Reasoning About Privacy Using Axioms

Bing-Rong Lin
Dept. of Computer Science & Engineering
Penn State University
University Park, PA 16802

Daniel Kifer
Dept. of Computer Science & Engineering
Penn State University
University Park, PA 16802

***Abstract*—In statistical privacy, privacy definitions are contracts that guide the behavior of algorithms that take in sensitive data and produce *sanitized* data. Historically, data privacy breaches have been the result of fundamental misunderstandings about what a particular privacy definition guarantees.**

Privacy definitions are often analyzed using a hit-or-miss approach: a *specific* attack strategy is evaluated to determine if a *specific* type of information can be inferred. If the attack works, the privacy definition is known to be too weak. If it doesn't work, little information is gained. Furthermore, these strategies will not identify cases where a privacy definition protects unnecessary pieces of information.

A systematic analysis of privacy definitions is a long-standing open problem. In this paper, we present initial steps towards a solution. Using privacy axioms, we identify two mathematical objects that are associated with privacy definitions – the *consistent closure* and the *row cone* (which is constructed from the consistent closure). The row cone is a geometric object which neatly encapsulates Bayesian guarantees provided by a privacy definition.

We apply these ideas to the study of randomized response to show that it provides unnecessarily strong protections on the parity of a dataset.

I. Introduction

Data collection and analysis help drive innovation in business and science. In many cases, it is beneficial to share the data (e.g., to crowd-source the analysis [1], to enable collaborations, etc.). The data often contain sensitive information and so the straightforward sharing of data is not possible. Instead, the original data are passed to a *sanitizing* algorithm which then outputs *sanitized* data. If the sanitizing algorithm is well-designed, it should be safe to release the sanitized data.

The design of a sanitizing algorithm is governed by a *privacy definition* which acts like a mathematical contract – if the behavior of the algorithm satisfies its prespecified constraints, then certain types of sensitive inference are blocked. As is the case with legal contracts, privacy definitions are often subtle and their implications can be difficult to understand. In fact, highly publicized privacy breaches (e.g., [2], [3], [4]) have resulted from fundamental misunderstandings about what can be guaranteed by a particular class of sanitizing algorithms.

Thus it is important to be able to systematically analyze privacy definitions. However, developing a framework for such an analysis is an open problem. Current analyses of privacy definitions are based on a hit-or-miss methodology: they evaluate the ability of a specific attack strategy to uncover specific types of sensitive information [5], [6], [7], [8].

We do note that tremendous insight can be obtained from such a methodology. For example, Dinur and Nissim [6] considered privacy definitions that allow algorithms to provide approximate answers (i.e. with at most $o(\sqrt{n})$ perturbation[1]) to polynomially many linear queries. They showed that linear programming can be used to approximately reconstruct the original data with $o(n)$ error in terms of Hamming distance. Their subsequent analyses were influential in the development of a state-of-the-art privacy definition called *differential privacy* [9].

However, the hit-or-miss methodology also has serious drawbacks. If a specific attack fails, little information is gained (i.e. it does not preclude the possibility that a modified attack would succeed). This methodology would also fail to identify non-sensitive pieces of information that are unnecessarily protected by a privacy definition; removing such protections would increase the utility of sanitized data.

In this paper, we introduce a new methodology that can serve as a foundation for a systematic analysis of privacy definitions. In order to explain this methodology, we present a concrete instantiation called the *row cone analysis*; the row cone analysis transforms the study of privacy definitions into the study of linear inequalities. We apply the row cone analysis to *randomized response* [10] to show that its privacy properties result from strong protections of the parity of a dataset.

A. Outline

After introducing notation in Section II, we present the row-cone analysis in Section III. The row cone analysis uses two mathematical structures – the consistent closure (Section III-A) and row cone (Section III-C) of a privacy definition – that can be generalized to create other types of analyses (these generalizations are discussed in Sections III-B and III-D, respectively). We use the row cone analysis to study randomized response in Section IV. Finally, we discuss other uses of the methodology, such as principled methods for relaxing privacy definitions, in Section V.

