \binom{n-1}{\lfloor (n-1)/2 \rfloor} \quad (7)$$

Notice that this formulation is not only faster to compute, but easier to analyze. In particular, as $n$ grows, the second term in (7) goes to 0, indicating the entire sum goes to $1/2$. This makes intuitive sense: given a large enough dataset, we expect that the frequency of each binary value in any histogram approaches the adversary's prior on the target individual's sensitive value.(Theorem 17 is a case of Theorem 19 ahead.)

Finally, we consider the posterior vulnerability of the combined use of $k$-RR and shuffling, modeled by the cascade $\mathsf{NS}$. At a first glance, this vulnerability may appear more challenging to compute than that of shuffle alone, but in fact it only requires a simple modification.

**Proposition 18** (Posterior single-target vulnerability under $k$-RR and shuffle, and a binary sensitive attribute)**.** *Given the the uniform prior $\pi$ on the set $\mathcal{K}^n$ of all possible datasets over a binary attribute set $\mathcal{K}$, the single-target gain function $g_{\mathrm{T}}$ per Def. 13, a $k$-RR channel* $\mathsf{N}$ *per Def. 10, and a reduced shuffle channel* $\mathsf{S}$ *per Def. 8, the corresponding posterior vulnerability is given by*

$$V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{NS}] = V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{NS}^r] = \frac{1}{2^n} \sum_{i=0}^{n} \binom{n}{i} \times \frac{\max(i, n-i)p + \min(i, n-i)(1-p)}{n}, \quad (8)$$

*where* $\mathsf{S}^r$ *is the reduced channel equivalent to* $\mathsf{S}$*, per Def. 8.*

Similarly to Prop. 16, the adversary's best guess is to pick the most represented value, either $i$ $\mathtt{a}$'s or $n-i$ $\mathtt{b}$'s. With probability $p$, the target will have responded truthfully, and the adversary's guess will be correct proportionally to $n$. However, with probability $1-p$, the target will have flipped their response and ended up in the smaller group by chance, thereby still making the adversary's guess correct. This sum is then weighed by the probability of each dataset, which, again under a uniform prior, is $1/2^n$.

Here again, it may not be easy to immediately grasp from (8) the behavior of vulnerability as $n$ grows, specially because the $k$-RR parameter $p$ influences the final result. Fortunately, we discovered an equivalent formula.

**Theorem 19** (Posterior single-target vulnerability under $k$-RR and shuffle, and a binary sensitive attribute: Fast formula)**.** *Given the same hypotheses as those of Prop. 18,*

$$V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{NS}] = V_{g_{\mathrm{T}}}[\pi \triangleright \mathsf{NS}^r] = \frac{1}{2} + \frac{1}{2^n} \binom{n-1}{\lfloor (n-1)/2 \rfloor} (2p-1). \quad (9)$$

Notice that the difference between the formulation (7) for shuffle $\mathsf{S}$ alone and the formulation (9) for the cascade $\mathsf{NS}$ is that, in the former, the second term in the sum is scaled by a factor $2p-1$, which is always in the range $[0, 1]$ given that $p \in [1/2, 1]$.

### *B. Analyses of leakage behavior*

We now turn our attention to how the leakage formulas behave as the size $n$ of the dataset grows, and the parameter $p$ controlling the noise in the $k$-RR mechanism varies.

First, since the cascade $\mathsf{NS}$ can be understood as the post-processing of $\mathsf{N}$ by $\mathsf{S}$, by the data-processing-inequality (Sec. III), we know that $\mathsf{NS}$ can never cause more leakage than $\mathsf{N}$ alone. Since $\mathsf{NS}$ is commutative with $\mathsf{SN}$, per Prop. 11, we conclude that adding noise on top of shuffle, or shuffle on top of noise, can never increase the leakage of information. With our exact equations, we can isolate how the respective mechanisms affect posterior vulnerability.

Fig. 2 compares posterior single-target vulnerability for binary attributes under the shuffling channel $\mathsf{S}$, under the $k$-RR mechanism $\mathsf{N}$, and under the $k$-RR and shuffle cascade $\mathsf{NS}$, for various values of the $k$-RR parameter $p$.

Notice that the dotted yellow line represents a baseline of no security measures: when only $k$-RR $\mathsf{N}$ is applied using parameter $p = 1$, every participant reports their true value,

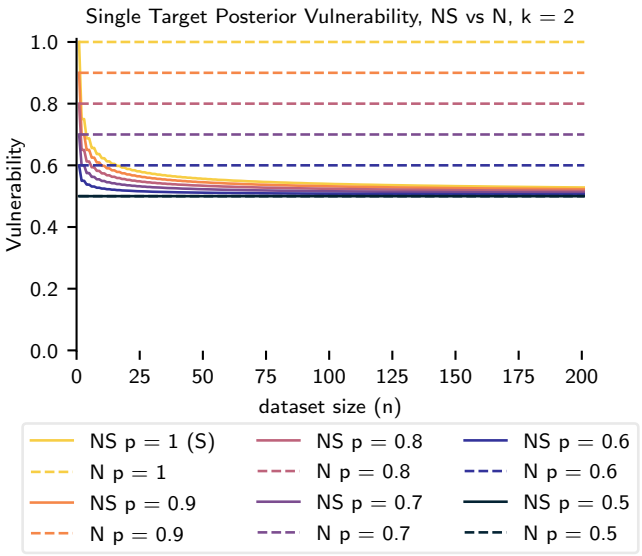


Fig. 2: Posterior single-target vulnerability for binary attributes under a uniform prior over various dataset sizes $n$. We consider the $k$-RR channel $\mathsf{N}$, the composition of $k$-RR & shuffle $\mathsf{NS}$, and the shuffle channel $\mathsf{S}$ (when $p = 1$).

and no shuffle is applied. This line represents an upper bound on the posterior vulnerability, since in the absence of any protective measure the adversary will know a target's value with certainty, and so the vulnerability is 1. However, when a shuffle is performed, as represented by the solid yellow line, the vulnerability drops immediately and dramatically, even if no noise is added. More precisely, since $p = 1$, this line represents exactly the vulnerability $V_{g_T}[\pi \triangleright \mathsf{S}]$.

