Proving Differential Privacy via Probabilistic Couplings

Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, Pierre-Yves Strub

Proving Differential Privacy via Probabilistic Couplings

Gilles Barthe[⋆] Marco Gaboardi[‡] Benjamin Grégoire[$] Justin Hsu[#] Pierre-Yves Strub[⋆]

[⋆] IMDEA Software [‡] University at Buffalo, SUNY [$] Inria [#] University of Pennsylvania

Abstract

Over the last decade, *differential privacy* has achieved widespread adoption within the privacy community. Moreover, it has attracted significant attention from the verification community, resulting in several successful tools for formally proving differential privacy. Although their technical approaches vary greatly, all existing tools rely on reasoning principles derived from the *composition theorem* of differential privacy. While this suffices to verify most common private algorithms, there are several important algorithms whose privacy analysis does not rely solely on the composition theorem. Their proofs are significantly more complex, and are currently beyond the reach of verification tools.

In this paper, we develop compositional methods for formally verifying differential privacy for algorithms whose analysis goes beyond the composition theorem. Our methods are based on deep connections between differential privacy and *probabilistic couplings*, an established mathematical tool for reasoning about stochastic processes. Even when the composition theorem is not helpful, we can often prove privacy by a coupling argument.

We demonstrate our methods on two algorithms: the *Exponential mechanism* and the *Above Threshold* algorithm, the critical component of the famous *Sparse Vector* algorithm. We verify these examples in a relational program logic apRHL$^+$, which can construct approximate couplings. This logic extends the existing apRHL logic with more general rules for the Laplace mechanism and the one-sided Laplace mechanism, and new structural rules enabling pointwise reasoning about privacy; all the rules are inspired by the connection with coupling. While our paper is presented from a formal verification perspective, we believe that its main insight is of independent interest for the differential privacy community.

Categories and Subject Descriptors F.3.1 [*Specifying and Verifying and Reasoning about Programs*]

General Terms Differential privacy, probabilistic couplings

1. Introduction

Differential privacy is a rigorous definition of statistical privacy proposed by Dwork, McSherry, Nissim and Smith [14], and considered to be the gold standard for privacy-preserving computations. Most differentially private computations are built from two fundamental tools: private primitives and composition theorems (see § 2). However, there are several important examples whose privacy proofs go beyond these tools, for instance:

- The *Above Threshold* algorithm, which takes a list of numerical queries as input and outputs the first query whose answer is above a certain threshold. Above Threshold is the critical component of the Sparse Vector technique. (See, e.g., Dwork and Roth [12].)
- The *Report-noisy-max* algorithm, which takes a list of numerical queries as input and privately selects the query with the highest answer. (See, e.g., Dwork and Roth [12].)
- The *Exponential mechanism* [27], which privately returns the element of a (possibly non-numeric) range with the highest score; this algorithm can be implemented as a variant of the Report-noisy-max algorithm with a different noise distribution.

Unfortunately, existing pen-and-paper proofs of these algorithms use ad hoc manipulations of probabilities, and as a consequence are difficult to understand and error-prone.

This raises a natural question: can we develop *compositional proof methods* for verifying differential privacy of these algorithms, even though their proofs appear non-compositional? Surprisingly, the answer is yes. Our method builds on two key insights.

1. A connection between probabilistic liftings and probabilistic couplings [6].
2. A connection between differential privacy and *approximate liftings* [2, 4], a generalization of probabilistic liftings used in probabilistic process algebra [21].

Probabilistic liftings and couplings

Relation lifting is a well-studied construction in mathematics and computer science. Abstractly, relation lifting transforms relations $R \subseteq A \times B$ into relations $R^\sharp \subseteq TA \times TB$, where T is a functor over sets [1]. Relation lifting satisfies a type of composition, so it is a natural foundation for compositional proof methods.

Relation lifting has historically been an important tool for analyzing of probabilistic systems. For example, *probabilistic lifting* specializes the notion of relation lifting for the probability monad, and appears in standard definitions of probabilistic bisimulation. Over the last 25 years, researchers have developed a wide variety of tools for reasoning about probabilistic liftings, explored applications in numerous areas including security and biology, and uncovered deep connections with the Kantorovich metric and the theory of optimal transport (for a survey, see Deng and Du [11]).

While research has traditionally considers probabilistic liftings for partial equivalence relations, recent works investigate liftings for more general relations. Applications include formalizing reduction-based cryptographic proofs [3] and modeling stochastic dominance and convergence of probabilistic processes [6]. Seeking to explain the power of liftings, Barthe et al. [6] establish a tight connection between probabilistic liftings and *probabilistic couplings*, a basic

LICS '16 July 5–8, 2016, New York, New York, USA

ACM 978-1-4503-4391-6/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2933575.2934554

tool in probability theory [24, 30]. Roughly, a probabilistic coupling places two distributions in the same probabilistic space by exhibiting a suitable *witness distribution* over pairs. Not only does this observation open new uses for probabilistic liftings, it offers an opportunity to revisit existing applications from a fresh perspective.

Differential privacy via approximate probabilistic liftings

Relational program logics [2, 4] and relational refinement type systems [8] are currently the most flexible techniques for reasoning formally about differentially private computations. Their expressive power stems from *approximate probabilistic liftings*, a generalization of probabilistic liftings involving a metric on distributions. In particular, differential privacy is a consequence of a particular form of approximate lifting.

These approaches have successfully verified differential privacy for many algorithms. However, they are unsuccessful when privacy does not follow from standard tools and composition properties. In fact, the present authors had long believed that the verification of such examples was beyond the capabilities of lifting-based methods.

Contributions

In this paper, we propose the first formal analysis of differentially private algorithms whose proof does not exclusively rely on the basic tools of differential privacy. We make three broad contributions.

New proof principles for approximate liftings We take inspiration from the connection between liftings and coupling to develop new proof principles for approximate liftings.

First, we introduce a principle for decomposing proofs of differential privacy *pointwise*, supporting a common pattern of proving privacy separately for each possible output value. This principle is used in pen-and-paper proofs, but is new to formal approaches.

Second, we provide new proof principles for the Laplace mechanism. Informally speaking, existing proof principles capture the intuition that different inputs can be made to look equal by the Laplace mechanism in exchange for paying some privacy cost. Our first new proof principle for the Laplace mechanism is dual, and captures the idea that equal inputs can be made to look arbitrarily *different* by the Laplace mechanism, provided that one pays sufficient privacy. Our second new proof principle for the Laplace mechanism states that if we add the same noise in two runs of the Laplace mechanism, the distance between the two values is preserved and there is no privacy cost. As far as we know, these proof principles are new to the differential privacy literature. They are the key ingredients to proving examples such as Sparse Vector using compositional proof methods.

We also propose approximate probabilistic liftings for the one-sided Laplace mechanism, which can be used to implement the Exponential mechanism. The one-sided Laplace mechanism nicely illustrates the benefits of our approach: although it is not differentially private, its properties can be formally captured by approximate probabilistic liftings. These properties can be combined to show privacy for a larger program.

An extended probabilistic relational program logic To demonstrate our techniques, we work with the relational program logic apRHL [4]. Conceived as a probabilistic variant of Benton's relational Hoare logic [9], apRHL has been used to verify differential privacy for examples using the standard composition theorems. Most importantly, the semantics of apRHL uses approximate liftings. We introduce new proof rules representing our new proof principles, and call the resulting logic apRHL$^+$.

New privacy proofs While the extensions amount to just a handful of rules, they significantly increase the power of apRHL: We provide the first formal verification of two algorithms whose privacy proofs use tools beyond the composition theorems.

- The *Exponential mechanism*. The standard private algorithm when the output is non-numeric, this construction is typically taken as a primitive in systems verifying privacy. In contrast, we prove its privacy within our logic.
- The *Sparse Vector* algorithm. Perhaps the most famous example not covered by existing techniques, the proof of this mechanism is quite involved; some of its variants are not provably private. We also prove the privacy of its core subroutine in our logic.

The proofs are based on coupling ideas, which avoid reasoning about probabilities explicitly. As a consequence, proofs are clean, concise, and, we believe, appealing to researchers from both the differential privacy and the formal verification communities.

We have formalized the proofs of these algorithms in an experimental branch of the EasyCrypt proof assistant supporting approximate probabilistic liftings.

2. Differential privacy

In this section, we review the basic tools of differential privacy, and we present the algorithm Above Threshold, which forms the main subroutine of the Sparse Vector algorithm.

2.1 Basics

The basic definition of differential privacy is due to Dwork et al. [14].

Definition 1 (Differential privacy)**.** *A probabilistic computation $M : A \to \mathbf{Distr}(B)$ satisfies (ϵ, δ)*-differential privacy *w.r.t. an adjacency relation $\Phi \subseteq A \times A$ if for every pair of inputs $a, a' \in A$ such that $a\ \Phi\ a'$ and every subset of outputs $E \subseteq B$, we have*

$$\Pr_{y \leftarrow Ma}[y \in E] \leq \exp(\epsilon) \Pr_{y \leftarrow Ma'}[y \in E] + \delta.$$

*When $\delta = 0$, we say that M is ϵ-*differentially private*.*

Intuitively, the probabilistic condition ensures that any two inputs satisfying the adjacency relation Φ result in similar distributions over outputs. The relation Φ models which pairs of databases should be protected, i.e., what data should be nearly indistinguishable. While it may not be obvious from the definition, differential privacy has a number of features that allow simple construction of private algorithms with straightforward proofs of privacy. Specifically, the vast majority of differential privacy proofs use two basic tools: private primitives and composition theorems.