II. Notation

Let $\mathbb{I} = \{D_1, D_2, \ldots, D_N\}$ be a the input domain: a finite collection possible datasets. A *sanitizing algorithm* $\mathfrak{M}$ is a deterministic or randomized algorithm whose domain contains $\mathbb{I}$. We view sanitizing algorithms as conditional probability

[1] n is the number of records in the data.

distributions: $P_{\mathfrak{M}}(\omega \mid D) \stackrel{\text{def}}{=} P(\mathfrak{M}(D) = \omega)$. For convenience, we also represent a sanitizing algorithm $\mathfrak{M}$ as a matrix where the columns are indexed by $\mathbb{I}$, rows are indexed by the countable set $\text{range}(\mathfrak{M})$, and whose entries are $P(\mathfrak{M}(D) = \omega)$, as shown below.

$$\begin{array}{c} \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & & D_2 & \dots \end{array} \\ \begin{pmatrix} P(\mathfrak{M}(D_1) = \omega_1) & P(\mathfrak{M}(D_2) = \omega_1) & \dots \\ P(\mathfrak{M}(D_1) = \omega_2) & P(\mathfrak{M}(D_2) = \omega_2) & \dots \\ P(\mathfrak{M}(D_1) = \omega_3) & P(\mathfrak{M}(D_2) = \omega_3) & \dots \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

We use the notation $P(\mathfrak{M}(\cdot) = \omega)$ to refer to the vector $\langle P(\mathfrak{M}(D_1) = \omega),\ P(\mathfrak{M}(D_2) = \omega), \dots \rangle$, which is the row of the matrix representation of $\mathfrak{M}$ that is indexed by ω.

A privacy definition $\mathfrak{Priv}$ is just a *set* of sanitizing algorithms with the same input domain $\mathbb{I}$. For example, ϵ-differential privacy [9] is the set of algorithms that satisfy certain constraints on their probabilistic behavior, while randomized response [10] is just a set containing one algorithm.

III. The Row Cone Analysis

In this section we present the row cone analysis while discussing how it can be generalized to other types of systematic analyses of privacy definitions. To analyze a privacy definition $\mathfrak{Priv}$, the main idea is to identify implicit privacy assumptions (Section III-A), specify the type of information about $\mathfrak{M} \in \mathfrak{Priv}$ that should be used for inference, identify how this information is constrained by $\mathfrak{Priv}$, and interpret those constraints in terms of statistical inference (Section III-C). We explain in more detail below.

A. The Consistent Closure

We would like to think of a privacy definition $\mathfrak{Priv}$ as a complete specification of algorithms we should trust to produce sanitized data from sensitive datasets. However, in many cases, these specifications are incomplete.

For example, the principle of k-anonymity [2] defines a "k-anonymous table format" and states that we should trust algorithms that produce output tables in this format. Let $\mathfrak{M}_*$ be one such algorithm, let $\mathcal{A}$ be an algorithm that builds a statistical model from k-anonymous tables, and let $\mathcal{A} \circ \mathfrak{M}_*$ be the composite algorithm that first runs $\mathfrak{M}_*$ on the sensitive data and then runs $\mathcal{A}$ on the output of $\mathfrak{M}_*$. If we accept the principle of k-anonymity, then we should trust $\mathfrak{M}_*$. Should we also trust $\mathcal{A} \circ \mathfrak{M}_*$? Intuitively yes, because if an output table ω produced by $\mathfrak{M}_*$ is safe to release, then building a model using only the table ω should be safe too. However, k-anonymity *does not* tell us that we should trust $\mathcal{A} \circ \mathfrak{M}_*$.

The logical conclusion is that we should start with a (possibly incomplete) privacy definition $\mathfrak{Priv}$ and use rules of the form "if you trust algorithm M_a then you should also trust algorithm M_b" to expand $\mathfrak{Priv}$ until it encompasses all algorithms we should trust.

We can interpret two privacy axioms from [11] as "if-then" rules to suit our purposes:

Axiom 3.1 (Post-processing [11]): Let $\mathfrak{Priv}$ be a privacy definition. Let $\mathfrak{M} \in \mathfrak{Priv}$ and let $\mathcal{A}$ be any algorithm whose domain contains the range of $\mathfrak{M}$ (and whose randomness is independent of the randomness in $\mathfrak{M}$). Then $\mathcal{A} \circ \mathfrak{M}$ should belong to $\mathfrak{Priv}$.