The dotted lines represent the posterior vulnerability under a $k$-RR channel $\mathsf{N}$ for different values of $p$. We see that adding more noise improves the security of a single target's value. As $p$ decreases, users are less likely to report their true values, thereby decreasing the adversary's probability of guessing correctly. For example, if a user reports their true value with probability $p = 0.9$, the adversary will only guess the correct value $9/10$ of the time, represented by the dotted orange line. The vulnerability under $k$-RR is not affected by the dataset size, and thus remains constant for all $n$.

The remaining solid lines indicate the posterior vulnerability under the cascade $\mathsf{NS}$. We see here that shuffling noisy data exponentially decreases the vulnerability. At $n = 1$, the adversary observes only one value, and will know with certainty that it belongs to the target. When $p = 0.9$, she will guess the reported value and be correct $90\%$ of the time. But when $n = 2$, she could observe any of the following histograms: $\mathtt{a}{:}2, \mathtt{b}{:}0$, or $\mathtt{a}{:}1, \mathtt{b}{:}1$, or $\mathtt{a}{:}0, \mathtt{b}{:}2$. If she observes $\mathtt{a}{:}2, \mathtt{b}{:}0$, she will guess $\mathtt{a}\mathtt{a}$and be correct $9/10$ of the time; likewise if she observes $\mathtt{a}{:}0, \mathtt{b}{:}2$, she will guess $\mathtt{b}$. However, if she observes $\mathtt{a}{:}1, \mathtt{b}{:}1$ (which she will observe $1/2$ the time), either guess has an equal probability of being correct, decreasing the total vulnerability from $9/10$ to $7/10$. As $n$ grows, there are more possible outputs where the division of $n$ is less clear, bring the vulnerability closer to $1/2$. At $n = 200$, the posterior vulnerability of $\mathsf{NS}$ $p = 0.9$ is $0.5225$.

Note that the solid lines representing $\mathsf{NS}$ do not coincide at $n = 200$, although they will converge as $n$ grows. Concretely, given cascade $\mathsf{NS}$ at $n = 200$, the vulnerability when $p = 1$ is approximately $0.5282$ and the vulnerability when $p = 0.6$ is approximately $0.5056$.[5]

This graph shows that, for many values of $p$, shuffling alone is more effective in reducing vulnerability than simply adding noise. At $n = 200$, the posterior vulnerability of a shuffle with no noise represented by the solid yellow line is approximately $0.5286$, which means a shuffle alone has lower vulnerability than an application of noise where $p \geq 0.53$. This happens for most values of $p$. The only choice of $p$ that improves upon the security of a shuffle alone is when $p$ approaches $1/2$.

## V. Single-Target Leakage of a Binary Value Against a Strong Adversary

Thus far, our analysis has centered on specific adversary with limited prior knowledge, since they can often benefit the most from inference attacks. However, since differential privacy guarantees are crafted for the *strong adversary*, who a priori knows *everyone's* data but one, in this section we introduce a preliminary discussion about how vulnerability changes under this strong, *all-but-one (ABO)*, adversary.

Returning to Example 6, consider a dataset $n = 3$ and assume the adversary knows a priori that the last two people have $\mathtt{a}$'s. The set of secrets is reduced from $2^8$ to 2: $\mathtt{aaa}$ or $\mathtt{baa}$. How does observing the channel output affect the adversary's ability to guess the target's value?

Under shuffling alone, the adversary could observe histogram $\mathtt{a} : 2,\ \mathtt{b} : 1$ and know with certainty that $x_0 = \mathtt{b}$, or she could observe $\mathtt{a} : 3$ and know with certainty that $x_0 = \mathtt{a}$; the posterior ABO vulnerability is therefore 1. Under the $k$-RR channel, she will guess what she sees and she can be confident about this guess with probability $p$. Through QIF, we can illustrate how vulnerability can decrease when the two mechanisms are combined in $\mathsf{NS}$.

From [14], posterior vulnerability can be calculated by summing column maximums and scaling by the prior probability.

$$V_{ABO}[\pi \triangleright \mathsf{NS}^r] = \frac{1}{2} \sum_y \max_{\substack{x_0 = \mathtt{a}, \\ x_0 = \mathtt{b}}} \mathsf{NS}^r_{x,y} \tag{10}$$

Figure 3 shows the posterior ABO vulnerability when $n = 201$, over different sets of adversary prior knowledge. The first $0\%$ tick represents an adversary who knows the last two hundred people have $\mathtt{b}$'s, and who is guessing the value of the first person. Given this knowledge, when $p = 0.8$, the ABO vulnerability is approx. $0.52111$. Changing the prior distribution affects the ABO vulnerability, but only marginally. For instance, when the adversary knows that one hundred people have $\mathtt{a}$'s and the other hundred have $\mathtt{b}$'s (represented

[5]Also note that the step-like nature of the line representing $\mathsf{NS}$ is not a graphing artifact, but a reflection of the equation itself: every two consecutive values of $n$ have the same vulnerability because of the floor function.

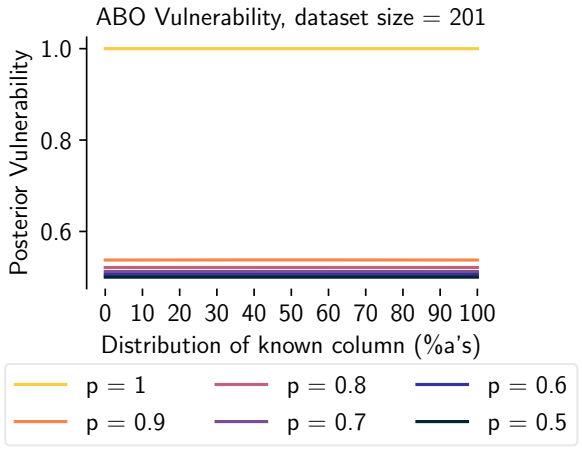


Fig. 3: All-But-One posterior vulnerability when the dataset size $n = 201$ and the adversary knows 200 values a priori.

by the $50\%$ tick), the vulnerability increases a bit, but only to approx. 0.52116.