Private primitives These components form the building blocks of private algorithms. The most basic example is the *Laplace mechanism*, which achieves differential privacy for numerical computations by adding probabilistic noise to the output. We will work with the discrete version of this mechanism throughout the paper.

Definition 2 (Laplace mechanism [14])**.** *Let $\epsilon > 0$. The* (discrete) Laplace mechanism *$\mathcal{L}_\epsilon : \mathbb{Z} \to \mathbf{SDistr}(\mathbb{Z})$ is defined by $\mathcal{L}_\epsilon(t) = t + \nu$, where $\nu \in \mathbb{Z}$ is drawn from the Laplace distribution* $\mathrm{Laplace}(1/\epsilon)$*, i.e. with probabilities proportional to*

$$\Pr[\nu] \propto \exp\left(-\epsilon \cdot |\nu|\right).$$

The level of privacy depends on the sensitivity of the query.

Definition 3 (Sensitivity)**.** *Let $k \in \mathbb{N}$. A function $F : A \to \mathbb{Z}$ is k-*sensitive with respect to *$\Phi \subseteq A \times A$ if $|F(a_1) - F(a_2)| \leq k$ for every $a_1, a_2 \in A$ such that $a_1\ \Phi\ a_2$.*

The following theorem shows that k-sensitive functions can be made differentially private through the Laplace mechanism [14].

Theorem 1. *Assume that $F : A \to \mathbb{Z}$ is k-sensitive with respect to Φ. Let $M : A \to \mathbf{Distr}(\mathbb{Z})$ be the probabilistic function that maps*

$i \leftarrow 1; r \leftarrow |Q| + 1;$
$T \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(t);$
while $i < |Q|$ **do**
  $S \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(\mathsf{evalQ}(Q[i], d));$
  if $(T \leq S \wedge r = |Q| + 1)$ **then** $r \leftarrow i;$
  $i \leftarrow i + 1;$
return r

Figure 1. The Above Threshold algorithm

a to $\mathcal{L}_\epsilon(F(a))$. Then M is $k \cdot \epsilon$-differentially private with respect to Φ.

Another private primitive is the Exponential mechanism, which is the tool of choice when the desired output is non-numeric. While this mechanism is often taken as a primitive construct, we will see in § 5 how to verify its privacy.

Composition theorems These tools prove the privacy of a combination of private components, significantly simplifying the privacy analysis. The most commonly instance, by far, is the powerful *sequential composition theorem*.

Theorem 2 (Sequential composition [13])**.** *Let $M : D \to \mathbf{Distr}(R)$ be an (ϵ, δ)-private computation, and let $M' : D \to R \to \mathbf{Distr}(R')$ be an (ϵ', δ')-private computation in the first argument for any fixed value of the second argument. Then, the function*

$$d \mapsto bind\ M(d)\ M'(d)$$

is $(\epsilon + \epsilon', \delta + \delta')$-private.

One specific form of composition is post-processing. Informally, the post-processing theorem states that the output of a differentially private computation can be transformed while remaining private, so long as the transformation does not depend on the private data directly; such a transformation can be thought of as $(0, 0)$-differentially private.

2.2 Above Threshold

While most private algorithms can be analyzed using composition theorems and proofs of private primitives, some algorithms require more intricate proofs. To give an example, we consider the Above Threshold algorithm, which is the core of the Sparse Vector technique.[1] The Sparse Vector algorithm takes as input a database d, a list of numerical queries Q, a threshold t, and a natural number k, and privately selects the first k queries from Q whose output on d are approximately above the threshold. The Above Threshold algorithm corresponds to the case $k = 1$.

The code of the algorithm is given in Figure 1. In words, AboveT computes a noisy version T of the threshold t, computes for every query q in the list Q a noisy version S of $q(d)$, and returns the index of the first query q such that $T \leq S$ or a default value if there is no such query. It is easy to see that $(\epsilon, 0)$-differential privacy of AboveT directly implies $(k \cdot \epsilon, 0)$-differential privacy of Sparse Vector, since we can simply run AboveT k times in sequence and apply the sequential composition theorem.

If we try applying the sequential composition theorem (with the privacy of the Laplace mechanism) to AboveT we can show $(|Q| \cdot \epsilon, 0)$-differential privacy when all queries in Q are 1-sensitive, where $|Q|$ denotes the length of the list Q. However, a sophisticated analysis gives a more precise privacy guarantee.

[1] As this algorithm was not formally proposed in a canonical work, there exist different variants of the algorithm. Some variants take as input a stream rather than a list of queries, and/or output the result of a noisy query, rather than its index; see the final remark in § 6 for further discussion.

Theorem 3 (see, e.g., Dwork and Roth [12])**.** *Assuming all queries in Q are 1-sensitive, AboveT is $(\epsilon, 0)$-differentially private.*

In other words, AboveT is provably ϵ-differentially private, *independent of the number of queries*. This is a remarkable feature of the Above Threshold algorithm.

3. Generalized probabilistic liftings

To verify advanced algorithms like AboveT, we will leverage the power of *approximate probabilistic liftings*. In a nuthsell, our proofs will replace the sequential composition theorem of differential privacy—which we've seen is not enough to verify our target examples—with the more general composition principle of liftings. This section reviews existing notions of (approximate) probabilistic liftings and introduces proof principles for establishing their existence. Most of these proof principles are new, including those for equality (Proposition 2), differential privacy (Proposition 6), the Laplace mechanism (Propositions 8 and 9), and the one-sided Laplace mechanism (Propositions 10 and 11).

To avoid measure-theoretic issues, we base our technical development on sub-distributions over discrete sets (*discrete sub-distributions*). For simplicity, we will work with distributions over the integers when considering distributions over numeric values.

We start by reviewing the standard definition of sub-distributions. Let B be a countable set. A function $\mu : B \to \mathbb{R}^{\geq 0}$ is

- a *sub-distribution* over B if $\sum_{b \in \mathsf{supp}(\mu)} \mu(b) \leq 1$; and
- a *distribution* over B if $\sum_{b \in \mathsf{supp}(\mu)} \mu(b) = 1$.

As usual, the *support* $\mathsf{supp}(\mu)$ is the subset of B with non-zero weight under μ. Let $\mathbf{Distr}(B)$ and $\mathbf{SDistr}(B)$ denote the sets of discrete sub-distributions and distributions respectively over B. Equality of distributions is defined as pointwise equality of functions.

Probabilistic liftings and couplings are defined in terms of a distribution over products, and its *marginal distributions*. Formally, the first and second marginals of a sub-distribution $\mu \in \mathbf{Distr}(B_1 \times B_2)$ are simply the projections: the sub-distributions $\pi_1(\mu) \in \mathbf{Distr}(B_1)$ and $\pi_2(\mu) \in \mathbf{Distr}(B_2)$ given by

$$\pi_1(\mu)(b_1) = \sum_{b_2 \in B_2} \mu(b_1, b_2) \qquad \pi_2(\mu)(b_2) = \sum_{b_1 \in B_1} \mu(b_1, b_2).$$

3.1 Probabilistic couplings and liftings

Probabilistic couplings and liftings are standard tools in probability theory, and semantics and verification, respectively. We present their definitions to highlight their similarities before discussing some useful consequences.

Definition 4 (Coupling)**.** *There is a* coupling *between two sub-distributions $\mu_1 \in \mathbf{Distr}(B_1)$ and $\mu_2 \in \mathbf{Distr}(B_2)$ if there exists a sub-distribution (called the* witness*) $\mu \in \mathbf{Distr}(B_1 \times B_2)$ s.t. $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$.*

Probabilistic liftings are a special class of couplings.

Definition 5 (Lifting)**.** *Two sub-distributions $\mu_1 \in \mathbf{Distr}(B_1)$ and $\mu_2 \in \mathbf{Distr}(B_2)$ are related by the* (probabilistic) lifting *of $\Psi \subseteq B_1 \times B_2$, written $\mu_1 \Psi^\sharp \mu_2$, if there exists a coupling $\mu \in \mathbf{Distr}(B_1 \times B_2)$ of μ_1 and μ_2 such that $\mathsf{supp}(\mu) \subseteq \Psi$.*

Probabilistic liftings have many useful consequences. For example, $\mu_1 =^\sharp \mu_2$ holds exactly when the sub-distributions μ_1 and μ_2 are equal. Less trivially, liftings can bound the probability of one event by the probability of another event. This observation is useful for formalizing reduction-based cryptographic proofs.

Proposition 1 (Barthe et al. [3]). *Let $E_1 \subseteq B_1$, $E_2 \subseteq B_2$, $\mu_1 \in \mathbf{Distr}(B_1)$ and $\mu_2 \in \mathbf{Distr}(B_2)$. Define*

$$\Psi = \{(x_1, x_2) \in B_1 \times B_2 \mid x_1 \in E_1 \Rightarrow x_2 \in E_2\}.$$

If $\mu_1 \Psi^\sharp \mu_2$, then

$$\Pr_{x_1 \leftarrow \mu_1}[x_1 \in E_1] \leq \Pr_{x_2 \leftarrow \mu_2}[x_2 \in E_2].$$

One key observation for our approach is that this result can also be used to prove equality between distributions in a pointwise style.