Axiom 3.2 (Convexity [11]): Let $\mathfrak{Priv}$ be a privacy definition. Let $\mathfrak{M}_1 \in \mathfrak{Priv}$ and $\mathfrak{M}_2 \in \mathfrak{Priv}$. Define $p\,\mathfrak{M}_1 + (1-p)\,\mathfrak{M}_2$ to be the algorithm that runs $\mathfrak{M}_1$ with probability p and $\mathfrak{M}_2$ with probability $1-p$. Then $p\,\mathfrak{M}_1 + (1-p)\,\mathfrak{M}_2$ should belong to $\mathfrak{Priv}$.

We can then define the consistent closure of $\mathfrak{Priv}$ as:

Definition 3.3 (Consistent Closure): The consistent closure of $\mathfrak{Priv}$, denoted by $\text{closure}(\mathfrak{Priv})$, is the smallest set of algorithms that contains $\mathfrak{Priv}$ and is consistent with Axioms 3.1 and 3.2.

The following theorem shows that the consistent closure is indeed obtained by adding algorithms using the if-then rules from the axioms.

Theorem 3.4: Given a privacy definition $\mathfrak{Priv}$, its consistent closure $\text{closure}(\mathfrak{Priv})$ can be obtained from the following process:

1) Define $\mathfrak{Priv}^{(1)}$ to be the set of all (deterministic and randomized algorithms) of the form $\mathcal{A} \circ \mathfrak{M}$, where $\mathfrak{M} \in \mathfrak{Priv}$, $\text{range}(\mathfrak{M}) \subseteq \text{domain}(\mathcal{A})$, and the random bits of $\mathcal{A}$ and $\mathfrak{M}$ are independent of each other.
2) Define $\mathfrak{Priv}^{(2)}$ to be the set of all algorithms of the form $p_1\,\mathfrak{M}_1 + p_2\,\mathfrak{M}_2 + \dots + p_n\,\mathfrak{M}_n$ for all positive integers n, finite sequences $\mathfrak{M}_1, \dots, \mathfrak{M}_n \in \mathfrak{Priv}^{(1)}$, and probability vectors $\vec{p} = \langle p_1, \dots, p_n \rangle$.
3) Set $\text{closure}(\mathfrak{Priv}) = \mathfrak{Priv}^{(2)}$.

The proof can be found in [12]. We derive the consistent closure for randomized response in Section IV

B. Implications of $\text{closure}(\mathfrak{Priv})$

The significance of $\text{closure}(\mathfrak{Priv})$ is that this is the complete set of algorithms we should trust if we decide to trust $\mathfrak{Priv}$ (and accept Axioms 3.1 and 3.2). In this section we briefly discuss generalizations and computational issues.

1) Computation and design guidelines: for a given privacy definition $\mathfrak{Priv}$, it may not always be possible to derive $\text{closure}(\mathfrak{Priv})$. This means it is difficult to describe the set of algorithms we should trust. Such complexity is to be expected when working with sets of algorithms, but it does introduce a design guideline for privacy definitions: they should be defined so that $\text{closure}(\mathfrak{Priv})$ can be determined. For example, differential privacy [9] and Pufferfish [13] are two privacy definitions for which $\mathfrak{Priv} = \text{closure}(\mathfrak{Priv})$.

2) Other generalizations: In some applications, the data curator may trust additional sanitizing algorithms as well (for example, an algorithm whose probabilistic behavior is close to some $\mathfrak{M} \in \text{closure}(\mathfrak{Priv})$ based on some distance measure). In those cases it is up to the data curator to propose additional privacy axioms to formalize extra assumptions about which algorithms can be trusted. The definition of consistent closure can then be extended in the obvious way to account for those axioms.

C. The Row Cone

The consistent closure of a privacy definition is the complete set of algorithms we should trust. To identify what a privacy definition protects, we need to extract semantic guarantees from its consistent closure. Recent results [13] suggest that semantic guarantees are more meaningful (and less confusing) when they are expressed in terms of inferences an attacker can make.