From the graph, we can confirm that the shuffle alone (represented by the yellow line when $p = 1$) does not affect vulnerability—the adversary can guess the last person with perfect confidence. Recall that the opposite effect was observed in Figure 2 where the shuffle alone dramatically reduced vulnerability; this speaks to how adversary choice influences our understanding of a given privacy mechanism. However, when noise is introduced, the vulnerability decreases *below* $p$, indicating that the combination of noise and shuffling provides the most security. A formal investigation of this phenomena is set for future work.

## VI. Single-target leakage of a generic value $k{\geq}2$ against an uninformed adversary

In this section, we extend our investigation of Sec. IV to attack scenarios in which: (i) as before, the adversary is attempting to guess the secret value of a single individual from $\mathcal{N}$ chosen as the target in the dataset; but (ii) contrarily to before, the set $\mathcal{K}$ of sensitive values can have any size $k{\geq}2$. Here again, we focus on an uninformed adversary, with a uniform prior on datasets.

For this general case, we develop exact equations with which to calculate single-target posterior vulnerability. However, these equations are not as computationally efficient or as easy to analyze as the binary case. Nevertheless, we are able to provide asymptotic bounds on leakage by uncovering a surprising connection between our scenario and a well-known combinatorial problem.

### A. Leakage formulas

The prior vulnerability remains $1/k$, the same as that of the binary case, per Prop. 14. Indeed, given a uniform distribution, an adversary will guess any of the $k$ values and will have an equal probability of being correct.

For posterior vulnerability, we investigate the same combinations of sanitization mechanisms as before: the $k$-RR channel $\mathsf{N}$, the full shuffle channel $\mathsf{S}$, and the composition of $k$-RR with shuffle via the cascade $\mathsf{NS}$. The posterior vulnerability of the $k$-RR channel $\mathsf{N}$ has already been derived in Prop. 15 for any $k \geq 2$, and it is simply $p$. However, the vulnerability of the shuffle channel $\mathsf{S}$ and consequently of the cascade $\mathsf{NS}$ depends on $k$, necessitating a generalization of the earlier results.

Recall that under shuffling, the exact mapping from individuals to values is lost, and only the dataset's histogram is preserved. In the binary case, the posterior vulnerability of shuffling was computed by iterating through all histograms of size two, which could be counted using a summation of binomial coefficients. In the general case, the adversary observes multi-sets, which must be counted using multinomial coefficients, as formalized below.

**Proposition 20** (Posterior single-target vulnerability under shuffle and $k \geq 2$)**.** *Given the same hypotheses as Prop. 16, except that the sensitive set $\mathcal{K}$ can have $k \geq 2$ elements,*

$$V_{g_T}[\pi \triangleright \mathsf{S}] = V_{g_T}[\pi \triangleright \mathsf{S}^r] = \frac{1}{k^n} \sum_{\substack{n_1,\ldots,n_k: \\ n_1+\cdots+n_k=n}} \binom{n}{n_1,\ldots,n_k} \frac{n^*}{n}, \tag{11}$$

*where* $n^* = \max(n_1, \ldots, n_k)$.

Finally, the vulnerability of the combination of $k$-RR and shuffle is an extension of the binary case from Prop. 19.

**Proposition 21** (Posterior single-target vulnerability under $k$-RR & shuffle, $k \geq 2$)**.** *Given the same hypotheses as Prop. 18, except that the sensitive set $\mathcal{K}$ can have $k \geq 2$ elements,*

$$V_{g_T}[\pi \triangleright \mathsf{NS}] = V_{g_T}[\pi \triangleright \mathsf{NS}^r] = \frac{1}{k^n} \sum_{\substack{n_1,\ldots,n_k: \\ n_1+\cdots+n_k=n}} \binom{n}{n_1,\ldots,n_k} \frac{n^*p + (n-n^*)^{(1-p)/(k-1)}}{n} \tag{12}$$

*where* $n^* = \max(n_1, \ldots, n_k)$.

Notice, however, that the formulas given in Prop. 20 and Prop. 21, although exact, are not easy to compute or interpret. Indeed, they depend on the computation of a number of binomial coefficients that grows linearly with the dataset size $n$. In Sec. VI-C, we shall discuss asymptotic bounds for these vulnerabilities, but first we analyze the exact behavior of leakage for tractable values of $n$, $k$, and $p$.

### B. Analyses of leakage behavior

Fig. 4 compares the exact values for the posterior vulnerabilities of the three channels $\mathsf{N}$, $\mathsf{S}$, and $\mathsf{NS}$ for a set $\mathcal{K}$ of sensitive values with $k = 5$ elements.

As before, the dotted yellow line represents the case of no noise nor shuffle; here, the adversary can observe and know the target's value exactly. By observing the vulnerability under the shuffle channel $\mathsf{S}$, represented by the solid yellow line, we can see that shuffle alone decreases the adversary's probability

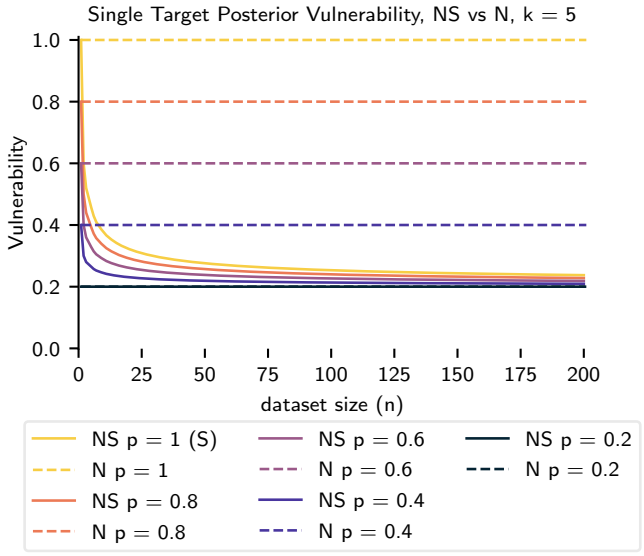


Fig. 4: Posterior single-target vulnerability for $k = 5$ values, under a uniform prior over various dataset sizes $n$. We consider the $k$-RR channel $\mathsf{N}$, the composition of $k$-RR & shuffle $\mathsf{NS}$, and the shuffle channel $\mathsf{S}$ ($p = 1$).

