

# The Privacy Funnel from the Viewpoint of Local Differential Privacy

Milan Lopuhaä-Zwakenberg  
 Department of Mathematics and Computer Science  
 Eindhoven University of Technology  
 Eindhoven, the Netherlands  
 email: m.a.lopuhaa@tue.nl

**Abstract**—In the Open Data approach, governments want to share their datasets with the public, for accountability and to support participation. Data must be opened in such a way that individual privacy is safeguarded. The Privacy Funnel is a mathematical approach that produces a sanitised database that does not leak private data beyond a chosen threshold. The downsides to this approach are that it does not give worst-case privacy guarantees, and that finding optimal sanitisation protocols can be computationally prohibitive. We tackle these problems by using differential privacy metrics, and by considering local protocols which operate on one entry at a time. We show that under both the Local Differential Privacy and Local Information Privacy leakage metrics, one can efficiently obtain optimal protocols; however, Local Information Privacy is both more closely aligned to the privacy requirements of the Privacy Funnel scenario, and more efficiently computable. We also consider the scenario where each user has multiple attributes, for which we define *Side-channel Resistant Local Information Privacy*, and we give efficient methods to find protocols satisfying this criterion while still offering good utility. Exploratory experiments confirm the validity of these methods.

**Keywords**—Privacy funnel; local differential privacy; information privacy; database sanitisation; complexity.

## I. INTRODUCTION

Under the Open Data paradigm, governments and other public organisations want to share their collected data with the general public. This increases a governments transparency, and it also gives citizens and businesses the means to participate in decision-making, as well as using the data for their own purposes. However, while the released data should be as faithful to the raw data as possible, individual citizen’s private data should not be compromised by such data publication.

To state this problem mathematically, let  $\mathcal{X}$  be a finite set. Consider a database  $\vec{X} = (X_1, \dots, X_n) \in \mathcal{X}^n$  owned by a data aggregator, containing a data item  $X_i \in \mathcal{X}$  for each user  $i$  (For typical database settings, each user’s data is a vector of attributes  $X_i = (X_i^1, \dots, X_i^m)$ ; we will consider this in more detail in Section V). This data may not be considered sensitive by itself, however, it might be correlated to a secret  $S_i$ . The aggregator wants to release the database to the general public while preventing adversaries from retrieving the secret values  $S_i$ . For instance,  $X_i$  might contain the age, sex, weight, skin colour, and average blood pressure of person  $i$ , while  $S_i$  is the presence of some medical condition. To publicise the data without leaking the  $S_i$ , the aggregator releases a privatised database  $\vec{Y} = (Y_1, \dots, Y_n)$ , obtained from applying a sanitisation mechanism  $\mathcal{R}$  to  $\vec{X}$ . One way to formulate this is by considering the *Privacy Funnel*:

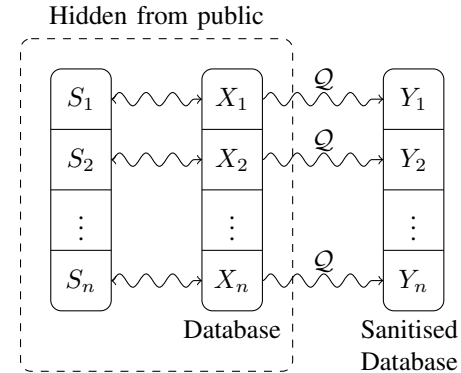


Figure 1. Model of the Privacy Funnel with local protocols.

**Problem 1.** (Privacy Funnel, [4]) Suppose the joint probability distribution of  $\vec{S}$  and  $\vec{X}$  is known to the aggregator, and let  $M \in \mathbb{R}_{\geq 0}$ . Then, find the privatization mechanism  $\mathcal{R}$  such that  $I(\vec{X}; \vec{Y})$  is maximised while  $I(\vec{S}; \vec{Y}) \leq M$ .

There are two difficulties with this approach:

- 1) Finding and implementing good privatization mechanisms that operate on all of  $\vec{X}$  can be computationally prohibitive for large  $n$ , as the complexity is exponential in  $n$  [6] [14].
- 2) Taking mutual information as a leakage measure has as a disadvantage that it gives guarantees about the leakage in the average case. If  $n$  is large, this still leaves room for the sanitisation protocol to leak undesirably much information about a few unlucky users.

To deal with these two difficulties, we make two changes to the general approach. First, we look at *local* data sanitisation, i.e., we consider optimization protocols  $\mathcal{Q}: \mathcal{X} \rightarrow \mathcal{Y}$ , for some finite set  $\mathcal{Y}$ , and we apply  $\mathcal{Q}$  to each  $X_i$  individually; this situation is depicted in Figure 1. These can be efficiently implemented. Second, to ensure strong privacy guarantees even in worst-case scenarios, we take stricter notions of privacy, based on Local Differential Privacy (LDP) [11].