To identify restrictions on inference, the next step is to derive the *row cone*, a mathematical object which turns questions about privacy definitions into questions about geometry and linear inequalities.

The row cone is based on the likelihood principle in statistics. That is, if the data sanitizer $\mathfrak{M}$ processes sensitive data D and outputs ω, the attacker's inference should be based on $P(\mathfrak{M}(\cdot) = \omega)$ (i.e. the vector of probabilities $\langle P(\mathfrak{M}(D_1) = \omega),\ P(\mathfrak{M}(D_2) = \omega), \dots \rangle$ of generating ω) rather than some other property of $\mathfrak{M}$ and ω. The set of all such vectors, intuitively, is the set of possible pieces of information that a privacy definition could reveal about the true dataset. Noting that rescaling such vectors by a positive constant does not affect inferences made with maximum likelihood or Bayesian methods, we define $\text{rowcone}(\mathfrak{Priv})$ as follows:

Definition 3.5 (Row Cone): Let $\mathbb{I} = \{D_1, D_2, \dots\}$ be the set of possible input datasets and let $\mathfrak{Priv}$ be a privacy definition. The *row cone* of $\mathfrak{Priv}$, denoted by $\text{rowcone}(\mathfrak{Priv})$, is defined as the set of vectors:

$$\{cP(\mathfrak{M}(\cdot) = \omega) : \mathfrak{M} \in \text{closure}(\mathfrak{Priv}), \omega \in \text{range}(\mathfrak{M}), c \geq 0\}$$

Note that $\text{rowcone}(\mathfrak{Priv})$ is constructed from $\text{closure}(\mathfrak{Priv})$ and that $\text{closure}(\mathfrak{Priv})$ is a convex set (due to Axiom 3.2). Therefore $\text{rowcone}(\mathfrak{Priv})$ is a convex set too; it can be visualized as in Figure 1.

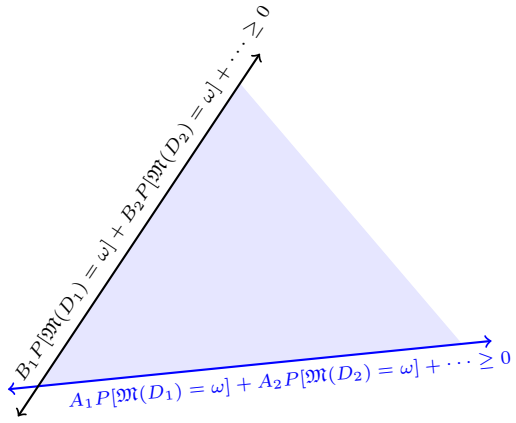

Fig. 1. An example of a row cone (shaded) and its defining linear inequalities.

In order to extract semantic guarantees, we need to find out what is common to all of the vectors in $\text{rowcone}(\mathfrak{Priv})$. This is where the geometry of $\text{rowcone}(\mathfrak{Priv})$ is particularly useful.

The closure of a convex set is equivalent to the intersection of halfplanes containing the set. Since halfplanes are defined by linear inequalities, and since $\text{rowcone}(\mathfrak{Priv})$ is actually a convex cone, it can be described using linear (in)equalities of the form:[2]

$$\begin{aligned} A_1P(\mathfrak{M}(D_1) = \omega) + A_2P(\mathfrak{M}(D_2) = \omega) + \dots &\geq 0 \text{ or} \\ A_1P(\mathfrak{M}(D_1) = \omega) + A_2P(\mathfrak{M}(D_2) = \omega) + \dots &= 0 \text{ or} \\ A_1P(\mathfrak{M}(D_1) = \omega) + A_2P(\mathfrak{M}(D_2) = \omega) + \dots &> 0 \end{aligned}$$

that must hold for each $\mathfrak{M} \in \text{closure}(\mathfrak{Priv})$ and each $\omega \in \text{range}(\mathfrak{M})$.