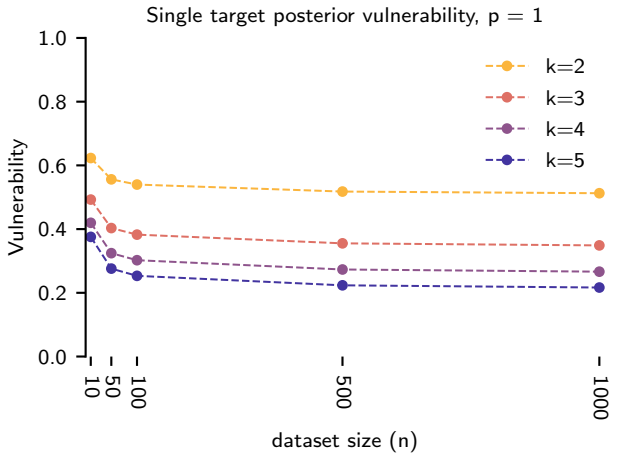


Fig. 5: Posterior single-target vulnerability of the shuffle channel $\mathsf{S}$ under a uniform prior, for varying attribute set sizes $k$ and dataset sizes $n$.

of success considerably. The dotted lines represent the vulnerability under the $k$-RR channel $\mathsf{N}$ for different values of $p$, while the solid lines represent posterior vulnerability under the $\mathsf{NS}$ cascade. For example, when $p = 0.8$, represented by the dotted orange line, the posterior vulnerability drops to $0.8$, regardless of the dataset size. When shuffle is added, the vulnerability drops exponentially with $n$, represented by the solid orange line, asymptotically approaching the lower bound of $0.2 = 1/k$ (the prior vulnerability). The graph suggests that shuffle alone is a highly effective sanitization mechanism, significantly outperforming $k$-RR's leakage guarantees in most cases. Indeed, the only cases in which $k$-RR alone provides a sanitization comparable to that provided by shuffle alone are when the parameter $p$ approaches $1/k$. However, these are exactly the cases in which the mechanisms are noisiest, so their utility suffers the most.

Fig. 5 provides more insight into how shuffle alone affects posterior vulnerability. As exact values are computationally expensive to calculate as $n$ and $k$ increase, here we show the exact vulnerability for a subset of dataset sizes $n$ ranging from 10 to 1000. We see that posterior vulnerability decreases with $n$, and approaches the prior vulnerability of $1/k$. For the sake of comparison, at $n = 100$ the posterior vulnerability when $k = 3$ is $0.3826$, and at $n = 1000$ it is $0.3488$.

### C. Asymptotic Bounds

Interestingly, (11) has a relevant combinatorial interpretation when scaled up by $n$, which can be written explicitly as

$$V_{g_T}[\pi \triangleright \mathsf{S}] \times n = \frac{1}{k^n} \sum_{\substack{n_1,n_2,\ldots,n_k: \\ n_1+n_2+\cdots+n_k=n}} \binom{n}{n_1, n_2, \ldots, n_k} n^*. \tag{13}$$

Intuitively, when $n$ balls are thrown uniformly at random and independently into $k$ distinct bins, (13) gives the exact number of expected balls in the fullest bin (also referred to as "maximum load"). This perspective provides an interpretation of (11): For every division of $n$ into $k$ parts, this equation counts the ways these parts can be labeled, chooses the largest part, then takes the average by dividing by the total number of the possibilities. Brown's work on surmising remixed keys [53] posits an equivalent formula for Equ. 13, however he too was unable to find a faster way to compute these values exactly.

There are well-known asymptotic bounds for this combinatorial problem [54]. In particular, when $n \geq k \ln k$,

$$V_{g_T}[\pi \triangleright \mathsf{S}] \times n = \frac{n}{k} + \Theta\left(\sqrt{\frac{n \ln k}{k}}\right) \quad \text{(maximum load)} \tag{14}$$

This holds with high probability, meaning an event $M$ counting the maximum number of $n$ balls in $k$ bins occurs with probability $\Pr(M) \geq 1 - k^{-c}$ for an arbitrarily chosen constant $c \geq 0$ [55]. Dividing both sides by $n$, we get asymptotic bounds for posterior single-target vulnerability.

$$V_{g_T}[\pi \triangleright \mathsf{S}] = \frac{1}{k} + \Theta\left(\sqrt{\frac{\ln k}{kn}}\right) \tag{15}$$

These bounds show the maximum load grows sublinearly with $n$ while posterior vulnerability decreases monotonically.

While the big-theta implies the existence of a constant factor $f$, we can test how well this equation matches the true posterior vulnerability when $f = 1$. Fig. 6 shows how well the asymptotic bounds for posterior single-target vulnerability under the shuffle channel approximate the exact value. The yellow line, representing the exact vulnerability when $k = 3$, has a slight offset from the approximation, but the violet and blue lines, representing $k = 4$ and $k = 5$ respectively, are visually indistinguishable from the value provided by the asymptotic bounds.

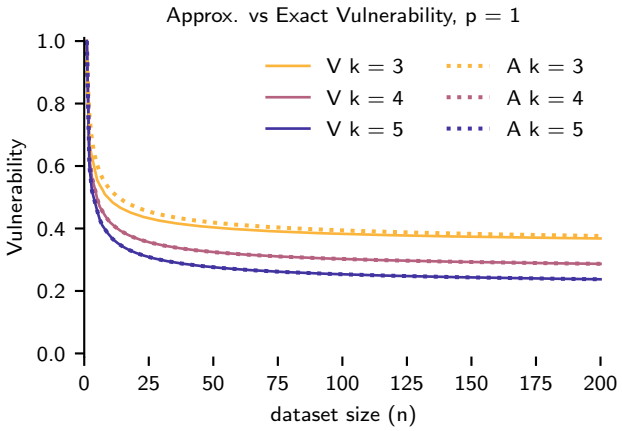


Fig. 6: Comparison of the approximation with the true value of posterior single-target vulnerability through the shuffle channel for different values of $k$.