The structure of this paper is as follows. In Section II, we define the mathematical setting of our problem. We discuss two privacy notions, LDP and Local Information Privacy (LIP), and discuss their relation to the Privacy Funnel. In Sections III and IV, we show that for a given level of LDP or LIP, respectively, one can efficiently find the optimal sanitisation protocol. In Section V, we consider the setting where every  $X_i$  is a vector of attributes, and we show how

to make protocols that protect against side-channel attacks. In Section VI, we numerically assess the methods presented in this paper.

## II. MATHEMATICAL SETTING

The database  $\vec{X} = (X_1, \dots, X_n)$  consists out of a data item  $X_i$  for each user  $i$ , each an element of a given finite set  $\mathcal{X}$ . Furthermore, each user has sensitive data  $S_i \in \mathcal{S}$ , which is correlated with  $X_i$ ; again we assume  $\mathcal{S}$  to be finite (see Figure 1). We assume each  $(S_i, X_i)$  is drawn independently from the same distribution  $p_{S,X}$  on  $\mathcal{S} \times \mathcal{X}$  which is known to the aggregator through observing  $(\vec{S}, \vec{X})$  (if one allows for non-independent  $X_i$ , then differential privacy is no longer an adequate privacy metric [5] [16]). The aggregator, who has access to  $\vec{X}$ , sanitises the database by applying a sanitisation protocol (i.e., a random function)  $\mathcal{Q}: \mathcal{X} \rightarrow \mathcal{Y}$  to each  $X_i$ , outputting  $\vec{Y} = (Y_1, \dots, Y_n) = (\mathcal{Q}(X_1), \dots, \mathcal{Q}(X_n))$ . The aggregator's goal is to find a  $\mathcal{Q}$  that maximises the information about  $X_i$  preserved in  $Y_i$  (measured as  $I(X_i; Y_i)$ ) while leaking only minimal information about  $S_i$ .

Without loss of generality we write  $\mathcal{X} = \{1, \dots, a\}$  and  $\mathcal{Y} = \{1, \dots, b\}$  for integers  $a, b$ . We omit the subscript  $i$  from  $X_i, Y_i, S_i$  as no probabilities depend on it, and we write such probabilities as  $p_x, p_s, p_{x|s}$ , etc., which form vectors  $p_X, p_{S|X}$ , etc., and matrices  $p_{X|S}$ , etc.

As noted before, instead of looking at the mutual information  $I(S; Y)$ , we consider two different, related measures of sensitive information leakage known from the literature. The first one is an adaptation of LDP, the *de facto* standard in information privacy [11]:

**Definition 1.** ( $\epsilon$ -LDP) Let  $\epsilon \in \mathbb{R}_{\geq 0}$ . We say that  $\mathcal{Q}$  satisfies  $\epsilon$ -LDP w.r.t.  $S$  if for all  $y \in \mathcal{Y}$  and all  $s, s' \in \mathcal{S}$  one has

$$\frac{\mathbb{P}(Y = y | S = s)}{\mathbb{P}(Y = y | S = s')} \leq e^\epsilon. \quad (1)$$

This is less strict than the 'standard' notion of  $\epsilon$ -LDP, which measures the information about  $X$  leaked in  $Y$ . This reflects the fact that we are only interested in hiding sensitive data, rather than all data; it is a specific case of what has been named 'pufferfish privacy' [12]. The advantage of LDP compared to mutual information is that it gives privacy guarantees for the worst case, not just the average case. This is desirable in the database setting, as a worst-case metric guarantees the security of the private data of all users, while average-case metrics are only concerned with the average user. Another useful privacy metric is *Local Information Privacy* (LIP) [9] [16], also called Removal Local Differential Privacy [8]:

**Definition 2.** ( $\epsilon$ -LIP) Let  $\epsilon \in \mathbb{R}_{\geq 0}$ . We say that  $\mathcal{Q}$  satisfies  $\epsilon$ -LIP w.r.t.  $S$  if for all  $s \in \mathcal{S}$  and  $y \in \mathcal{Y}$  we have

$$e^{-\epsilon} \leq \frac{\mathbb{P}(Y = y | S = s)}{\mathbb{P}(Y = y)} \leq e^\epsilon. \quad (2)$$

Compared to LDP, the disadvantage of LIP is that it depends on the distribution of  $S$ ; this is less relevant in our scenario, as the aggregator, who chooses  $\mathcal{Q}$ , has access to the distribution

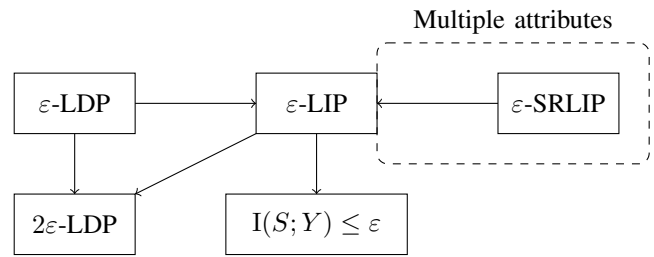


Figure 2. Relations between privacy notions. The multiple attributes setting is discussed in Section V.

of  $S$ . The advantage of LIP is that it is more closely related to an attacker's capabilities: since  $\frac{\mathbb{P}(Y=y|S=s)}{\mathbb{P}(Y=y)} = \frac{\mathbb{P}(S=s|Y=y)}{\mathbb{P}(S=s)}$ , satisfying  $\epsilon$ -LIP means that an attacker's posterior distribution of  $S$  given  $Y = y$  does not deviate from their prior distribution by more than a factor  $e^\epsilon$ . The following Lemma outlines the relations between LDP, LIP and mutual information (see Figure 2).