To extract semantic guarantees, the key insight is that each of these (in)equalities can be interpreted as statements about prior and posterior distributions. We can view each coefficient A_{i_j} as a possible value for a prior probability $P(\text{data} = D_{i_j})$ or a value proportional to a prior probability. We can define S_+ to be the collection of all databases D_i for which A_i (i.e. the coefficient of $P(\mathfrak{M}(D_i) = \omega)$) is positive. Similarly, S_- can be defined as the collection of datasets associated with negative coefficients. With these interpretations (and some simple algebraic manipulations), an inequality of the form $A_1P(\mathfrak{M}(D_1) = \omega) + A_2P(\mathfrak{M}(D_2) = \omega) + \dots \geq 0$ can be turned into statements such as:

$$\alpha \geq \frac{P(\text{data} \in S_+ \mid \mathfrak{M}(\text{data}) = \omega)}{P(\text{data} \in S_- \mid \mathfrak{M}(\text{data}) = \omega)} \quad (1)$$

$$\alpha' \geq \frac{P(\text{data} \in S_+ \mid \mathfrak{M}(\text{data}) = \omega)}{P(\text{data} \in S_- \mid \mathfrak{M}(\text{data}) = \omega)} \Big/ \frac{P(\text{data} \in S_+)}{P(\text{data} \in S_-)} \quad (2)$$

Equation 1 means that if an attacker uses a certain class of prior distributions then after seeing the sanitized data, the probability of some set S_+ is no more than α times the probability of some set S_-. Equation 2 means that if an attacker uses a certain class of priors, then the relative odds of S_+ vs. S_- can increase by at most α' after seeing the sanitized data. Of particular importance are the sets S_+ and S_- of possible input datasets, whose relative probabilities are constrained by the privacy definition. In an ideal world they would correspond to something we are trying to protect; for example, S_+ could be the set of potential databases in which Bob has cancer and S_- could be the set of potential databases in which Bob is healthy. If a privacy definition is not properly designed, S_+ and S_- could correspond to concepts that may not need protection for certain applications (for example, S_+ could be the set of databases with even parity and S_- could be the set of databases with odd parity).

D. Implications of $\text{rowcone}(\mathfrak{Priv})$

The row cone turns the study of privacy definitions into the study of convex geometry and linear inequalities. Thus many existing mathematical tools can be used to study privacy definitions. In this section we again discuss computational issues and further generalizations.

[2]This is true for finite-dimensional vector spaces. In infinite dimensional vector spaces, finitely additive measures may be needed. For this reason, we require that $\mathbb{I}$, the set of possible datasets, be finite since $|\mathbb{I}|$ is the dimensionality of each vector in $\text{rowcone}(\mathfrak{Priv})$.

1) Computation and design guidelines: for some privacy definitions $\mathfrak{Priv}$, it may not be possible to determine $\text{rowcone}(\mathfrak{Priv})$. Note that $\text{rowcone}(\mathfrak{Priv})$ captures the notion of Bayesian guarantees (i.e. the restrictions on the relationships between prior and posterior distributions that must hold when using an algorithm that satisfies the privacy definition). For this reason, we believe that intractability of the row cone of a privacy definition $\mathfrak{Priv}$ points to possible design flaws in $\mathfrak{Priv}$ – how can one justify a privacy definition whose semantics are obscure? Thus another design principles for privacy definitions is to ensure that $\text{rowcone}(\mathfrak{Priv})$ is easily computable. For example, both differential privacy [9] and Pufferfish [13] have the property that $\mathfrak{M} \in \mathfrak{Priv}$ if and only if every row of the matrix representation of $\mathfrak{M}$ (as defined in Section II) belongs to $\text{rowcone}(\mathfrak{Priv})$. Pufferfish [13], in particular, was designed with these kinds of Bayesian semantics in mind.

2) Other generalizations: the row cone is intended to provide privacy semantics for attackers that only use the likelihood vector $P(\mathfrak{M}(\cdot) = \omega)$ for inference when they see a sanitized output ω. Other types of semantics are also possible. For example, one could consider randomized semantics such as "with probability p_1, the attacker's computed odds of Bob having cancer vs. Bob being healthy increase by at most α_1, with probability p_2, the odds increase by at most α_2, etc." In those cases, instead of the row cone, one could extract all probability vectors $P(\mathfrak{M}(\cdot) = \omega)$ (but not rescale them as in Definition 3.5) or one could extract and interpret some other mathematical structure from $\text{closure}(\mathfrak{Priv})$.