Proposition 2 (Equality by pointwise lifting)**.**

- *Let $\mu_1, \mu_2 \in \mathbf{SDistr}(B)$. For every $b \in B$, define*

$$\Psi_b = \{(x_1, x_2) \in B \times B \mid \ x_1 = b \Rightarrow x_2 = b\}.$$

  If $\mu_1 \ \Psi_b^\sharp \ \mu_2$ for all $b \in B$, then $\mu_1 = \mu_2$.
- *Let $\mu_1, \mu_2 \in \mathbf{Distr}(B)$. For every $b \in B$, define*

$$\Psi_b = \{(x_1, x_2) \in B \times B \mid \ x_1 = b \Leftrightarrow x_2 = b\}.$$

  If $\mu_1 \ \Psi_b^\sharp \ \mu_2$ for all $b \in B$, then $\mu_1 = \mu_2$.

Proof. We prove the first item; the second item follows similarly.

First, a simple observation: two distributions μ_1 and μ_2 are equal iff $\mu_1(b) \leq \mu_2(b)$ for every $b \in B$. Indeed, suppose that $\mu_1(\bar{b}) \neq \mu_2(\bar{b})$ for some $\bar{b} \in B$. Then, $\mu_1(\bar{b}) < \mu_2(\bar{b})$, so

$$\sum_{b \in B} \mu_1(b) < \sum_{b \in B} \mu_2(b),$$

contradicting the fact that μ_1 and μ_2 are distributions:

$$\sum_{b \in B} \mu_1(b) = \sum_{b \in B} \mu_2(b) = 1.$$

Thus, in order to show $\mu_1 = \mu_2$, it is sufficient to prove $\Pr_{x \leftarrow \mu_1}[x = b] \leq \Pr_{x \leftarrow \mu_2}[x = b]$ for every $b \in B$. These inequalities follow from Proposition 1. □

3.2 Approximate liftings

It has previously been shown that differential privacy follows from an approximate version of liftings [4]. Our presentation follows subsequent refinements by Barthe and Olmedo [2]. We start by defining a notion of distance between sub-distributions.

Definition 6 (Barthe et al. [4])**.** *Let $\epsilon \geq 0$. The* ϵ-DP divergence *$\Delta_\epsilon(\mu_1, \mu_2)$ between two sub-distributions $\mu_1 \in \mathbf{Distr}(B)$ and $\mu_2 \in \mathbf{Distr}(B)$ is defined as*

$$\sup_{E \subseteq B} \left(\Pr_{x \leftarrow \mu_1}[x \in E] - \exp(\epsilon) \Pr_{x \leftarrow \mu_2}[x \in E] \right)$$

The following proposition relates ϵ-DP divergence with (ϵ, δ)-differential privacy.

Proposition 3 (Barthe et al. [4])**.** *A probabilistic computation $M : A \to \mathbf{Distr}(B)$ is (ϵ, δ)-differentially private w.r.t. an adjacency relation Φ iff*

$$\Delta_\epsilon(M(a), M(a')) \leq \delta$$

for every two adjacent inputs a and a' (i.e. such that $a \ \Phi \ a'$).

We can use DP-divergence to define an approximate version of probabilistic lifting, called (ϵ, δ)-*lifting*. We adopt the definition by Barthe and Olmedo [2], which extends to a general class of distances called f-divergences.

Definition 7 ((ϵ, δ)-lifting)**.** *Two sub-distributions $\mu_1 \in \mathbf{Distr}(B_1)$ and $\mu_2 \in \mathbf{Distr}(B_2)$ are related by the* (ϵ, δ)-lifting *of $\Psi \subseteq B_1 \times B_2$, written $\mu_1 \Psi^{\sharp(\epsilon,\delta)} \mu_2$, if there exist two witness sub-distributions $\mu_L \in \mathbf{Distr}(B_1 \times B_2)$ and $\mu_R \in \mathbf{Distr}(B_1 \times B_2)$ such that*

1. *$\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$;*
2. *$\mathsf{supp}(\mu_L) \subseteq \Psi$ and $\mathsf{supp}(\mu_R) \subseteq \Psi$; and*
3. *$\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$.*

It is relatively easy to see that two sub-distributions μ_1 and μ_2 are related by $=^{\sharp(\epsilon,\delta)}$ iff $\Delta_\epsilon(\mu_1, \mu_2) \leq \delta$. Therefore, a probabilistic computation $M : A \to \mathbf{Distr}(B)$ is (ϵ, δ)-differentially private w.r.t. an adjacency relation Φ iff

$$M(a) \ =^{\sharp(\epsilon,\delta)} \ M(a')$$

for every two adjacent inputs a and a' (i.e. such that $a \ \Phi \ a'$). This fact forms the basis of previous lifting-based approaches for differential privacy [2, 4, 5, 8].

A useful preliminary fact is that approximate liftings generalize probabilistic liftings (which we will sometimes call *exact* liftings).

Proposition 4. *Suppose we are given distributions $\mu_1 \in \mathbf{SDistr}(B_1)$ and $\mu_2 \in \mathbf{SDistr}(B_2)$ and a relation $\Psi \subseteq B_1 \times B_2$. Then, $\mu_1 \Psi^\sharp \mu_2$ if and only if $\mu_1 \Psi^{\sharp(0,0)} \mu_2$.*

Proof. The forward direction is easy: simply define $\mu_L = \mu_R$ to be the witness of the exact lift. The reverse direction follows from the observations that the witnesses μ_L and μ_R are necessarily distributions, and that Δ_0 is the total variation distance (a.k.a. statistical distance) on distributions, in particular $\Delta_0(\mu_L, \mu_R) = 0$ iff $\mu_L = \mu_R$. To see this last point, $\Delta_0(\mu_L, \mu_R) = 0$ entails

$$\mu_L(b_1, b_2) \leq \mu_R(b_1, b_2)$$

for every $(b_1, b_2) \in B_1 \times B_2$. So $\mu_L = \mu_R$ by Proposition 2. □

The previous results for exact liftings generalize smoothly to approximate liftings. First, we can generalize Proposition 1.

Proposition 5 (Barthe and Olmedo [2])**.** *Let $E_1 \subseteq B_1$, $E_2 \subseteq B_2$, $\mu_1 \in \mathbf{Distr}(B_1)$ and $\mu_2 \in \mathbf{Distr}(B_2)$. Let*

$$\Psi = \{(x_1, x_2) \in B_1 \times B_2 \mid x_1 \in E_1 \Rightarrow x_2 \in E_2\}.$$

If $\mu_1 \Psi^{\sharp(\epsilon,\delta)} \mu_2$, then

$$\Pr_{x_1 \leftarrow \mu_1}[x_1 \in E_1] \leq \exp(\epsilon) \Pr_{x_2 \leftarrow \mu_2}[x_2 \in E_2] + \delta.$$

We can use this proposition to generalize Proposition 2, which provides a way to prove that two distributions μ_1 and μ_2 are equal—equivalently, $\mu_1 =^\sharp \mu_2$. Generalizing this lifting from exact to approximate yields the following pointwise characterization of differential privacy, a staple technique of pen-and-paper proofs.

Proposition 6 (Differential privacy from pointwise lifting)**.** *A probabilistic computation $M : A \to \mathbf{Distr}(B)$ is (ϵ, δ)-differentially private w.r.t. an adjacency relation Φ iff there exists $(\delta_b)_{b \in B} \in \mathbb{R}^{\geq 0}$ such that $\sum_{b \in B} \delta_b \leq \delta$, and $M(a) \ \Psi_b^{\sharp(\epsilon,\delta_b)} \ M(a')$ for every $b \in B$ and every two adjacent inputs a and a', where*

$$\Psi_b = \{(x_1, x_2) \in B \times B \mid x_1 = b \Rightarrow x_2 = b\}.$$

Proof. First note that $\Delta_\epsilon(\mu_1, \mu_2) \leq \delta$ iff there exists $(\delta_b)_{b \in B} \in \mathbb{R}^{\geq 0}$ s.t. $\mu_1(b) \leq \exp(\epsilon)\mu_2(b) + \delta_b$ for every $b \in B$, and $\sum_{b \in B} \delta_b \leq \delta$. So, it is sufficient to show that for every $b \in B$ and every two adjacent inputs a and a', we have

$$\Pr_{x \leftarrow M(a)}[x = b] \leq \exp(\epsilon) \Pr_{x \leftarrow M(a')}[x = b] + \delta_b$$

with $\sum_{b \in B} \delta_b \leq \delta$. This follows from Proposition 5. □

3.3 Probabilistic liftings for the Laplace mechanism

So far, we have seen general properties about approximate liftings and differential privacy. Now, we turn to more specific liftings relevant to typical distributions in differential privacy. In terms

of approximate liftings, we can state the privacy of the Laplace mechanism (Theorem 1)in the following form.