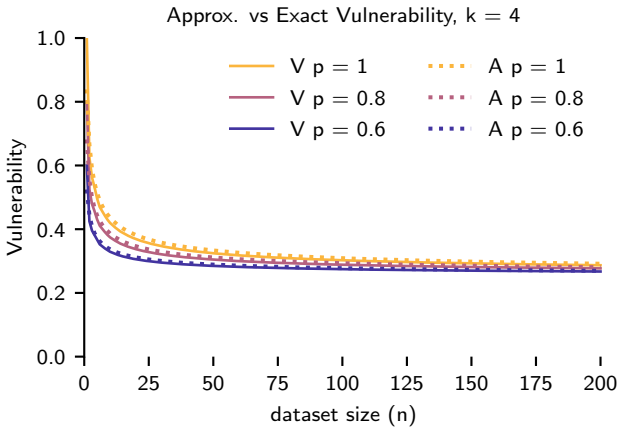


Fig. 7: Comparison of the approximation with the true value of posterior single-target vulnerability through the shuffle and $k$-RR channels when $k = 4$ for different values of $p$.

Given an approximation for $V_{g_T}[\pi \triangleright \mathsf{S}]$, we can derive an approximation for $V_{g_T}[\pi \triangleright \mathsf{NS}]$.

**Theorem 22.** *Posterior single-target vulnerability for a general $k$ has the following asymptotic bounds.*

$$V_{g_T}[\pi \triangleright \mathsf{NS}] = \frac{1}{k} + \Theta\left(\sqrt{\frac{\ln k}{kn}}\right)\left(\frac{kp-1}{k-1}\right) \qquad (16)$$

The proof relies on the fact that the equation for posterior vulnerability under NS can be re-written as a function of the posterior vulnerability under S. Explicitly,

$$V_{g_T}[\pi \triangleright \mathsf{NS}] = V_{g_T}[\pi \triangleright \mathsf{S}] \times \left(\frac{kp-1}{k-1}\right) + \frac{1-p}{k-1}. \qquad (17)$$

Fig. 7 sets $k = 4$ and compares the exact posterior single-target vulnerability with the asymptotic bounds assuming $f = 1$. When $p = 1$, we see the dotted yellow line representing the bounds is slightly above the straight yellow line representing the exact value for small values of $n$. As $n$ increases, the two lines converge. This is seen again for $p = 0.8$ and $p = 0.6$. From this anecdotal test, it seems the asymptotic bound closely approximates the true value when $f = 1$.

## VII. Discussion

The main lessons learned from this work are the following:

- The shuffle and the $k$-RR mechanisms commute, in the sense that their composition as probabilistic functions commute. We believe that this is the case for the composition of the shuffle with *any* local obfuscation mechanism.
- By using the formulas derived for vulnerabilities and leakage, we have shown that under the uninformed adversary, the shuffling accounts for most of the privacy, as by itself it achieves almost the same level of resilience to inference attacks as its composition with $k$-RR. (The only exception is when the probability of reporting the true value is $p = 1/k$, but this case has no utility.) We also noted that the level of posterior vulnerability decreases very fast with the number of participants, and that it converges to the prior, which is $1/k$, i.e., the probability of guessing the true value by making a random guess.
- We have shown how to model the strong adversary in our framework, and how to calculate the posterior vulnerability. Our experiments show that, under the strong adversary, shuffling alone is ineffective, but it substantially increases privacy when combined with $k$-RR. This reinforces the results from the shuffle model literature pointing out the merits of shuffling, such as [27], although generally those works consider adversaries more powerful than the strong adversary of DP.
- As a consequence, we point out that, for achieving the best trade-off between privacy and utility, in the strong model it is better to compose $k$-RR and shuffle. In contrast, under the uninformed adversary, it may be better to use the shuffle alone, as $k$-RR reduces utility and does not have a significant impact on privacy.
- Our results can help to choose which mechanism(s) (for instance, the shuffle model alone or the shuffle combined with $k$-RR) gives the best trade-off between privacy and utility. Indeed, by using the formulas derived for the posterior vulnerability, we can compute the level of privacy (i.e., resilience to inference attacks) of the mechanisms and compare them. This process may also require to tune $\epsilon$ to achieve the desired utility. We remark that different adversary models may lead to different results in this comparison, as shown in the introduction. We argue that if adversarial knowledge is not known, it is generally better to choose the more likely one. In the case of DP, a basic assumption is that the data consumer cannot access directly the database and can only query it via an interface. Hence, the uninformed adversary is more natural and likely than the strong one.

## VIII. Conclusion

In this work, we proved that $k$-RR and shuffling can commute without affecting information leakage, and derived exact formulas for prior and posterior vulnerabilities for an uninformed adversary focusing on a single target. For binary attributes, these formulas are computationally efficient, and for the general case we found asymptotic bounds. We studied how the leakage of shuffling and $k$-RR mechanisms behaves as the dataset size increases for different privacy parameters. For the uninformed adversary, we found that shuffling alone may dramatically reduce leakage in many cases, outperforming the application of $k$-RR alone.

## Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments. Mireya Jurado was supported by the U.S. Department of Homeland Security under Grant Award № 2017-ST-062-000002. Catuscia Palamidessi was supported by the European Research Council (ERC) under the Horizon 2020 research and innovation programme, grant agreement № 835294. Ramon G. Gonze and Mário S. Alvim were supported by CNPq, CAPES, and FAPEMIG. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

## References

[1] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. of EUROCRYPT*, ser. LNCS, vol. 4004. Springer, 2006, pp. 486–503.

[2] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *In Proc. of TCC*, ser. LNCS, vol. 3876. Springer, 2006, pp. 265–284.

[3] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith, "What can we learn privately?" *SIAM Journal of Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[4] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. of FOCS*. IEEE Computer Society, 2013, pp. 429–438.

[5] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proc. of ACM SIGSAC CCS*. ACM, 2014, pp. 1054–1067.

[6] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *PoPETS*, vol. 2016, no. 3, pp. 41–61, 2016.

[7] Apple Differential Privacy Team, "Learning with privacy at scale," *Apple Machine Learning Journal*, vol. 1, no. 9, December 2017.

[8] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudinger, V. V. Prakash, A. Legendre, and S. Duplinsky, "Emoji frequency detection and deep link frequency," US Patent 9,705,908., July 11 2017.

[9] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. of NeurIPS*, ser. NIPS'17. Curran Associates Inc., 2017, pp. 3574–3583.

[10] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *Advances in Cryptology – CRYPTO 2008*. Springer Berlin Heidelberg, 2008, pp. 451–468.