IV. Application to Randomized Response

In this section, we derive the consistent closure and row cone of randomized response. We then extract previously unknown semantic guarantees for this privacy definition. For randomized response, the dataset can be thought of as a bit string of length k; where each bit j corresponds to the value of a yes/no question posed to individual j.

Definition 4.1 (Domain of randomized reponse): Let the input domain $\mathbb{I} = \{D_1, \ldots, D_{2^k}\}$ be the set of all bit strings of length k. The bit strings are ordered in reverse lexicographic order. Thus D_1 is the string whose bits are all 1 and D_{2^k} is the string whose bits are all 0.

Definition 4.2 (Randomized response algorithm): Given a privacy parameter $p > 1/2$, let $\mathfrak{M}_{rr(p)}$ be the algorithm that, on input $D \in \mathbb{I}$, independently flips each bit of D with probability $1 - p$.

Note that randomized response, as a privacy definition, is the set $\{\mathfrak{M}_{rr(p)}\}$. To extract semantic guarantees from $\{\mathfrak{M}_{rr(p)}\}$, we first derive the consistent closure and row cone (Theorem 4.3), and then reinterpret constraints on the row cone as statements about prior and posterior beliefs (Theorem 4.5).

Theorem 4.3: Given input space $\mathbb{I} = \{D_1, \ldots, D_{2^k}\}$ of bit strings of length k and a privacy parameter $p > 1/2$,

- A vector $\vec{x} = (x_1, \ldots, x_{2^k}) \in \text{rowcone}(\{\mathfrak{M}_{rr(p)}\})$ if and only if for every bit string s of length k,
$$\sum_{i=1}^{2^k} p^{\text{ham}(s,D_i)}(p-1)^{k-\text{ham}(s,D_i)} x_i \geq 0$$
where $\text{ham}(s, D_i)$ is the Hamming distance between s and D_i.
- An algorithm $\mathfrak{M}$ with matrix representation M (see Section II) belongs to $\text{closure}(\{\mathfrak{M}_{rr(p)}\})$ if and only if every row of M belongs to $\text{rowcone}(\{\mathfrak{M}_{rr(p)}\})$.

The proof can be found in [12].

We illustrate this theorem with an example of tables with $k = 2$ tuples.

Example 4.4: With 2 tuples and one binary attribute, the domain $\mathbb{I} = \{11, 10, 01, 00\}$. By Theorem 4.3, an algorithm $\mathfrak{M}$ with matrix representation M belongs to the closure of randomized response (with privacy parameter p) if for every vector $\vec{x} = (x_{11}, x_{10}, x_{01}, x_{00})$ that is a row of M, the following four constraints hold:

$$p^2 x_{00} + (1-p)^2 x_{11} \geq p(1-p)x_{01} + p(1-p)x_{10} \quad (3)$$
$$(1-p)^2 x_{00} + p^2 x_{11} \geq p(1-p)x_{01} + p(1-p)x_{10} \quad (4)$$
$$p^2 x_{01} + (1-p)^2 x_{10} \geq p(1-p)x_{00} + p(1-p)x_{11} \quad (5)$$
$$(1-p)^2 x_{01} + p^2 x_{10} \geq p(1-p)x_{00} + p(1-p)x_{11} \quad (6)$$

We use Example 4.4 to explain the intuition behind the process of extracting Bayesian semantic guarantees from the row cone of randomized response, as given by the constraints in Equations 3, 4, 5, and 6. Let us consider the following two attackers.