**Lemma 1.** (See [16]) Let  $\mathcal{Q}$  be a sanitisation protocol, and let  $\epsilon \in \mathbb{R}_{\geq 0}$ .

- 1) If  $\mathcal{Q}$  satisfies  $\epsilon$ -LDP, then it satisfies  $\epsilon$ -LIP.
- 2) If  $\mathcal{Q}$  satisfies  $\epsilon$ -LIP, then it satisfies  $2\epsilon$ -LDP, and  $I(S; Y) \leq \epsilon$ .

**Remark 1.** One can choose to employ more stringent privacy metrics for LDP and LIP by demanding that  $\mathcal{Q}$  satisfy  $\epsilon$ -LIP ( $\epsilon$ -LDP) for a set of  $p_{S,X}$ , instead of only one [12]. Letting  $p_{S,X}$  range over all possible distributions on  $\mathcal{S} \times \mathcal{X}$  yields standard LIP (LDP) (i.e., w.r.t.  $X$ ).

In this notation, instead of Problem 1 we consider the following problem:

**Problem 2.** Suppose  $p_{S,X}$  is known to the aggregator, and let  $\epsilon \in \mathbb{R}_{\geq 0}$ . Then, find the sanitisation protocol  $\mathcal{Q}$  such that  $I(X; Y)$  is maximised while  $\mathcal{Q}$  satisfies  $\epsilon$ -LDP ( $\epsilon$ -LIP, respectively) with respect to  $S$ .

Note that this problem does not depend on the number of users  $n$ , and as such this approach will find solutions that are scalable w.r.t.  $n$ .

## III. OPTIMIZING $\mathcal{Q}$ FOR $\epsilon$ -LDP

Our goal is now to find the optimal  $\mathcal{Q}$ , i.e., the protocol that maximises  $I(X; Y)$  while satisfying  $\epsilon$ -LDP, for a given  $\epsilon$ . We can represent any sanitisation protocol as a matrix  $Q \in \mathbb{R}^{b \times a}$ , where  $Q_{y|x} = \mathbb{P}(Y = y | X = x)$ . Then,  $Q$  defines a sanitisation protocol  $\mathcal{Q}$  satisfying  $\epsilon$ -LDP if and only if

$$\forall x: \sum_y Q_{y|x} = 1, \quad (3)$$

$$\forall x, y: 0 \leq Q_{y|x}, \quad (4)$$

$$\forall s, s', y: (Q p_{X|s})_y \leq e^\epsilon (Q p_{X|s'})_y. \quad (5)$$

As such, for a given  $\mathcal{Y}$ , the set of  $\epsilon$ -LDP-satisfying sanitisation protocols can be considered a closed, bounded, convex poly-

tope  $\Gamma$  in  $\mathbb{R}^{b \times a}$ . This fact allows us to efficiently find optimal protocols.

**Theorem 1.** *Let  $\varepsilon \in \mathbb{R}_{\geq 0}$ . Let  $\mathcal{Q}: \mathcal{X} \rightarrow \mathcal{Y}$  be the  $\varepsilon$ -LDP protocol that maximises  $I(X; Y)$ , i.e., the protocol that solves Problem 2 w.r.t. LDP.*

- 1) One has  $b \leq a$ .
- 2) Let  $\Gamma$  be the polytope described above. Then one can find  $\mathcal{Q}$  by maximising a convex function on  $\Gamma$ .

This result is obtained by generalising the results of [10]: there this is proven for regular  $\varepsilon$ -LDP (i.e., w.r.t.  $X$ ), but the arguments given in that proof hold just as well in our situation; the only difference is that their polytope is defined by the  $\varepsilon$ -LDP conditions w.r.t.  $X$ , but this has no impact on the proof. Together, these results reduce our problem to a finite optimisation problem: By point 1, we only need to consider  $\mathcal{Y} = \mathcal{X}$ , and, by point 2, we only need to find the set of vertices of  $\Gamma$ , a  $a(a-1)$ -dimensional convex polytope.

One might argue that, since the optimal  $\mathcal{Q}$  depends on  $p_{S,X}$ , the publication of  $\mathcal{Q}$  might provide an aggregator with information about the distribution of  $S$ . However, information on the distribution (as opposed to information of individual users' data) is not considered sensitive [13]. In fact, the reason why the aggregator sanitises the data is because an attacker is assumed to have knowledge about this correlation, and revealing too much information about  $X$  would cause the aggregator to use this information to infer information about  $S$ .