Proposition 7. *Let $v_1, v_2 \in \mathbb{Z}$ and $k \in \mathbb{N}$ s.t. $|v_1 - v_2| \leq k$. Then $\mathcal{L}_\epsilon(v_1) \ =^{\sharp(k\cdot\epsilon,0)} \ \mathcal{L}_\epsilon(v_2)$.*

Proposition 7 is sufficiently general to capture most examples from the literature, but not for the examples of this paper; informally, applying Proposition 7 only allows us to prove privacy using the standard composition theorems. To see how we might generalize the principle, note that privacy from pointwise liftings (Proposition 6) involves liftings of an *asymmetric* relation, rather than equality. This suggests that it could be profitable to consider asymmetric liftings. Indeed, we propose the following generalization of Proposition 7.

Proposition 8. *Let $v_1, v_2, k \in \mathbb{Z}$. Then*

$$\mathcal{L}_\epsilon(v_1) \ \Psi^{\sharp(|k+v_1-v_2|\cdot\epsilon,0)} \ \mathcal{L}_\epsilon(v_2),$$

where

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 + k = x_2\}.$$

Proof. It suffices to prove $\mu_1 \ \Psi^{\sharp(|k+v_1-v_2|\cdot\epsilon,0)} \ \mu_2$, where μ_1 is the distribution of $v_1 + \eta_1 + k$ and μ_2 is the distribution of $v_2 + \eta_2$, with η_1, η_2 draws from the discrete Laplace distribution $\text{Laplace}(1/\epsilon)$. By the definition of the Laplace mechanism, $\mu_1 = \mathcal{L}_\epsilon(v_1 + k)$ and $\mu_2 = \mathcal{L}_\epsilon(v_2)$. Now, we can conclude by Proposition 7. □

Proposition 8 has several useful consequences. For instance, when $|v_1 - v_2| \leq k$ we have $\mathcal{L}_\epsilon(v_1) \ \Psi^{\sharp(2k\cdot\epsilon,0)} \ \mathcal{L}_\epsilon(v_2)$ with

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 + k = x_2\}, \tag{1}$$

following from Proposition 8 and the triangle inequality

$$|v_1 - v_2| \leq k \Rightarrow |k + (v_1 - v_2)| \leq k + k = 2k.$$

Informally, this instance of Proposition 8 shows that by "paying" privacy cost ϵ, we can ensure that the samples are a certain distance apart. This stands in contrast to Proposition 7, which ensures that the samples are equal.

Another useful consequence is that adding identical noise to both v_1 and v_2 incurs no privacy cost, and we can assume the difference between the samples is the difference between v_1 and v_2.

Proposition 9. *Let $v_1, v_2 \in \mathbb{Z}$. Then $\mathcal{L}_\epsilon(v_1) \ \Psi^{\sharp(0,0)} \ \mathcal{L}_\epsilon(v_2)$, where*

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 - x_2 = v_1 - v_2\}.$$

Proof. Immediate by Proposition 8 with $k = v_2 - v_1$. □

3.4 Probabilistic liftings for one-sided Laplace mechanism

While the Laplace mechanism is already sufficient to implement a wide variety of private algorithms, a few algorithms use other distributions. In particular, the Exponential mechanism can be implemented in terms of the *one-sided Laplace* mechanism. This algorithm is the same as the Laplace mechanism except noise is drawn from the *one-sided Laplace distribution* (also called the *exponential distribution*), which outputs non-negative integers.

Definition 8 (One-sided Laplace mechanism)**.** *Let $\epsilon > 0$. The* discrete one-sided Laplace mechanism $\mathcal{L}^{\text{os}}_\epsilon : \mathbb{Z} \to \mathbf{SDistr}(\mathbb{Z})$ *is defined by*

$$\mathcal{L}^{\text{os}}_\epsilon(t) = t + \nu,$$

where ν non-negative integer drawn from the Laplace distribution $\text{Laplace}^+(1/\epsilon)$*, i.e. with probabilities proportional to*

$$\Pr[\nu] \propto \exp(-\epsilon \cdot \nu).$$

While this mechanism is not ϵ-differentially private for any ϵ, we can still consider probabilistic liftings for the samples. We have the following two results, analogous to Propositions 8 and 9.

Proposition 10. *Let $v_1, v_2, k \in \mathbb{Z}$ such that $k \geq v_2 - v_1$. Then*

$$\mathcal{L}^{\text{os}}_\epsilon(v_1) \ \Psi^{\sharp((k+v_1-v_2)\cdot\epsilon,0)} \ \mathcal{L}^{\text{os}}_\epsilon(v_2),$$

where

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 + k = x_2\}.$$

Proof. It suffices to consider the case where $v_1 = v_2 = 0$: $\mathcal{L}^{\text{os}}_\epsilon(v)$ is the same distribution as sampling from $\mathcal{L}^{\text{os}}_\epsilon(0)$ and adding v, so the desired conclusion follows from

$$\mathcal{L}^{\text{os}}_\epsilon(0) \ \Psi'^{\sharp((k+v_1-v_2)\cdot\epsilon,0)} \ \mathcal{L}^{\text{os}}_\epsilon(0),$$

where

$$\begin{aligned}\Psi' &= \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid (x_1 + v_1) + k = (x_2 + v_2)\} \\ &= \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 + (k + v_1 - v_2) = x_2\},\end{aligned}$$

which follows from the $v_1 = v_2 = 0$ case since $k + v_1 - v_2 \geq 0$ by assumption.

So, we assume $v_1 = v_2 = 0$ and $k \geq 0$. We will directly define the two witnesses of the approximate lifting. Let

$$G(v) = \Pr_{x \leftarrow \mathcal{L}^{\text{os}}_\epsilon(0)}[x = v].$$

Define the left witness μ_L on its support by

$$\mu_L(i - k, i) = G(i)$$

for $i \geq k$, and the right witness μ_R on its support by

$$\mu_R(j, j + k) = G(j)$$

for $j \geq -k$. Evidently the marginals are correct—$\pi_1(\mu_L) = \pi_2(\mu_R) = \mathcal{L}^{\text{os}}_\epsilon(0)$—so it remains to check that $\Delta_{k\epsilon}(\mu_L, \mu_R) \leq 0$:

$$\max_{E \subseteq \mathbb{Z}\times\mathbb{Z}} \left(\Pr_{x \leftarrow \mu_L}[x \in E] - \exp(k\epsilon) \Pr_{x \leftarrow \mu_R}[x \in E] \right) \leq 0.$$

It suffices to prove this pointwise over the union of the supports of μ_L and μ_R: for each $j \geq -k$, we need

$$\Pr_{x \leftarrow \mu_L}[x = (j, j + k)] - \exp(k\epsilon) \Pr_{x \leftarrow \mu_R}[x = (j, j + k)] \leq 0.$$

This is evident for $j < 0$, when the first term is zero and the second term is non-negative. For $j \geq 0$ we need to show

$$G(j) - \exp(k\epsilon) G(j + k) \leq 0,$$

which follows by direct calculation (or, the privacy of the standard Laplace distribution). □

Proposition 11. *Let $v_1, v_2 \in \mathbb{Z}$. Then $\mathcal{L}^{\text{os}}_\epsilon(v_1) \ \Psi^{\sharp(0,0)} \ \mathcal{L}^{\text{os}}_\epsilon(v_2)$, where*

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 - x_2 = v_1 - v_2\}.$$

Proof. It suffices to prove

$$\mathcal{L}^{\text{os}}_\epsilon(v_1) \ \Psi'^{\sharp(0,0)} \ \mathcal{L}^{\text{os}}_\epsilon(v_2),$$

where

$$\Psi' = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 - v_1 = x_2 - v_2\}.$$

This is equivalent to

$$\mathcal{L}^{\text{os}}_\epsilon(v_1 - v_1) \ =^{\sharp(0,0)} \ \mathcal{L}^{\text{os}}_\epsilon(v_2 - v_2),$$

which is obvious by Proposition 4 since both sides are the same distribution. □

$$\vdash x_1 \leftarrow e_1 \sim_{\langle 0,0\rangle} x_2 \leftarrow e_2 : \Psi\{e_1\langle 1\rangle, e_2\langle 2\rangle / x_1\langle 1\rangle, x_2\langle 2\rangle\} \Longrightarrow \Psi \text{[ASSN]}$$

$$\frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \wedge b_1\langle 1\rangle \Longrightarrow \Psi \quad \vdash d_1 \sim_{\langle \epsilon,\delta\rangle} d_2 : \Phi \wedge \neg b_1\langle 1\rangle \Longrightarrow \Psi}{\vdash \mathsf{if}\ b_1\ \mathsf{then}\ c_1\ \mathsf{else}\ d_1 \sim_{\langle \epsilon,\delta\rangle} \mathsf{if}\ b_2\ \mathsf{then}\ c_2\ \mathsf{else}\ d_2 : \Phi \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \Longrightarrow \Psi} \text{[COND]}$$

$$\frac{\vdash c_1 \sim_{\langle \epsilon_k,\delta_k\rangle} c_2 : \Theta \wedge b_1\langle 1\rangle \wedge b_2\langle 2\rangle \wedge k = e\langle 1\rangle \wedge e\langle 1\rangle \leq n \Longrightarrow \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge k < e\langle 1\rangle \qquad \Theta \wedge e\langle 1\rangle \leq 0 \Rightarrow \neg b_1\langle 1\rangle}{\vdash \mathsf{while}\ b_1\ \mathsf{do}\ c_1 \sim_{\langle \sum_{k=1}^{n} \epsilon_k, \sum_{k=1}^{n} \delta_k\rangle} \mathsf{while}\ b_2\ \mathsf{do}\ c_2 : \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle \leq n \Longrightarrow \Theta \wedge \neg b_1\langle 1\rangle \wedge \neg b_2\langle 2\rangle} \text{[WHILE]}$$

$$\frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi' \quad \vdash c_1' \sim_{\langle \epsilon',\delta'\rangle} c_2' : \Psi' \Longrightarrow \Psi}{\vdash c_1; c_1' \sim_{\langle \epsilon+\epsilon',\delta+\delta'\rangle} c_2; c_2' : \Phi \Longrightarrow \Psi} \text{[SEQ]}$$

$$\frac{\vdash c_1 \sim_{\langle \epsilon',\delta'\rangle} c_2 : \Phi' \Longrightarrow \Psi' \qquad \Phi \Rightarrow \Phi' \qquad \Psi' \Rightarrow \Psi \qquad \epsilon' \leq \epsilon \qquad \delta' \leq \delta}{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi} \text{[CONSEQ]}$$

Figure 2. Proof rules from apRHL

4. Formalization in a program logic

In this section we present a new program logic called apRHL$^+$ for reasoning about differential privacy of programs written in a core programming language with samplings from the Laplace mechanism and the one-sided Laplace Mechanism. Our program logic apRHL$^+$ extends apRHL, a relational Hoare logic that has been used to verify many examples of differentially private algorithms [4]. The main result of this section is a proof of soundness of the logic (Theorem 4).