Attacker 1. This attacker has the prior belief that $P(\text{data} = 11) = p^2$, $P(\text{data} = 00) = (1-p)^2$ and $P(\text{data} = 01) = P(\text{data} = 10) = p(1-p)$, so that each bit is independent and equals 1 with probability p (this p is the same as the privacy parameter p in randomized response). Let us consider the effect of the constraint in Equation 3 on the attacker's inference. This constraint says that for all trusted algorithms $\mathfrak{M}$ (i.e. $\mathfrak{M} \in \text{closure}(\{\mathfrak{M}_{rr(p)}\})$) and for all $\omega \in \text{range}(\mathfrak{M})$,

$$\begin{aligned} &p^2 P[\mathfrak{M}(11) = \omega] + (1-p)^2 P[\mathfrak{M}(00) = \omega] \\ &\quad \geq \quad p(1-p)P[\mathfrak{M}(01) = \omega] + p(1-p)P[\mathfrak{M}(10) = \omega] \end{aligned} \quad (7)$$

Note that the coefficients in the linear constraints have the same values as the prior probabilities of the possible input datasets. Substituting those prior beliefs into Equation 7, we get the constraint that for all $\omega \in \text{range}(\mathfrak{M})$:

$$\begin{aligned} &P(\text{data} = 11)P[\mathfrak{M}(11) = \omega] + P(\text{data} = 00)P[\mathfrak{M}(00) = \omega] \\ &\geq P(\text{data} = 01)P[\mathfrak{M}(01) = \omega] + P(\text{data} = 10)P[\mathfrak{M}(10) = \omega] \end{aligned}$$

This, in turn, is equal to the following constraint on the attacker's belief about the joint distribution of the input and output of $\mathfrak{M}$:

$$\begin{aligned} &P[\text{parity}(\text{data}) = 0 \wedge \mathfrak{M}(\text{data}) = \omega] \\ &\quad \geq \quad P[\text{parity}(\text{data}) = 1 \wedge \mathfrak{M}(\text{data}) = \omega] \end{aligned}$$

Dividing both sides by $P(\mathfrak{M}(\text{data}) = \omega)$, where "data" is a random variable, we see that Equation 3 eventually leads to the following constraint on the attacker's posterior distribution:

$$\begin{aligned} &P[\text{parity}(\text{data}) = 0 \mid \mathfrak{M}(\text{data}) = \omega] \\ &\quad \geq \quad P[\text{parity}(\text{data}) = 1 \mid \mathfrak{M}(\text{data}) = \omega] \end{aligned}$$

Thus if an attacker believes that bits in the database are generated independently with probability p, then after seeing the sanitized output, the attacker will believe that the true input is more likely to have even parity. Looking at the attacker's *prior* beliefs, we see that the prior probability of even parity, $p^2 + (1-p)^2$, is greater than the prior probability of odd parity, $2p(1-p)$. Thus the belief about which parity is most likely remains unchanged. Equations 4, 5, and 6 lead to similar guarantees, which are summarized in Theorem 4.5.

Attacker 2. This attacker believes that the first bit is 1 with probability $1/2$ and believes the second bit is 1 with probability p (the bits are independent of each other). In this case, the attacker's prior beliefs are that odd parity and even parity are *equally likely*. It is easy to see that now the output of $\mathfrak{M} \in \text{closure}\{\mathfrak{M}_{rr(p)}\}$ can make the attacker change his mind about which parity is more likely (for example, consider what happens when $\mathfrak{M}_{rr(p)}$ outputs 01 or 00). This is true because the attacker was so unsure about parity that even the slightest amount of evidence can change his beliefs about which parity is (slightly) more likely. However, the attacker will not change his mind about the parity of the second bit, for which he has greater confidence. This result is a consequence of Theorem 4.5 below, which formally presents the semantic guarantees of randomized response.

The following theorem generalizes these observations to sets of bits for which the attacker's prior is bounded away from $1/2$ (i.e. each bit has its own prior probability that is $\leq 1-p$ or $\geq p$). For those sets, the parity that is *a priori* more likely is also *a posteriori* more likely. It also shows that only algorithms $\mathfrak{M} \in \text{closure}(\{\mathfrak{M}_{rr(p)}\})$ have this property (as a consequence of the connection between the consistent closure and row cone of randomized response that was identified in Theorem 4.3). Thus these semantics completely characterize the protections offered by randomized response.