#### IV. OPTIMIZING $\mathcal{Q}$ FOR $\varepsilon$ -LIP

If one uses  $\varepsilon$ -LIP as a privacy metric, one can find the optimal sanitisation protocol in a similar fashion. To do this, we again describe  $\mathcal{Q}$  as a matrix, but this time a different one. Let  $q \in \mathbb{R}^b$  be the probability mass function of  $Y$ , and let  $R \in \mathbb{R}^{a \times b}$  be given by  $R_{x|y} = \mathbb{P}(X = x | Y = y)$ ; we denote its  $y$ -th row by  $R_{X|y} \in \mathbb{R}^a$ . Then, a pair  $(R, q)$  defines a sanitisation protocol  $\mathcal{Q}$  satisfying  $\varepsilon$ -LIP if and only if

$$\forall y: 0 \leq q_y, \quad (6)$$

$$Rq = p_X, \quad (7)$$

$$\forall y: \sum_x R_{x|y} = 1, \quad (8)$$

$$\forall x, y: 0 \leq R_{x|y}, \quad (9)$$

$$\forall y, s: e^{-\varepsilon} p_s \leq p_{s|X} R_{X|y} \leq e^{\varepsilon} p_s. \quad (10)$$

Note that (10) defines the  $\varepsilon$ -LIP condition, since for a given  $s, y$  we have  $\frac{p_{s|X} R_{X|y}}{p_s} = \frac{\mathbb{P}(S=s|Y=y)}{\mathbb{P}(S=s)} = \frac{\mathbb{P}(Y=y|S=s)}{\mathbb{P}(Y=y)}$ . (In)equalities (8–10) can be expressed as saying that for every  $y \in \mathcal{Y}$  one has that  $R_{X|y} \in \Delta$ , where  $\Delta$  is the convex closed bounded polytope in  $\mathbb{R}^{\mathcal{X}}$  given by

$$\Delta = \left\{ v \in \mathbb{R}^{\mathcal{X}} : \begin{array}{l} \sum_x v_x = 1, \\ \forall x: 0 \leq v_x, \\ \forall s: e^{-\varepsilon} p_s \leq p_{s|X} v \leq e^{\varepsilon} p_s \end{array} \right\}. \quad (11)$$

As in Theorem 1, we can use this polytope to find optimal protocols:

**Theorem 2.** *Let  $\varepsilon \in \mathbb{R}_{\geq 0}$ . Let  $\mathcal{Q}: \mathcal{X} \rightarrow \mathcal{Y}$  be the  $\varepsilon$ -LIP protocol that maximises  $I(X; Y)$ , i.e., the protocol that solves Problem 2 w.r.t. LIP.*

- 1) One has  $b \leq a$ .
- 2) Let  $\Delta$  be the polytope described above, and let  $\mathcal{V}$  be its set of vertices. Then one can find  $\mathcal{Q}$  by solving a  $\#\mathcal{V}$ -dimensional linear optimization problem.

This is proven for  $\varepsilon = 0$  (i.e., when  $S$  and  $Y$  are independent) in [15], but the proof works similarly for  $\varepsilon > 0$ ; the main difference is that the equality constraints of their (10) will be replaced by the inequality constraints of our (10), but this has no impact on the proof presented there. Since linear optimization problems can be solved fast, again the optimization problem reduces to finding the vertices of a polytope. The advantage of this approach, however, is that  $\Delta$  is a  $(a-1)$ -dimensional polytope, while  $\Gamma$  is  $a(a-1)$ -dimensional. The time complexity of vertex enumeration is linear in the number of vertices [1], while the number of vertices can grow exponentially in the dimension of the polyhedron [2]. Together, this means that the dimension plays a huge role in the time complexity, hence we expect finding the optimum under LIP to be significantly faster than under LDP.

#### V. MULTIPLE ATTRIBUTES

An often-occurring scenario is that a user's data consists out of multiple attributes, i.e.,  $X_i = (X_i^1, \dots, X_i^m) \in \mathcal{X} = \prod_{j=1}^m \mathcal{X}^j$ . This can be problematic for our approach for two reasons:

- 1) Such a large  $\mathcal{X}$  can be problematic, since the computing time for optimisation both under LDP and LIP will depend heavily on  $a$ .
- 2) In practice, an attacker might sometimes utilise side channels to access to some subsets of attributes  $X_i^j$  for some users. For these users, a sanitisation protocol can leak more information (w.r.t. to the attacker's updated prior information) than its LDP/LIP parameter would suggest.

To see how the second problem might arise in practice, suppose that  $X_i^1$  is the height of individual  $i$ ,  $X_i^2$  is their weight, and  $S_i$  is whether  $i$  is obese or not. Since height is only lightly correlated with obesity, taking  $Y_i = X_i^1$  would satisfy  $\varepsilon$ -LIP for some reasonably small  $\varepsilon$ . However, suppose that an attacker has access to  $X_i^2$  via a side channel. While knowing  $i$ 's weight gives the attacker some, but not perfect knowledge about  $i$ 's obesity, the combination of the weight from the side channel, and the height from the  $Y_i$ , allows the attacker to calculate  $i$ 's BMI, giving much more information about  $i$ 's obesity. Therefore, the given protocol gives much less privacy in the presence of this side channel.