Programs We consider a simple imperative language with random sampling. The set of commands is defined inductively:

$\mathcal{C} ::=$	skip	noop
	$\mathcal{C}; \mathcal{C}$	sequencing
	$\mathcal{X} \leftarrow \mathcal{E}$	deterministic assignment
	$\mathcal{X} \xleftarrow{\$} \mathcal{L}_\epsilon(\mathcal{E})$	Laplace mechanism
	$\mathcal{X} \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_\epsilon(\mathcal{E})$	one-sided Laplace mechanism
	$\mathsf{if}\ \mathcal{E}\ \mathsf{then}\ \mathcal{C}\ \mathsf{else}\ \mathcal{C}$	conditional
	$\mathsf{while}\ \mathcal{E}\ \mathsf{do}\ \mathcal{C}$	while loop

where $\mathcal{X}$ is a set of *variables* and $\mathcal{E}$ is a set of *expressions*. Variables and expressions are typed, and range over boolean, integers, databases, queries, and lists.

The semantics of programs is standard [4, 22]. We first define the set Mem of memories to contain all well-typed functions from variables to values. Expressions and distribution expressions map memories to values and distributions over values, respectively: an expression e of type T is interpreted as a function $[\![e]\!] : \mathsf{Mem} \to T$, whereas a distribution expression g is interpreted as a function $[\![g]\!] : \mathsf{Mem} \to \mathbf{SDistr}(\mathbb{Z})$. Finally, commands are interpreted as functions from memories to sub-distributions over memories, i.e. the interpretation of c is a function $[\![c]\!] : \mathsf{Mem} \to \mathbf{Distr}(\mathsf{Mem})$. We refer to Barthe et al. [4], Kozen [22] for an account of the semantics.

Assertions and judgments Assertions in the logic are first-order formulae over generalized expressions. The latter are expressions built from tagged variables $x\langle 1\rangle$ and $x\langle 2\rangle$, where the tag is used to determine whether the interpretation of the variable is taken in the first memory or in the second memory. For instance, $x\langle 1\rangle = x\langle 2\rangle + 1$ is the assertion which states that the interpretation of the variable x in the first memory is equal to the interpretation of the variable x in the second memory plus 1. More formally, assertions are interpreted as predicates over pairs of memories. We let $[\![\Phi]\!]$ denote the set of memories (m_1, m_2) that satisfy Φ. The interpretation is standard (besides the use of tagged variables) and is omitted. By abuse of notation, we write $e\langle 1\rangle$ or $e\langle 2\rangle$, where e is a program expression, to denote the generalized expression built according to e, but in which all variables are tagged with a $\langle 1\rangle$ or $\langle 2\rangle$, respectively.

Judgments in both apRHL and apRHL$^+$ are of the form

$$\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi$$

where c_1 and c_2 are statements, the precondition Φ and postcondition Ψ are relational assertions, and ϵ and δ are non-negative reals.[2] Informally, a judgment of the above form is valid if the two distributions produced by the executions of c_1 and c_2 on any two initial memories satisfying the precondition Φ are related by the (ϵ, δ)-lifting of the postcondition Ψ. Formally, the judgment

$$\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi$$

is *valid* iff for every two memories m_1 and m_2, such that $m_1\ [\![\Phi]\!]\ m_2$, we have

$$([\![c_1]\!]_{m_1})\ [\![\Psi]\!]^{\sharp(\epsilon,\delta)}\ ([\![c_2]\!]_{m_2}).$$

Proof system Figure 2 presents the main rules from apRHL excluding the sampling rule, which we generalize in apRHL$^+$. We briefly comment on some of these rules.

The rule [SEQ] for sequential composition generalizes the sequential composition theorem of differential privacy, which intuitively corresponds to the case where the postcondition of the composed commands is equality. This generalization allows apRHL to prove differential privacy using the coupling composition principle when the standard composition theorem is insufficient.

The rule [WHILE] for while loops can be seen as a generalization of a k-fold composition theorem for differential privacy. Again, it allows to consider arbitrary postconditions, whereas the composition theorem would correspond to the case where the postcondition of the loop is equality (in conjunction with negation of the guards). We often use two simpler instances of the rule. The first one corresponds to the case where the values of ϵ_k and δ_k are independent of k, i.e. $\epsilon_k = \epsilon$ and $\delta_k = \delta$, yielding a bound of $\langle n \cdot \epsilon, n \cdot \delta\rangle$. The second one corresponds to the case where a single iteration carries a privacy cost, as shown in the rule [WHILEEXT] in Figure 4. This weaker rule is in fact sufficient for proving privacy of several of our examples, including the Above Threshold algorithm (but not the Sparse Vector algorithm, which also uses the aforementioned instance of the while rule), the Exponential mechanism, and Report-noisy-max.

[2] The original apRHL rules are based on a multiplicative privacy budget. We adapt the rules to an additive privacy parameter for consistency with the rest of the article and the broader privacy literature.

$$\frac{\forall i.\ \vdash c_1 \sim_{\langle \epsilon,\delta_i\rangle} c_2 : \Phi \Longrightarrow x\langle 1\rangle = i \Rightarrow x\langle 2\rangle = i \qquad \sum_{i\in I}\delta_i \le \delta}{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow x\langle 1\rangle = x\langle 2\rangle}\,[\textsc{Forall-Eq}]$$

$$\frac{}{\vdash y_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\langle k'\cdot\epsilon,0\rangle} y_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) : |k + e_1\langle 1\rangle - e_2\langle 2\rangle| \le k' \Longrightarrow y_1\langle 1\rangle + k = y_2\langle 2\rangle}\,[\textsc{LapGen}]$$

$$\frac{y_1 \notin FV(e_1) \qquad y_2 \notin FV(e_2)}{\vdash y_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e_1) \sim_{\langle 0,0\rangle} y_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e_2) : \top \Longrightarrow y_1\langle 1\rangle - y_2\langle 2\rangle = e_1\langle 1\rangle - e_2\langle 2\rangle}\,[\textsc{LapNull}]$$

$$\frac{}{\vdash y_1 \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_\epsilon(e_1) \sim_{\langle k'\cdot\epsilon,0\rangle} y_2 \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_\epsilon(e_2) : 0 \le k + e_1\langle 1\rangle - e_2\langle 2\rangle \le k' \Longrightarrow y_1\langle 1\rangle + k = y_2\langle 2\rangle}\,[\textsc{OneLapGen}]$$

$$\frac{y_1 \notin FV(e_1) \qquad y_2 \notin FV(e_2)}{\vdash y_1 \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_\epsilon(e_1) \sim_{\langle 0,0\rangle} y_2 \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_\epsilon(e_2) : \top \Longrightarrow y_1\langle 1\rangle - y_2\langle 2\rangle = e_1\langle 1\rangle - e_2\langle 2\rangle}\,[\textsc{OneLapNull}]$$

$$\frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c : \Phi \wedge b_1\langle 1\rangle \Longrightarrow \Psi \quad \vdash d_1 \sim_{\langle \epsilon,\delta\rangle} c : \Phi \wedge \neg b_1\langle 1\rangle \Longrightarrow \Psi}{\vdash \mathsf{if}\ b_1\ \mathsf{then}\ c_1\ \mathsf{else}\ d_1 \sim_{\langle \epsilon,\delta\rangle} c : \Phi \Longrightarrow \Psi}\,[\textsc{Cond-L}]$$

$$\frac{\vdash c \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \wedge b_2\langle 2\rangle \Longrightarrow \Psi \quad \vdash c \sim_{\langle \epsilon,\delta\rangle} d_2 : \Phi \wedge \neg b_2\langle 2\rangle \Longrightarrow \Psi}{\vdash c \sim_{\langle \epsilon,\delta\rangle} \mathsf{if}\ b_2\ \mathsf{then}\ c_2\ \mathsf{else}\ d_2 : \Phi \Longrightarrow \Psi}\,[\textsc{Cond-R}]$$

Figure 3. Proof rules from apRHL$^+$

Figure 3 collects the new rules in apRHL$^+$, which are all derived from the new proof principles we saw in the previous section. The first rule [Forall-Eq] allows proving differential privacy via pointwise privacy; this rule reflects Proposition 6.