Theorem 4.5: Let p be a privacy parameter and let $\mathbb{I} = \{D_1, \ldots, D_{2^k}\}$. Let $\mathfrak{M}$ be an algorithm that has a matrix representation M such that every row of M belongs to the row cone of randomized response. If the attacker believes that the bits in the data are independent and bit i is equal to 1 with probability q_i, then $\mathfrak{M}$ protects the parity of any subset of bits that have prior probability $\geq p$ or $\leq 1-p$. That is, for any subset $J \equiv \{\ell_1, \ldots, \ell_m\}$ of bits of the input data such that $q_{\ell_j} \geq p \vee q_{\ell_j} \leq 1-p$ for $j = 1, \ldots, m$, the following holds:

- If $P(\text{parity}(J) = 0) \geq P(\text{parity}(J) = 1)$ then
  $P(\text{parity}(J) = 0 \mid \mathfrak{M}(\text{data})) \geq P(\text{parity}(J) = 1 \mid \mathfrak{M}(\text{data}))$
- If $P(\text{parity}(J) = 1) \geq P(\text{parity}(J) = 0)$ then
  $P(\text{parity}(J) = 1 \mid \mathfrak{M}(\text{data})) \geq P(\text{parity}(J) = 0 \mid \mathfrak{M}(\text{data}))$

Furthermore, an algorithm $\mathfrak{M}$ can only provide these guarantees if every row of its matrix representation belongs to $\text{rowcone}(\{\mathfrak{M}_{rr(p)}\})$.
The proof can be found in [12].

V. Relaxing Privacy Definitions

Theorems 4.3 and 4.5 described properties of the set of algorithms we should trust if we are prepared to trust randomized response – this set of algorithms is completely characterized by protections of the parity of any subset of the input dataset. Protecting the parity of individual bits is important, since each bit corresponds to an individual. However, protecting the parity of larger subsets of bits is often irrelevant in statistical privacy.

If one is interested in weakening randomized response to get rid of these unnecessary protections, then the row cone provides a useful starting point. Geometrically, the row cone is a set of vectors that is described by linear inequalities. One approach to weakening a privacy definition is to enlarge the row cone. Such an enlarged row cone R can become the basis of a new privacy definition $\mathfrak{Priv}$ that can be defined as follows: $\mathfrak{M} \in \mathfrak{Priv}$ if and only if $P(\mathfrak{M}(\cdot) = \omega) \in R$ for all $\omega \in \text{range}(\mathfrak{M})$.

One way of enlarging the row cone is to use Fourier-Motzkin elimination to relax the original linear inequalities. Applying this technique to randomized response results in a privacy definition that requires that $P(\mathfrak{M}(D) = \omega) \leq \frac{p}{1-p} P(\mathfrak{M}(D') = \omega)$ whenever the Hamming distance between D and D' is equal to 1. This is precisely the definition of $\log\left(\frac{p}{1-p}\right)$-differential privacy [9].

These observations lead us to a moment of wild speculation: if this framework existed almost half a century ago – around the time Warner was developing randomized response [10] – would it have accelerated the development of differential privacy and related technologies? While this question cannot be definitively answered, we believe the framework presented here could aid other discoveries in privacy technology.

VI. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1054389.

References

[1] Kaggle, http://www.kaggle.com.
[2] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
[3] A. Narayanan and V. Shmatikov, "How to break anonymity of the netflix prize dataset," 2006. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0610105
[4] M. Barbaro and T. Zeller, "A face is exposed for AOL searcher no. 4417749," *New York Times*, August 9 2006.
[5] D. Kifer, "Attacks on privacy and de finetti's theorem," in *SIGMOD*, 2009.
[6] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *PODS*, 2003.
[7] R. Wong, A. Fu, K. Wang, and J. Pei, "Minimality attack in privacy preserving data publishing," in *VLDB*, 2007.
[8] J. Reiter, "Estimating risks of identification disclosure for microdata," *Journal of the American Statistical Association*, vol. 100, pp. 1103 – 1113, 2005.
[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis." in *TCC*, 2006.
[10] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, 1965.
[11] D. Kifer and B.-R. Lin, "An axiomatic view of statistical privacy and utility," To appear in Journal of Privacy and Confidentiality.
[12] B.-R. Lin and D. Kifer, "A framework for extracting semantic guarantees from privacy definitions," http://arxiv.org/abs/1208.5443.
[13] D. Kifer and A. Machanavajjhala, "A rigorous and customizable framework for privacy," in *PODS*, 2012.