To solve the second problem, we introduce a more stringent privacy notion called *Side-channel Resistant LIP* (SRLIP), which ensures that no matter which attributes an attacker has access to, the protocol still satisfies  $\varepsilon$ -LIP with respect to the attacker's new prior distribution. One could similarly introduce

SRLDP, and many results will still hold for this privacy measure; nevertheless, since we concluded that LIP is preferable over LDP, we focus on SRLIP. For  $J \subset \{1, \dots, m\}$ , we write  $\mathcal{X}^J = \prod_{j \in J} \mathcal{X}^j$  and its elements as  $x^J$ .

**Definition 3.** ( $\varepsilon$ -SRLIP). Let  $\varepsilon > 0$ , and let  $\mathcal{X} = \prod_{j=1}^m \mathcal{X}^j$ . We say that  $\mathcal{Q}$  satisfies  $\varepsilon$ -SRLIP if for every  $y \in \mathcal{Y}$ , for every  $s \in \mathcal{S}$ , for every  $J \subset \{1, \dots, m\}$ , and for every  $x^J \in \mathcal{X}^J$  one has

$$e^{-\varepsilon} \leq \frac{\mathbb{P}(Y = y | S = s, X^J = x^J)}{\mathbb{P}(Y = y | X^J = x^J)} \leq e^{\varepsilon}. \quad (12)$$

In terms of Remark 1,  $\mathcal{Q}$  satisfies  $\varepsilon$ -SRLIP if and only if it satisfies  $\varepsilon$ -LIP w.r.t.  $p_{S, X | x^J}$  for all  $J$  and  $x^J$ . Taking  $J = \emptyset$  gives us the regular definition of  $\varepsilon$ -LIP, proving the following Lemma:

**Lemma 2.** Let  $\varepsilon > 0$ . If  $\mathcal{Q}$  satisfies  $\varepsilon$ -SRLIP, then  $\mathcal{Q}$  satisfies  $\varepsilon$ -LIP.

While SRLIP is stricter than LIP itself, it has the advantage that even when an attacker has access to some data of a user, the sanitisation protocol still does not leak an unwanted amount of information beyond the knowledge the attacker has gained via the side channel. Another advantage is that, contrary to LIP itself, SRLIP satisfies an analogon of the concept of *privacy budget* [7]:

**Theorem 3.** Let  $\mathcal{X} = \prod_{j=1}^m \mathcal{X}^j$ , and for every  $j$ , let  $Q^j: \mathcal{X}^j \rightarrow \mathcal{Y}^j$  be a sanitisation protocol. Let  $\varepsilon^j \in \mathbb{R}_{\geq 0}$  for every  $j$ . Suppose that for every  $j \leq m$ , for every  $J \subset \{1, \dots, j-1, j+1, \dots, m\}$ , and every  $x^J \in \mathcal{X}^J$ ,  $Q^j$  satisfies  $\varepsilon^j$ -LIP w.r.t.  $p_{S, X | x^J}$ . Then  $\prod_j Q^j: \mathcal{X} \rightarrow \prod_j \mathcal{Y}^j$  satisfies  $\sum_j \varepsilon^j$ -SRLIP.

The proof is presented in Appendix A. This theorem tells us that to find a  $\varepsilon$ -SRLIP protocol for  $\mathcal{X}$ , it suffices to find a sanitisation protocol for each  $\mathcal{X}^j$  that is  $\frac{\varepsilon}{m}$ -LIP w.r.t. a number of prior distributions. Unfortunately, the method of finding an optimal  $\varepsilon$ -LIP protocol w.r.t. one prior  $p_{S, X}$  of Theorem 2 does not transfer to the multiple prior setting. This is because this method only finds one  $(R, q)$ , while by (7) we need a different  $(R, q)$  for each prior distribution. Therefore, we are forced to adopt an approach similar to the one in Theorem 1. The matrix  $Q^j$  (given by  $Q^j_{y^j | x^j} = \mathbb{P}(Q^j(x^j) = y^j)$ ) corresponding to  $Q^j: \mathcal{X}^j \rightarrow \mathcal{Y}^j$  satisfies the criteria of Theorem 3 if and only if the following criteria are satisfied:

$$\forall x^j: \sum_{y^j} Q^j_{y^j | x^j} = 1, \quad (13)$$

$$\forall x^j, y^j: 0 \leq Q^j_{y^j | x^j}, \quad (14)$$

$$\forall J, x^J, s, y^j: e^{-\varepsilon/m} (Q^j p_{X^j | x^J})_{y^j} \leq (Q^j p_{X^j | s, x^J})_{y^j}, \quad (15)$$

$$\forall J, x^J, s, y^j: (Q^j p_{X^j | s, x^J})_{y^j} \leq e^{\varepsilon/m} (Q^j p_{X^j | x^J})_{y^j}. \quad (16)$$

Similar to Theorem 1, we can find the optimal  $Q^j$  satisfying these conditions by finding the vertices of the polytope defined by these equations. In terms of time complexity, the

comparison to finding the optimal  $\varepsilon$ -LIP protocol via Theorem 2 versus finding a  $\varepsilon$ -SRLIP protocol via Theorem 3 is not straightforward. The complexity of enumerating the vertices of a polytope is  $\mathcal{O}(ndv)$ , where  $n$  is the number of inequalities,  $d$  is the dimension, and  $v$  is the number of vertices [1]. For  $\Delta$  of Theorem 2 we have  $d = a - 1$  and  $n = a + 2c$ . By contrast, for the polytope defined by (13–16) satisfies  $d = a^j(a^j - 1)$  and  $n = (a^j)^2 + 2c \prod_{j' \neq j} (a^{j'} + 1)$ . Finding  $v$  for both these polytopes is difficult, but in general  $v \leq \binom{n}{d}$ . Since this grows exponentially in  $d$ , we expect Theorem 3 to be faster when the  $a^j$  are small compared to  $a$ , i.e., when  $m$  is large. We will investigate this experimentally in the next section.