The next pair of rules, [LapGen] and [LapNull], reflect the liftings of the distributions of the Laplace mechanism presented in Propositions 8 and 9 respectively. Note that we need a side-condition on the free variables in [LapNull]—otherwise, the sample may change e_1 and e_2.The following pair of rules, [OneLapGen] and [OneLapNull], give similar liftings for the one-sided Laplace mechanism following Propositions 10 and 11 respectively.

Finally, the last pair of rules allows reasoning about a conditional while treating the other command abstractly. These so-called *one-sided* rules were already present in the logic pRHL, a predecessor of apRHL based on exact liftings [3], but they were never needed in apRHL. In apRHL$^+$ the one-sided rules are quite useful, in conjunction with our richer sampling rules, for reasoning about two conditionals that may take different branches.

Soundness The soundness of the new rules immediately follows from the results of the previous section, while soundness for the apRHL rules was established previously [4].

Theorem 4. *All judgments derivable in apRHL$^+$ are valid.*

5. Exponential mechanism

In this section, we provide a formal proof of the *Exponential mechanism* of McSherry and Talwar [27]. While there is existing work that proves differential privacy of this mechanism [4], the proofs operate on the raw denotational semantics. In contrast, we work entirely within our program logic.

The Exponential mechanism is designed to privately compute the best response from a set $\mathcal{R}$ of possible response, according to some integer-valued *quality score* function qscore that takes as input an element in $\mathcal{R}$ and a database d. Given a database d and a k-sensitive quality score function qscore, the Exponential mechanism $\mathsf{ExpM}(d, \mathsf{qscore})$ outputs an element r of the range $\mathcal{R}$ with probability proportional to

$$\Pr[r] \propto \exp\left(\frac{\epsilon \cdot \mathsf{qscore}(r,d)}{2k}\right).$$

The shape of the distribution ensures that the Exponential mechanism favors elements with higher quality scores.

The seminal result of McSherry and Talwar [27] establishes differential privacy for this mechanism.

Theorem 5. *Assume that the quality score is* 1*-sensitive, i.e. for every output r and adjacent databases d, d',*

$$|\mathsf{qscore}(r,d) - \mathsf{qscore}(r,d')| \le 1.$$

Then the probabilistic computation that maps d to $\mathsf{ExpM}(d, \mathsf{qscore})$ *is $(\epsilon, 0)$-differentially private.*

While there does not seem to be much of a program to verify, it is known that the Exponential mechanism can be implemented more explicitly in terms of the one-sided Laplace mechanism [12]. Informally, the code loops through all the possible output values, adding one-sided Laplace noise to the quality score for the value/database pair. Throughout the computation, the code tracks the current highest noisy score and the corresponding element. Finally, it returns the top element. For the sake of simplicity we assume that $\mathcal{R} = \{1, \ldots, R\}$ for some $R \in \mathbb{N}$; generalizing to an arbitrary finite set poses little difficulty for the verification. Figure 5 shows the code of the implementation.

Informal proof The privacy proof for the Exponential mechanism cannot follow from the composition theorems of differential privacy—the one-sided Laplace noise does not satisfy differential privacy, so there is nothing to compose. Nonetheless, we can still show $(\epsilon, 0)$-differential privacy using our lifting-based techniques. By Proposition 6, it suffices to show that for every integer i and quality score qscore, the output of ExpM on two adjacent databases yields sub-distributions on memories that are related by the $(\epsilon, 0)$-lifting of the interpretation of the assertion

$$max\langle 1\rangle = i \Rightarrow max\langle 2\rangle = i.$$

We outline a coupling argument. First, we consider iterations of the loop body in which the loop counter r satisfies $r < i$. In this case,

$$\frac{\vdash c_1 \sim_{\langle 0,0\rangle} c_2 : \Phi \wedge i < e\langle 1\rangle \Longrightarrow \Psi \qquad \vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \wedge e\langle 1\rangle = i \Longrightarrow \Psi \qquad \vdash c_1 \sim_{\langle 0,0\rangle} c_2 : \Phi \wedge e\langle 1\rangle < i \Longrightarrow \Psi}{\Theta \wedge e\langle 1\rangle \leq 0 \Rightarrow \neg b_1\langle 1\rangle \qquad \Phi \triangleq \Theta \wedge b_1\langle 1\rangle \wedge b_2\langle 2\rangle \wedge k = e\langle 1\rangle \qquad \Psi \triangleq \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge k < e\langle 1\rangle}{\vdash \mathsf{while}\ b_1\ \mathsf{do}\ c_1 \sim_{\langle \epsilon,\delta\rangle} \mathsf{while}\ b_2\ \mathsf{do}\ c_2 : \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle \leq n \Longrightarrow \Theta \wedge \neg b_1\langle 1\rangle \wedge \neg b_2\langle 2\rangle}\ [\textsc{WhileExt}]$$

(Note that the two premises for $i < e\langle 1\rangle$ and $i > e\langle 1\rangle$ can be combined. However, we often use different reasoning for these cases, so we prefer to present the rule with 3 premises.)

Figure 4. Specialized proof rule for while loops

$r \leftarrow 1; bq \leftarrow 0;$
$\mathsf{while}\ r \leq R\ \mathsf{do}$
  $cq \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_{\epsilon/2}(\mathsf{qscore}(d, r));$
  $\mathsf{if}\ (cq > bq \vee r = 1)\ \mathsf{then}\ max \leftarrow r; bq \leftarrow cq;$
  $r \leftarrow r + 1;$
$\mathsf{return}\ max$

Figure 5. Implementation of the Exponential mechanism

we couple the two samplings using the rule [ONELAPNULL], using adjacency of the two databases and 1-sensitivity of the quality score function to establish the $(0, 0)$-lifting:

$$max\langle 1\rangle < i \ \wedge\ max\langle 2\rangle < i \wedge |bq\langle 1\rangle - bq\langle 2\rangle| \leq 1.$$

The interesting case is $r = i$. In this case, we use the rule [ONELAPGEN] to couple the random samplings so that

$$cq\langle 1\rangle + 1 = cq\langle 2\rangle.$$

This coupling has privacy cost $(\epsilon, 0)$ and ensures that the following $(\epsilon, 0)$-lifting holds at the end of the ith iteration:

$$(max\langle 1\rangle = max\langle 2\rangle = i \wedge bq\langle 1\rangle + 1 = bq\langle 2\rangle) \vee max\langle 1\rangle \neq i$$

Using the rule [ONELAPNULL] repeatedly, we couple the random samplings from the remaining iterations to prove that the above $(\epsilon, 0)$-lifting remains valid through subsequent iterations—note that couplings for iterations beyond i incur no privacy cost. Finally, we apply the rule of consequence to conclude the desired $(\epsilon, 0)$-lifting:

$$max\langle 1\rangle = i \Rightarrow max\langle 2\rangle = i$$

Formal proof We prove the following apRHL^+ judgment, which entails $(\epsilon, 0)$-differential privacy:

$$\vdash \mathsf{ExpM} \sim_{\langle \epsilon,0\rangle} \mathsf{ExpM} : \Phi \Longrightarrow max\langle 1\rangle = max\langle 2\rangle$$

where Φ denotes the precondition

$$\begin{array}{ll} & \mathsf{adj}(d\langle 1\rangle, d\langle 2\rangle) \\ \wedge & \mathsf{qscore}\langle 1\rangle = \mathsf{qscore}\langle 2\rangle \\ \wedge & \forall r \in \mathcal{R}.\ |\mathsf{qscore}\langle 1\rangle(d\langle 1\rangle, r) - \mathsf{qscore}\langle 1\rangle(d\langle 2\rangle, r)| \leq 1. \end{array}$$

The conjuncts of the precondition are self-explanatory: the first states that the two databases are adjacent, the second states that the two score functions are equal, and the last states that the quality score function is 1-sensitive.

By the rule [FORALL-EQ], it suffices to prove

$$\vdash \mathsf{ExpM} \sim_{\langle \epsilon,0\rangle} \mathsf{ExpM} : \Phi \Longrightarrow (max\langle 1\rangle\ = i) \Rightarrow (max\langle 2\rangle\ = i).$$

for every $i \in \mathbb{Z}$. The main step is to apply the [WHILEEXT] rule with a suitably chosen loop invariant Θ. We set Θ to be

$$(r\langle 1\rangle < i \Rightarrow \Theta_<) \wedge (r\langle 1\rangle \geq i \Rightarrow \Theta_\geq) \wedge r\langle 1\rangle = r\langle 2\rangle,$$

where $\Theta_<$ stands for

$$max\langle 1\rangle < i \ \wedge\ max\langle 2\rangle < i \wedge |bq\langle 1\rangle - bq\langle 2\rangle| \leq 1$$

and $\Theta_\geq$ stands for

$$(max\langle 1\rangle = max\langle 2\rangle = i \wedge bq\langle 1\rangle + 1 = bq\langle 2\rangle) \vee max\langle 1\rangle \neq i.$$

Omitting the assertions required for proving termination and synchronization of the loop iterations (which follows from the conjunct $r\langle 1\rangle = r\langle 2\rangle$), we have to prove three different judgments:

- case $r < i$: $\vdash c \sim_{\langle 0,0\rangle} c : r\langle 1\rangle < i \wedge \Theta_< \Longrightarrow \Theta_<$
- case $r = i$: $\vdash c \sim_{\langle \epsilon,0\rangle} c : r\langle 1\rangle = i \wedge \Theta_< \Longrightarrow \Theta_\geq$
- case $r > i$: $\vdash c \sim_{\langle 0,0\rangle} c : r\langle 1\rangle > i \wedge \Theta_\geq \Longrightarrow \Theta_\geq$

where c denotes the loop body of ExpM:

$cq \xleftarrow{\$} \mathcal{L}^{\mathsf{os}}_{\epsilon/2}(\mathsf{qscore}(d, r));$
$\mathsf{if}\ (cq > bq \vee r = 1)\ \mathsf{then}\ max \leftarrow r; bq \leftarrow cq;$
$r \leftarrow r + 1$

Corresponding conditional statements may take the different branches, so we apply one sided-rules [COND-L] and [COND-R].