[11] T.-H. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *Proc. of the 20th ESA*, ser. ESA'12. Springer-Verlag, 2012, pp. 277–288.

[12] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld, "Prochlo: Strong privacy for analytics in the crowd," in *Proc. of SOSP*, 2017, pp. 441–459.

[13] A. Cheu, A. D. Smith, J. R. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Proc. of EUROCRYPT*, ser. LNCS, vol. 11476. Springer, 2019, pp. 375–403.

[14] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, *The Science of Quantitative Information Flow*, ser. Information Security and Cryptography. Springer Int. Publishing, 2020.

[15] M. Jurado, C. Palamidessi, and G. Smith, "A formal information-theoretic leakage analysis of order-revealing encryption," in *CSF*, 2021, pp. 1–16.

[16] N. Fernandes, M. Dras, and A. McIver, "Processing text for privacy: an information flow perspective," in *FM*, 2018, pp. 3–21.

[17] M. S. Alvim, N. Fernandes, A. McIver, C. Morgan, and G. H. Nunes, "Flexible and scalable privacy assessment for very large datasets, with an application to official governmental microdata," *PoPETS*, vol. 2022, pp. 378–399, 2022.

[18] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "On the information leakage of differentially-private mechanisms," *J. of Comp. Security*, vol. 23, no. 4, pp. 427–469, 2015.

[19] K. Chatzikokolakis, N. Fernandes, and C. Palamidessi, "Comparing systems: Max-case refinement orders and application to differential privacy," in *CSF*. IEEE, 2019, pp. 442–457.

[20] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Proc. of CSF*, 2012, pp. 265–279.

[21] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 492–542, Jan. 2016.

[22] M. C. Tschantz, S. Sen, and A. Datta, "Sok: Differential privacy as a causal property," in *2020 IEEE S&P*, 2020, pp. 354–371.

[23] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *Proc. of ACM SIGSAC CCS*. ACM, 2016, p. 43–54.

[24] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in *Proc. of the 2015 ACM SIGMOD Int. Conf. on Management of Data*, ser. SIGMOD '15. ACM, 2015, p. 747–762.

[25] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: A unifying framework for privacy definitions," in *Proc. of ACM SIGSAC CCS*, ser. CCS '13. ACM, 2013, p. 889–900.

[26] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[27] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *CRYPTO*, 2019, pp. 638–667.

[28] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, S. Song, K. Talwar, and A. Thakurta, "Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation," *arXiv:2001.03618*, 2020.

[29] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proc. of SODA*, 2019, pp. 2468–2479.

[30] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *Proc. of FOCS*. IEEE, 2021, pp. 954–964.

[31] A. Koskela, M. A. Heikkilä, and A. Honkela, "Tight accounting in the shuffle model of differential privacy," *arXiv:2106.00477*, 2021.

[32] A. Koskela, J. Jälkö, L. Prediger, and A. Honkela, "Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft," in *Proc. of AISTATS*, 2021, pp. 3358–3366.

[33] B. Balle, J. Bell, A. Gascon, and K. Nissim, "Differentially private summation with multi-message shuffling," *arXiv:1906.09116*, 2019.

[34] B. Balle, J. Bell, A. Gascón, and K. Nissim, "Private summation in the multi-message shuffle model," in *Proc. of ACM SIGSAC CCS*, 2020, pp. 657–676.

[35] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Cryptography from anonymity," in *Proc. of FOCS*, 2006, pp. 239–248.

[36] B. Ghazi, R. Pagh, and A. Velingker, "Scalable and differentially private distributed aggregation in the shuffled model," *arXiv:1906.08320*, 2019.

[37] B. Balle, J. Bell, A. Gascón, and K. Nissim, "Improved summation from shuffling," *arXiv:1909.11225*, 2019.

[38] V. Balcer and A. Cheu, "Separating local & shuffled differential privacy via histograms," *arXiv:1911.06879*, 2019.

[39] A. Cheu and M. Zhilyaev, "Differentially private histograms in the shuffle model from fake users," *arXiv:2104.02739*, 2021.

[40] V. Balcer, A. Cheu, M. Joseph, and J. Mao, "Connecting robust shuffle privacy and pan-privacy," in *Proc. of SODA*, 2021, pp. 2384–2403.

[41] A. Cheu, "Differential privacy in the shuffle model: A survey of separations," *arXiv:2107.11839*, 2021.
[42] A. Cheu and J. Ullman, "The limits of pan privacy and shuffle privacy for learning and estimation," in *Proc. of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 1081–1094.
[43] G. Barthe and B. Köpf, "Information-theoretic bounds for differentially private mechanisms," in *Proc. of CSF*. IEEE Computer Society, 2011, pp. 191–204.
[44] K. Chatzikokolakis, N. Fernandes, and C. Palamidessi, "Refinement orders for quantitative information flow and differential privacy," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 40–77, 2021.
[45] M. Jurado, R. G. Gonze, M. S. Alvim, and C. Palamidessi, "Analyzing the shuffle model through the lens of quantitative information flow," *arXiv:2305.13075*, 2023.
[46] D. Clark, S. Hunt, and P. Malacaria, "Quantitative analysis of the leakage of confidential data," *Electron. Notes Theor. Comput. Sci.*, vol. 59, no. 3, pp. 238–251, 2001.
[47] G. Smith, "On the Foundations of Quantitative Information Flow," in *FOSSACS*, ser. LNCS, vol. 5504. Springer, 2009, pp. 288–302.
[48] A. McIver, L. Meinicke, and C. Morgan, "Compositional closure for Bayes Risk in probabilistic noninterference," in *ICALP*, vol. 6199. Springer Verlag, 2010, pp. 223–235.
[49] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Axioms for information leakage," in *Proc. of CSF*, 2016, pp. 77–92.
[50] Y. Wang, X. Wu, and D. Hu, "Using randomized response for differential privacy preserving data collection." in *EDBT/ICDT Workshops*, vol. 1558, 2016, pp. 0090–6778.
[51] C. Braun, K. Chatzikokolakis, and C. Palamidessi, "Quantitative notions of leakage for one-try attacks," in *Proc. MFPS*, ser. ENTCS, vol. 249. Elsevier, 2009, pp. 75–91.
[52] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Additive and multiplicative notions of leakage, and their capacities," in *IEEE CSF*. IEEE Computer Society, 2014, pp. 308–322.
[53] D. R. L. Brown, "Bounds on surmising remixed keys," *IACR Cryptol. ePrint Arch.*, p. 375, 2015.
[54] M. Raab and A. Steger, ""balls into bins" — a simple and tight analysis," in *Randomization and Approximation Techniques in Computer Science*. Springer Berlin Heidelberg, 1998, pp. 159–170.
[55] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, "Balanced allocations: The heavily loaded case," *SIAM J. Comput.*, vol. 35, no. 6, p. 1350–1385, jun 2006.