## VI. EXPERIMENTS

We test the feasibility of the different methods and privacy definitions by performing small-scale experiments on synthetic data. All experiments are implemented in Matlab and conducted on a PC with Intel Core i7-7700HQ 2.8GHz and 32GB memory. We compare the computing time for finding optimal  $\varepsilon$ -LDP and  $\varepsilon$ -LIP protocols for  $c = 2$  and  $a = 5$  for 10 random  $p_{S, X}$ , obtained by generating each  $p_{s, x}$  uniformly from  $[0, 1]$  and then rescaling. We take  $\varepsilon \in \{0.5, 1, 1.5, 2\}$ ; the results are in Figure 3. As one can see, Theorem 2 gives significantly faster results than Theorem 1; the average computing time for Theorem 1 for  $\varepsilon = 0.5$  is 133s, while for Theorem 2 this is 0.0206s. With regards to the utility  $I(X; Y)$ , since  $\varepsilon$ -LDP implies  $\varepsilon$ -LIP, the optimal  $\varepsilon$ -LIP protocol will have better utility than the optimal  $\varepsilon$ -LDP protocol. However, as can be seen from the figure, the difference in utility is relatively low.

Note that for bigger  $\varepsilon$ , both the difference in computing time and the difference in  $I(X; Y)$  between LDP and LIP become less. This is because of the probabilistic relation between  $S$  and  $X$ , for  $\varepsilon$  large enough, any sanitisation protocol satisfies  $\varepsilon$ -LIP and  $\varepsilon$ -LDP. This means that as  $\varepsilon$  grows, the resulting polytopes will have less defining inequalities, hence they will have less vertices. This results in lower computation times, which affects LDP more than LIP. At the same time, the fact that every protocol is both  $\varepsilon$ -LIP and  $\varepsilon$ -LDP will result in the same optimal utility.

In Figure 4, we compare optimal  $\frac{\varepsilon}{2}$ -LDP protocols to optimal  $\varepsilon$ -LIP protocols. Again, LIP is significantly faster than LDP. Since  $\varepsilon$ -LIP implies  $\frac{\varepsilon}{2}$ -LDP, the optimal  $\frac{\varepsilon}{2}$ -LDP has higher utility; again the difference is low.

We also perform similar comparisons for multiple attributes, for  $c = 2$ ,  $a_1 = a_2 = 3$  and  $a_3 = 4$ , comparing the methods of Theorems 2 and 3. The results are presented in Figure 5. As one can see, Theorem 3 is significantly slower, with Theorem 2 being on average 476 times as fast. There is a sizable difference in utility, caused on one hand by the fact that  $\varepsilon$ -SRLIP is a stricter privacy requirement than  $\varepsilon$ -LIP, and on the other hand by the fact that Theorem 3 does not give us the optimal  $\varepsilon$ -SRLIP protocol.

## VII. CONCLUSIONS AND FUTURE WORK

Local data sanitisation protocols have the advantage of being scalable for large numbers of users. Furthermore, the

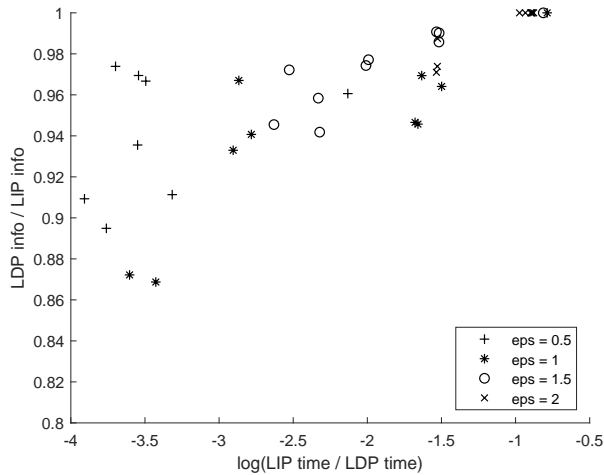


Figure 3. Comparison of computation time and  $I(X; Y)$  for  $\epsilon$ -LDP protocols found via Theorem 1 and  $\epsilon$ -LIP protocols found via Theorem 2, for random  $p_{S,X}$  with  $c = 2$ ,  $a = 5$ , and  $\epsilon \in \{0.5, 1, 1.5, 2\}$ .