Report-noisy-max A closely-related mechanism is *Report-noisy-max* (see, e.g., Dwork and Roth [12]). This algorithm has the exact same code except that it samples from the standard (two-sided) Laplace distribution rather than the one-sided Laplace distribution. It is straightforward to prove privacy for this modification with the axiom [LAPGEN] (resp. [LAPNULL]) for the standard Laplace distribution in place of [ONELAPGEN] (resp. [ONELAPNULL]).

6. Above Threshold algorithm

The *Sparse Vector* algorithm is the canonical example of a program whose privacy proof goes beyond proofs of privacy primitives and composition theorem. The core of the algorithm is the Above Threshold algorithm. In this section, we prove that the latter (as modeled by the program AboveT) is $(\epsilon, 0)$-differentially private; privacy for the full mechanism follows by sequential composition.

Informal proof By Proposition 6, it suffices to show that for every integer i, the output of AboveT on two adjacent databases yields two sub-distributions over Mem that are related by the $(\epsilon, 0)$-lifting of the interpretation of the assertion

$$r\langle 1\rangle = i \Rightarrow r\langle 2\rangle = i.$$

The coupling proof goes as follows. We start by coupling the samplings of the noisy thresholds so that $T\langle 1\rangle + 1 = T\langle 2\rangle$; the cost of this coupling is $(\epsilon/2, 0)$. For the first $i - 1$ queries, we couple the samplings of the noisy query outputs using the rule [LAPNULL]. By 1-sensitivity of the queries and adjacency of the two databases, we know $\mathsf{evalQ}(Q[j], d)\langle 2\rangle - \mathsf{evalQ}(Q[j], d)\langle 1\rangle \leq 1$, so

$$S\langle 1\rangle < T\langle 1\rangle \Rightarrow S\langle 2\rangle < T\langle 2\rangle.$$

Thus, if side $\langle 1\rangle$ does not change the value of r, neither does side $\langle 2\rangle$. In fact, we have the stronger invariant

$$r\langle 1\rangle = |Q| + 1 \Rightarrow r\langle 2\rangle = |Q| + 1 \wedge (r\langle 1\rangle = |Q| + 1 \vee r\langle 1\rangle < i),$$

where $r = |Q| + 1$ means that the loop has not exceeded the threshold yet.

When we reach the ith iteration and $i < |Q| + 1$, we couple the samplings of S so that $S\langle 1\rangle + 1 = S\langle 2\rangle$; the cost of this coupling is $(\epsilon/2, 0)$. Because $T\langle 1\rangle + 1 = T\langle 2\rangle$ and $S\langle 1\rangle + 1 = S\langle 2\rangle$, we enter the conditional in the second execution as soon as we enter the conditional in the first execution. For the remaining iterations $r > i$, it is easy to prove

$$r\langle 1\rangle = i \Rightarrow r\langle 2\rangle = i.$$

Formal proof We prove the following apRHL$^+$ judgment, which entails $(\epsilon, 0)$-differential privacy:

$$\vdash \mathsf{AboveT} \sim_{\langle \epsilon,0\rangle} \mathsf{AboveT} : \Phi \Longrightarrow r\langle 1\rangle = r\langle 2\rangle,$$

where Φ denotes the precondition

$$\begin{array}{ll} & \mathsf{adj}(d\langle 1\rangle, d\langle 2\rangle) \\ \wedge & t\langle 1\rangle = t\langle 2\rangle \\ \wedge & Q\langle 1\rangle = Q\langle 2\rangle \\ \wedge & \forall j.\ |\mathsf{evalQ}(Q\langle 1\rangle[j], d\langle 1\rangle) - \mathsf{evalQ}(Q\langle 2\rangle[j], d\langle 2\rangle)| \leq 1. \end{array}$$

The conjuncts of the precondition are straightforward: the first states that the two databases are adjacent, the second and third state that Q and t coincide in both runs, and the last states that all queries are 1-sensitive. By the rule [FORALL-EQ], it suffices to prove

$$\vdash \mathsf{AboveT} \sim_{\langle \epsilon,0\rangle} \mathsf{AboveT} : \Phi \Longrightarrow (r\langle 1\rangle = i) \Rightarrow (r\langle 2\rangle = i).$$

for every $i \in \mathbb{Z}$.

We begin with the three initializations:

$$\begin{array}{l} j \leftarrow 1; \\ r \leftarrow |Q| + 1; \\ T \xleftarrow{\$} \mathcal{L}_\epsilon(t); \end{array}$$

This command c_0 computes a noisy version of the threshold t. We use the rule [LAPGEN] with $\epsilon = \epsilon/2$, $k = 1$ and $k' = k$, noticing that t is the same value in both sides. This proves the judgment

$$\vdash c_0 \sim_{\epsilon/2} c_0 : \Phi \Longrightarrow T\langle 1\rangle + 1 = T\langle 2\rangle.$$

Notice that the $\epsilon/2$ we are paying here is *not* for the privacy of the threshold—which is not private information!—but rather for ensuring that the noisy thresholds are *one apart* in the two runs.

Next, we consider the main loop c_1:

$$\begin{array}{l} \mathsf{while}\ j < |Q|\ \mathsf{do} \\ \quad S \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(\mathsf{evalQ}(Q[j], d)); \\ \quad \mathsf{if}\ (T \leq S\ \wedge r = |Q| + 1)\ \mathsf{then}\ r \leftarrow j; \\ \quad j \leftarrow j + 1; \end{array}$$

and prove the judgment

$$\vdash c_1 \sim_{\epsilon/2} c_1 : \Phi \wedge T\langle 1\rangle + 1 = T\langle 2\rangle \Longrightarrow (r\langle 1\rangle = i) \Rightarrow (r\langle 2\rangle = i)$$

with the [WHILEEXT] rule. The proof is similar to the one for the Exponential mechanism, using invariants from the informal proof.

Other versions of Above Threshold As noted in the introduction, different versions of Above Threshold have been considered in the literature. One variant returns the first noisy value above threshold; see Figure 6 for the code. While this was thought to be private, errors in the proof were later uncovered. Under our coupling proof, the error is obvious: we need to prove $v\langle 1\rangle = v\langle 2\rangle$ for the result to be private, so we need $\mathsf{evalQ}(Q[i], d\langle 1\rangle) = \mathsf{evalQ}(Q[i], d\langle 2\rangle)$ after the critical iteration $r = i$. But we have already coupled $\mathsf{evalQ}(Q[i], d\langle 1\rangle) + 1 = \mathsf{evalQ}(Q[i], d\langle 2\rangle)$ during this iteration. Lyu et al. [25] provide further discussion of this, and other, incorrect implementations of the Sparse Vector technique.

On the other hand, it is possible to prove $(2\epsilon, 0)$-differential privacy for a modified version of the algorithm, where the returned value uses fresh noise (e.g. by adding after the loop has completed the sampling $v \xleftarrow{\$} \mathcal{L}_\epsilon(\mathsf{evalQ}(Q[r], d))$).

$$\begin{array}{l} i \leftarrow 1; v \leftarrow 0; r \leftarrow |Q| + 1; \\ T \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(t); \\ \mathsf{while}\ i < |Q|\ \mathsf{do} \\ \quad S \xleftarrow{\$} \mathcal{L}_{\epsilon/4}(\mathsf{evalQ}(Q[i], d)); \\ \quad \mathsf{if}\ (T \leq S\ \wedge r = |Q| + 1)\ \mathsf{then}\ r \leftarrow i;\ v \leftarrow S \\ \quad i \leftarrow i + 1; \\ \mathsf{return}\ v \end{array}$$

Figure 6. Buggy Above Threshold algorithm

Another interesting variant of the algorithm deals with streams of queries. If the output of the queries is uniformly bounded below, then the program terminates with probability 1 and the proof proceeds as usual. However, if the answers to the stream of queries are below the threshold and falling, the probability of non-termination can be positive. The interaction of non-termination and differential privacy is unusual; most works assume that algorithms always terminate.

The Sparse Vector technique has also been studied by the database community. Recent work by Chen and Machanavajjhala [10] shows that many proposed generalizations of the Sparse Vector algorithm are not differentially private.