## APPENDIX

Here we complement the discussion on reduced mechanisms, presented in Sec. III-B, by introducing a reduced version of the $k$-RR mechanism and proving its compositional properties with the reduced shuffling mechanism.

*a) Reduced channel for the $k$-RR mechanism:* For completeness, we can also consider a reduced version of the $k$-RR mechanism that operates by taking a histogram as input and producing a histogram over randomized values as output.

We want to define the reduced channel $k$-RR operating directly over histograms so that its final effect is the same as the expected effect of a full $k$-RR channel operating over all concrete datasets with the same histogram as the input histogram, weighted by a uniform prior distribution on datasets.

Formally, let $Pr(x, y, z_1, z_2)$ denote the joint probability over datasets $x, y \in \mathcal{K}^n$ and histograms $z_1, z_2$ over $\mathcal{K}$. We will derive the reduced $k$-RR channel $\mathsf{N}^r$ mapping histograms to histograms by imposing restrictions on this joint probability. We start by deriving, for all histograms $z_1, z_2$ over $\mathcal{K}$:

$$\begin{aligned}
&\mathsf{N}^r_{z_1,z_2} = \text{(def. of channel)}\\
&Pr(z_2 \mid z_1) = \text{(marginalization)}\\
&\textstyle\sum_{\substack{x\in\mathcal{K}^n\\ y\in\mathcal{K}^n}} Pr(x, y, z_2 \mid z_1) = \text{(chain rule)}\\
&\textstyle\sum_{\substack{x\in\mathcal{K}^n\\ y\in\mathcal{K}^n}} Pr(x \mid z_1) Pr(y \mid x, z_1) Pr(z_2 \mid x, y, z_1) \qquad (18)
\end{aligned}$$

Now, we impose the restriction that $Pr(y \mid x, z_1)$ must be exactly the probability $\mathsf{N}_{x,y}$ that the corresponding full channel $\mathsf{N}$ operating on a dataset $x$ would produce dataset $y$ as output. We also impose that $z_2$ must be the histogram of dataset $y$, so $Pr(z_2 \mid x, y, z_1)$ must be 1 exactly when $h(y) = z_2$, and 0 otherwise. By applying that to (18), we get:

$$\mathsf{N}^r_{z_1,z_2} = \sum_{y\in\mathcal{K}^n : h(y)=z_2} \sum_{x\in\mathcal{K}^n} Pr(x \mid z_1)\mathsf{N}_{x,y} \qquad (19)$$

Now let us denote by $\pi_{x|z_1}$ the conditional probability $Pr(x \mid z_1)$ that the input dataset is $x$ given that its histogram must be $z_1$. Considering that the distribution on input datasets is uniform, as per (5), we then derive:

$$\pi_{x|z_1} = Pr(x \mid z_1) = \begin{cases} 1/\#h(x), & \text{if } h(x) = z_1,\\ 0, & \text{otherwise.}\end{cases} \qquad (20)$$

Finally, by substituting (20) in (19), we reach the definition of a reduced shuffling channel $\mathsf{N}^r$ below.

**Definition 23** (Reduced $k$-RR channel)**.** *Given a uniform prior $\pi$ on all possible original datasets $x \in \mathcal{K}^n$, a* reduced $k$-RR channel $\mathsf{N}^r$ *(again, for "noise") is a channel from histograms on $\mathcal{K}$ to histograms on $\mathcal{K}$ s.t., for every input histogram $z_1$ and output histogram $z_2$,*

$$\mathsf{N}^r_{z_1,z_2} = \sum_{\substack{x\in\mathcal{K}^n : h(x)=z_1\\ y\in\mathcal{K}^n : h(y)=z_2}} \frac{1}{\#h(x)} \mathsf{N}_{x,y}.$$

**Example 24** (Reduced $k$-RR channel)**.** *Table IV contains the channel $\mathsf{N}^r$ representing the application of a reduced $k$-RR mechanism to the scenario of Example 6.*

| $\mathsf{N}^r$ | a:3, b:0 | a:2, b:1 | a:1, b:2 | a:0, b:3 |
|---|---|---|---|---|
| a:3, b:0 | $p^3$ | $3p^2\overline{p}$ | $3p\overline{p}^2$ | $\overline{p}^3$ |
| a:2, b:1 | $p^2\overline{p}$ | $(p^3+2p\overline{p}^2)/3$ | $(2p^2\overline{p}+\overline{p}^3)/3$ | $p\overline{p}^2$ |
| a:1, b:2 | $p\overline{p}^2$ | $(2p^2\overline{p}+\overline{p}^3)/3$ | $(p^3+2p\overline{p}^2)/3$ | $p^2\overline{p}$ |
| a:0, b:3 | $\overline{p}^3$ | $3p\overline{p}^2$ | $3p^2\overline{p}$ | $p^3$ |

TABLE IV: Reduced $k$-RR channel $\mathsf{N}^r$ for Example 24, with $k$=2 possible sensitive values and $n$=3 individuals. The channel receives histograms as inputs and produces histograms as outputs.

Unlike the shuffling channels, the full $k$-RR channel $\mathsf{N}$ and its reduced counterpart $\mathsf{N}^r$ are not equivalent. Indeed, they are not even comparable since they do not have the same input type.