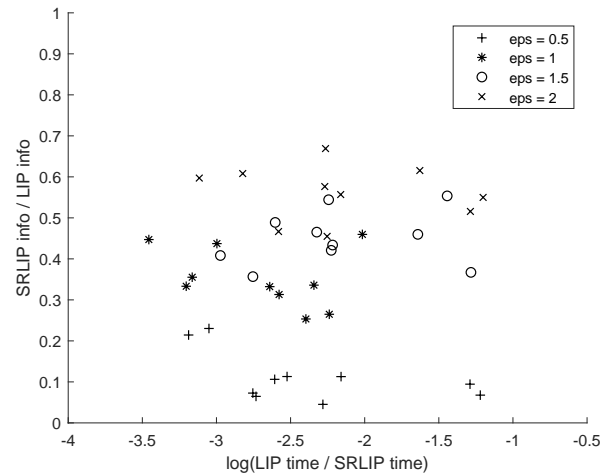


Figure 5. Comparison of computation time and  $I(X; Y)$  for  $\epsilon$ -(SR)LIP-protocols found via Theorems 2 and 3, for random  $p_{S,X}$  with  $c = 2$ ,  $a_1 = a_2 = 3$ ,  $a_3 = 4$ , and  $\epsilon \in \{0.5, 1, 1.5, 2\}$ .

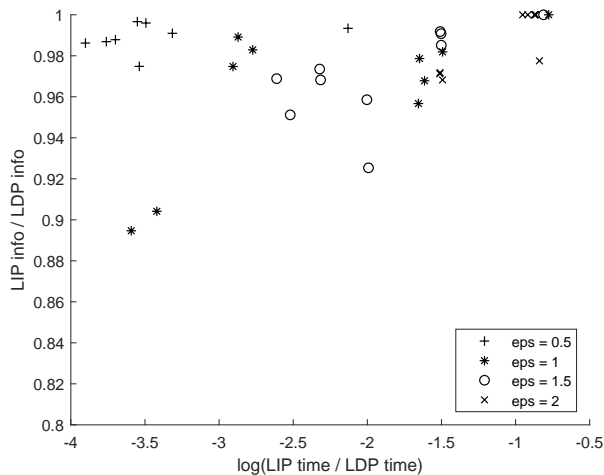


Figure 4. Comparison of computation time and  $I(X; Y)$  for  $\epsilon$ -LIP protocols found via Theorem 1 and  $\frac{\epsilon}{2}$ -LIP protocols found via Theorem 2, for random  $p_{S,X}$  with  $c = 2$ ,  $a = 5$ , and  $\epsilon \in \{0.5, 1, 1.5, 2\}$ .

advantage of using differential privacy-like privacy metrics is that they provide worst-case guarantees, ensuring that the privacy of every user is sufficiently protected. For both  $\epsilon$ -LDP and  $\epsilon$ -LIP we have found methods to find optimal sanitisation protocols. Within this setting, we have found that  $\epsilon$ -LIP has two main advantages over  $\epsilon$ -LDP. First, it fits better within the privacy funnel setting, where the distribution  $p_{S,X}$  is (at least approximately) known to the estimator. Second, finding the optimal protocol is significantly faster than under LDP, especially for small  $\epsilon$ . If one nevertheless prefers  $\epsilon$ -LDP as a privacy metric, then it is still worthwhile to find the optimal  $\frac{\epsilon}{2}$ -LIP protocol, as this can be found significantly faster, at a low utility cost.

In the multiple attributes setting, we have shown that  $\epsilon$ -SRLIP is a more sensible privacy metric than  $\epsilon$ -LIP, since

without this requirement a protocol can lose all its privacy protection in the presence of side channels. Unfortunately, however, experiments show that we pay for this both in computation time and in utility. Nevertheless, because of the robustness of  $\epsilon$ -SRLIP, it remains the preferred privacy notion in this setting.

For further research, two important avenues remain to be explored. First, the aggregator’s knowledge about  $p_{S,X}$  may not be perfect, because they may learn about  $p_{S,X}$  through observing  $(\vec{S}, \vec{X})$ . Incorporating this uncertainty leads to robust optimisation [3], which would give stronger privacy guarantees. Second, it might be possible to improve the method of obtaining  $\epsilon$ -SRLIP protocols via Theorem 3. Examining its proof shows that lower values of  $\epsilon^j$  may suffice to still ensure  $\epsilon$ -SRLIP. Furthermore, the optimal choice of  $(\epsilon^j)_{j \leq m}$  such that  $\sum_j \epsilon^j = \epsilon$  might not be  $\epsilon^j = \frac{\epsilon}{m}$ . However, it is computationally prohibitive to perform the vertex enumeration for many different choices of  $(\epsilon^j)_{j \leq m}$ , and as such a new theoretical approach is needed to determine the optimal  $(\epsilon^j)_{j \leq m}$  from  $\epsilon$  and  $p_{S,X}$ .

ACKNOWLEDGEMENTS

This work was supported by NWO grant 628.001.026 (Dutch Research Council, the Hague, the Netherlands). The author thanks Jasper Goseling and Boris Škorić for helpful discussions.

REFERENCES

- [1] D. Avis and K. Fukuda, “A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra,” *Discrete and Computational Geometry*, vol. 8, no. 1, pp. 174–190, 1992.
- [2] I. Bárány and A. Pór, “On 0-1 Polytopes with Many Facets,” *Advances in Mathematics*, vol. 161, no. 2, pp. 209–228, 2001.
- [3] D. Bertsimas, V. Gupta, and N. Kallus, “Data-driven robust optimization,” *Mathematical Programming*, vol. 167, no. 2, pp. 235–292, 2017.
- [4] F.P. Calmon et al., “Principal Inertia Components and Applications,” *IEEE Transactions on Information Theory*, vol. 63 no. 8, pp. 5011–5038, 2017.

- [5] P. Cuff and L. Yu, “Differential Privacy as a Mutual Information Constraint,” *ACM SIGSAC Conference on Computer and Communications Security 2016*, pp. 43–54, 2016.
- [6] N. Ding and P. Sadeghi, “A Submodularity-based Agglomerative Clustering Algorithm for the Privacy Funnel,” Unpublished preprint, *arXiv:1901.06629*, 2019 (retrieved: February, 2020).
- [7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” *Theory of Cryptography Conference*, pp. 265–284, 2006.
- [8] Ú. Erlingsson et al., “Encode, Shuffle, Analyze Privacy Revisited: Formalizations and Empirical Evaluation,” Unpublished preprint, *arXiv:2001.03618*, 2020 (retrieved: February, 2020).
- [9] B. Jiang, M. Li, and R. Tandon, “Local Information Privacy with Bounded Prior,” *IEEE International Conference on Communications*, pp. 1–7, 2019.
- [10] P. Kairouz, S. Oh, and P. Viswanath, “Extremal Mechanisms for Local Differential Privacy,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 2879–2887, 2014.
- [11] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, “What can we learn privately?,” *SIAM Journal of Computing*, vol. 40, no. 3, pp. 793–826, 2011.
- [12] D. Kifer and A. Machanavajjhala, “Pufferfish: A Framework for Mathematical Privacy Definitions,” *ACM Transactions on Database Systems*, vol. 39, no 1., pp. 1–36, 2014.
- [13] M. Lopuhaä-Zwakenberg, B. Škorić, and N. Li, “Information-theoretic metrics for Local Differential Privacy protocols,” Unpublished preprint, *arXiv:1910.07826*, 2019 (retrieved: February, 2020).
- [14] F. Prasser, F. Kohlmayer, R. Lautenschläger, and K. A. Kuhn, “ARX - A Comprehensive Tool for Anonymizing Biomedical Data,” *AMIA Annual Symposium Proceedings*, pp. 984–993, 2014.
- [15] B. Rassouli and D. Gündüz, “On Perfect Privacy and Maximal Correlation,” Unpublished preprint, *arXiv:1712.08500*, 2017 (retrieved: February, 2020).
- [16] S. Salamatian, F.P. Calmon, N. Fawaz, A. Makhdoumi, and M. Médard, “Privacy-Utility Tradeoff and Privacy Funnel,” Unpublished preprint, [http://www.mit.edu/~salmansa/files/privacy\\_TIFS.pdf](http://www.mit.edu/~salmansa/files/privacy_TIFS.pdf), 2020 (retrieved: February, 2020).

APPENDIX A  
PROOF OF THEOREM 3

For  $J \subset \{1, \dots, m\}$  and  $j \in \{1, \dots, m\}$ , we write  $J[j] := J \cup \{1, \dots, j - 1\}$ . Furthermore, we write  $\mathcal{X}^{\setminus J} = \prod_{j \notin J} \mathcal{X}^j$ , and its elements as  $x^{\setminus J}$ . We write  $\varepsilon := \sum_j \varepsilon^j$ . We then have

$$P_{y|s, x^J} = \sum_{x^{\setminus J}} P_{y|x} P_{x^{\setminus J}|s, x^J} \quad (17)$$

$$= P_{y^J|x^J} \sum_{x^{\setminus J}} \left( \prod_{j \notin J} P_{y^j|x^j} \right) P_{x^{\setminus J}|s, x^J} \quad (18)$$

$$= P_{y^J|x^J} \sum_{x^{\setminus J}} \prod_{j \notin J} P_{y^j|x^j} P_{x^j|s, x^J[j]} \quad (19)$$

$$= P_{y^J|x^J} \prod_{j \notin J} \sum_{x^j} P_{y^j|x^j} P_{x^j|s, x^J[j]} \quad (20)$$

$$= P_{y^J|x^J} \prod_{j \notin J} P_{y^j|s, x^J[j]} \quad (21)$$

$$\leq P_{y^J|x^J} \prod_{j \notin J} e^{\varepsilon^j} P_{y^j|x^J[j]} \quad (22)$$

$$\leq e^\varepsilon P_{y^J|x^J} \prod_{j \notin J} P_{y^j|x^J[j]} \quad (23)$$

$$= e^\varepsilon P_{y|x^J} . \quad (24)$$

The fact that  $e^{-\varepsilon} P_{y|x^J} \leq P_{y|s, x^J}$  is proven analogously.