**Proposition 25** (Non-equivalence between full and reduced $k$-RR)**.** *Let $\mathsf{N}$ be a full $k$-RR channel as per Def. 10, and $\mathsf{N}^r$ be the reduced $k$-RR channel obtained from $\mathsf{N}$ as per Def. 23. Then*

$$\mathsf{N} \not\equiv \mathsf{N}^r.$$

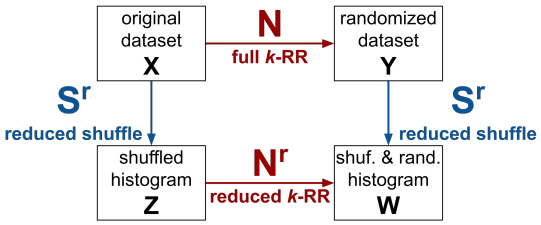


Fig. 8: Commutativity in the reduced case: $\mathsf{NS}^r = \mathsf{S}^r\mathsf{N}^r$. Channel $\mathsf{N}$ maps datasets to datasets, channel $\mathsf{S}^r$ maps datasets to histograms, and channel $\mathsf{N}^r$ maps histograms to histograms.

*b) Compositions of reduced mechanisms:* As with their full counterparts (mapping datasets to datasets), the reduced sanitization mechanisms (operating over histograms) can be applied to an input dataset either in isolation or in combination.

The situation regarding reduced mechanisms, however, is more subtle. Notice that we are allowed to cascade a reduced shuffling channel $\mathsf{S}^r$ with a reduced $k$-RR channel $\mathsf{N}^r$, since the output of the former and the input of the latter are both histograms.But we cannot start our sanitization process by applying a reduced $k$-RR channel $\mathsf{N}^r$ to a dataset, since this channel receives as input only histograms. However, a meaningful form of commutativity between shuffling and $k$-RR holds if we are careful with the types of the channels in the cascade, as depicted in Fig. 8. That is formalized in the following result.

**Proposition 26** (Commutativity of $k$-RR and shuffling: Reduced case)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ values for the sensitive attribute. Let also $\mathsf{N}$ be a full $k$-RR channel as per Definition 10, $\mathsf{N}^r$ be a reduced $k$-RR channel as per Definition 23, and $\mathsf{S}^r$ be a reduced shuffling channel per Definition 8. Then*

$$\mathsf{NS}^r = \mathsf{S}^r\mathsf{N}^r.$$

**Example 27** (Cascading of $k$-RR and shuffling in the reduced case)**.** *Consider Example 6. The channel $\mathsf{NS}^r$ representing the application of a full $k$-RR mechanism followed by reduced shuffling and the channel $\mathsf{S}^r\mathsf{N}^r$ representing the application of reduced shuffling followed by a reduced $k$-RR mechanism are identical, as represented in Table V.*

Finally, we provide the following result, stating that the composed mechanisms of shuffling and $k$-RR in their full and reduced forms are equivalent, in the sense that, for every $g$-vulnerability measure and prior distribution on secret values, both yield the same quantification of information leakage.

**Proposition 28** (Equivalence of full and reduced compositions)**.** *Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ be a set of $n \geq 1$ individuals and $\mathcal{K}$ be a set of $k \geq 2$ values for the sensitive attribute. Let also $\mathsf{N}$ be a full $k$-RR channel as per Definition 10, $\mathsf{N}^r$ be a reduced $k$-RR channel as per Definition 23, and $\mathsf{S}^r$ be a reduced shuffling channel per Definition 8. Then:*

$$\mathsf{SN} \equiv \mathsf{S}^r\mathsf{N}^r \tag{21}$$

| $\mathsf{NS}^r/\mathsf{S}^r\mathsf{N}^r$ | (a:3,b:0) | (a:2,b:1) | (a:1,b:2) | (a:0,b:3) |
|---|---|---|---|---|
| aaa | $p^3$ | $3p^2\overline{p}$ | $3p\overline{p}^2$ | $\overline{p}^3$ |
| aab | $p^2\overline{p}$ | $p^3 + 2p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p\overline{p}^2$ |
| aba | $p^2\overline{p}$ | $p^3 + 2p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p\overline{p}^2$ |
| baa | $p^2\overline{p}$ | $p^3 + 2p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p\overline{p}^2$ |
| abb | $p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p^3 + 2p\overline{p}^2$ | $p^2\overline{p}$ |
| bab | $p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p^3 + 2p\overline{p}^2$ | $p^2\overline{p}$ |
| bba | $p\overline{p}^2$ | $2p^2\overline{p} + \overline{p}^3$ | $p^3 + 2p\overline{p}^2$ | $p^2\overline{p}$ |
| bbb | $\overline{p}^3$ | $3p\overline{p}^2$ | $3p^2\overline{p}$ | $p^3$ |

TABLE V: Channel for Example 27, representing both (1) the cascading $\mathsf{NS}^r$ of full $k$-RR followed by reduced shuffling and (2) the cascading $\mathsf{S}^r\mathsf{N}^r$ of reduced shuffling followed by reduced $k$-RR, with $k$=2 possible sensitive values and $n$=3 individuals. Here $p$ is the probability a user responds with their true value, and $\overline{p}$=1−$p$. The cells are colored according to the entries' values when $p$=0.75 and $\overline{p}$=0.25. Here $p^3 + 2p\overline{p}^2$ represents the largest value at approximately 0.5156 while $\overline{p}^3$ is the smallest at approximately 0.0156.

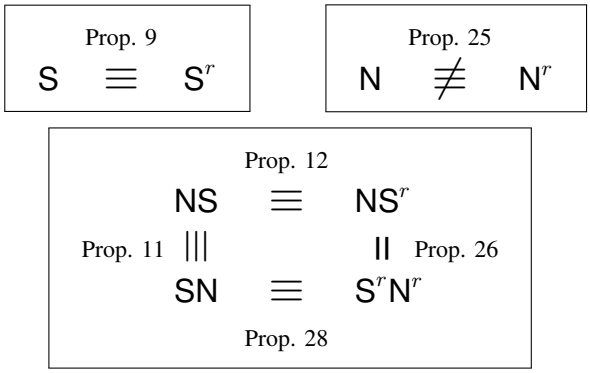


Fig. 9: All relationships among compositions of full and reduced $k$-RR and shuffling, extending the results from Fig. 1. Here $(=)$ denotes syntactic equality, whereas $(\equiv)$ denotes equivalence w.r.t. information leakage.

Fig. 9 extends Fig. 1 and summarizes all relationships among compositions of the full and reduced versions of both the $k$-RR mechanism and of shuffling.