7. Related work

Coupling is an established tool in probability theory, but it seems less familiar to computer science. It was only quite recently that couplings have been used in cryptography; according to Hoang and Rogaway [20], who use couplings to reason about generalized Feistel networks, Mironov [28] first used this technique in his analysis of RC4. Similarly, we are not aware of couplings in differential privacy, though there seems to be an implicit coupling argument by Dwork et al. [16]. There are seemingly few applications of coupling in formal verification, despite considerable research on probabilistic bisimulation (first introduced by Larsen and Skou [23]) and probabilistic relational program logics (first introduced by Barthe et al. [3]). The connection between liftings and couplings was recently noted by Barthe et al. [6].

There are many language-based techniques for proving differential privacy for programs, including dynamic checking [17, 26], the already mentioned relational program logic [2, 4] and relational refinement type systems [8], linear (dependent) type systems [18, 29], product programs [5], and methods based on computing bisimulations families for probabilistic automata [31, 32]. None of these techniques can deal with the examples in this paper.

8. Conclusion

We show new methods for proving differential privacy with approximate couplings. We take advantage of the full generality of approximate couplings, showing that the composition principle for couplings generalizes the standard composition principle for differential privacy. Our principles support concise and compositional proofs that are arguably more elegant than existing pen-and-paper proofs. Although our results are presented from the perspective of formal verification, we believe that our contributions are also relevant to the differential privacy communities.

In the future, we plan to use our methods also for the verification adaptive data analysis algorithms used to prevent false discoveries, such as the one proposed by Dwork et al. [15], and for the formal verification of mechanism design [7]. Beyond these examples, the pointwise characterization of equality can be adapted to stochastic dominance, and provides a useful tool to further investigate machine-checked verification of coupling arguments.

It could also be interesting to use the pointwise characterization of differential privacy to simplify existing formal proofs. For exam-

ple, Barthe et al. [4] prove differential privacy of the vertex cover algorithm [19]. This algorithm does not use standard primitives; instead, it samples from a custom distribution specific to the graph. The existing formal proof uses a custom rule for loops, reasoning by case analysis on the output of the random samplings. Pointwise differential privacy could handle this reasoning more elegantly.

Acknowledgments We warmly thank Aaron Roth for challenging us with the problem of verifying Sparse Vector. We also thank him and Jonathan Ullman for good discussions about challenges and subtleties of the proof of Sparse Vector. This work was partially supported by NSF grants TWC-1513694, CNS-1065060 and CNS-1237235, by EPSRC grant EP/M022358/1 and by a grant from the Simons Foundation (#360368 to Justin Hsu).

References

[1] M. Barr. Relational algebras. In S. Mac Lane, editor, *Reports of the Midwest Category Seminar, IV*, volume 137 of *Lecture Notes in Mathematics*, page 39–55. Springer-Verlag, 1970. URL http://www.math.mcgill.ca/barr/papers/relalgs.pdf.

[2] G. Barthe and F. Olmedo. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In *International Colloquium on Automata, Languages and Programming (ICALP), Riga, Latvia*, volume 7966 of *Lecture Notes in Computer Science*, pages 49–60. Springer, 2013. URL http://certicrypt.gforge.inria.fr/2013.ICALP.pdf.

[3] G. Barthe, B. Grégoire, and S. Zanella-Béguelin. Formal certification of code-based cryptographic proofs. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Savannah, Georgia*, pages 90–101, New York, 2009. URL http://research.microsoft.com/pubs/185309/Zanella.2009.POPL.pdf.

[4] G. Barthe, B. Köpf, F. Olmedo, and S. Zanella-Béguelin. Probabilistic relational reasoning for differential privacy. *ACM Transactions on Programming Languages and Systems*, 35(3):9, 2013. URL http://software.imdea.org/~bkoepf/papers/toplas13.pdf.

[5] G. Barthe, M. Gaboardi, E. J. Gallego Arias, J. Hsu, C. Kunz, and P.-Y. Strub. Proving differential privacy in Hoare logic. In *IEEE Computer Security Foundations Symposium (CSF), Vienna, Austria*, 2014. URL http://arxiv.org/abs/1407.2988.

[6] G. Barthe, T. Espitau, B. Grégoire, J. Hsu, L. Stefanesco, and P.-Y. Strub. Relational reasoning via probabilistic coupling. In *International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR), Suva, Fiji*, volume 9450, pages 387–401, 2015. URL http://arxiv.org/abs/1509.03476.

[7] G. Barthe, M. Gaboardi, E. J. G. Arias, J. Hsu, A. Roth, and P. Strub. Computer-aided verification in mechanism design. *CoRR*, abs/1502.04052, 2015. URL http://arxiv.org/abs/1502.04052.

[8] G. Barthe, M. Gaboardi, E. J. Gallego Arias, J. Hsu, A. Roth, and P.-Y. Strub. Higher-order approximate relational refinement types for mechanism design and differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Mumbai, India*, 2015. URL http://arxiv.org/abs/1407.6845.

[9] N. Benton. Simple relational correctness proofs for static analyses and program transformations. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Venice, Italy*, pages 14–25, 2004. URL http://research.microsoft.com/en-us/um/people/nick/correctnessfull.pdf.

[10] Y. Chen and A. Machanavajjhala. On the privacy properties of variants on the sparse vector technique. *CoRR*, abs/1508.07306, 2015. URL http://arxiv.org/abs/1508.07306.

[11] Y. Deng and W. Du. Logical, metric, and algorithmic characterisations of probabilistic bisimulation. Technical Report CMU-CS-11-110, Carnegie Mellon University, March 2011. URL http://arxiv.org/abs/1103.4577.

[12] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. URL http://dx.doi.org/10.1561/0400000042.

[13] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *IACR International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), Saint Petersburg, Russia*, pages 486–503. Springer, 2006.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *IACR Theory of Cryptography Conference (TCC), New York, New York*, pages 265–284, 2006. URL http://dx.doi.org/10.1007/11681878_14.

[15] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Preserving statistical validity in adaptive data analysis. In *ACM SIGACT Symposium on Theory of Computing (STOC), Portland, Oregon*, pages 117–126, 2015. URL http://doi.acm.org/10.1145/2746539.2746580.

[16] C. Dwork, M. Naor, O. Reingold, and G. N. Rothblum. Pure differential privacy for rectangle queries via private partitions. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), Auckland, New Zealand*, pages 735–751. Springer Berlin Heidelberg, 2015. ISBN 978-3-662-48799-0. URL http://dx.doi.org/10.1007/978-3-662-48800-3_30.

[17] H. Ebadi, D. Sands, and G. Schneider. Differential privacy: Now it's getting personal. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Mumbai, India*, pages 69–81, 2015. ISBN 978-1-4503-3300-9. URL http://dl.acm.org/citation.cfm?id=2676726.

[18] M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan, and B. C. Pierce. Linear dependent types for differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Rome, Italy*, pages 357–370, 2013. URL http://dl.acm.org/citation.cfm?id=2429113.

[19] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *ACM–SIAM Symposium on Discrete Algorithms (SODA), Austin, Texas*, pages 1106–1125, 2010. URL http://arxiv.org/pdf/0903.4510v2.

[20] V. T. Hoang and P. Rogaway. On generalized Feistel networks. In *IACR International Cryptology Conference (CRYPTO), Santa Barbara, California*, volume 6223 of *Lecture Notes in Computer Science*, pages 613–630. Springer, 2010. URL https://eprint.iacr.org/2010/301.pdf.

[21] B. Jonsson, W. Yi, and K. G. Larsen. Probabilistic extensions of process algebras. In *Handbook of Process Algebra*, pages 685–710. Elsevier, Amsterdam, 2001. URL citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.7376.

[22] D. Kozen. Semantics of probabilistic programs. In *IEEE Symposium on Foundations of Computer Science (FOCS), San Juan, Puerto Rico*, pages 101–114, 1979.

[23] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *ACM Symposium on Principles of Programming Languages (POPL), Austin, Texas*, pages 344–352. ACM Press, 1989. URL http://doi.acm.org/10.1145/75277.75307.

[24] T. Lindvall. *Lectures on the coupling method*. Courier Corporation, 2002.

[25] M. Lyu, D. Su, and N. Li. Understanding the sparse vector technique for differential privacy. 2016. URL http://arxiv.org/abs/1603.01699.

[26] F. McSherry. Privacy integrated queries. In *ACM SIGMOD International Conference on Management of Data (SIGMOD), Providence, Rhode Island*, 2009. URL http://research.microsoft.com/pubs/80218/sigmod115-mcsherry.pdf.

[27] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science (FOCS), Providence, Rhode Island*, pages 94–103, 2007. URL http://doi.ieeecomputersociety.org/10.1109/FOCS.2007.41.

[28] I. Mironov. (Not so) random shuffles of RC4. In *IACR International Cryptology Conference (CRYPTO), Santa Barbara, California*, volume 2442 of *Lecture Notes in Computer Science*, pages 304–319. Springer, 2002. URL https://eprint.iacr.org/2002/067.pdf.

[29] J. Reed and B. C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *ACM SIGPLAN International*

Conference on Functional Programming (ICFP), Baltimore, Maryland, 2010. URL http://dl.acm.org/citation.cfm?id=1863568.

[30] H. Thorisson. *Coupling, Stationarity, and Regeneration*. Springer, 2000.

[31] M. C. Tschantz, D. Kaynar, and A. Datta. Formal verification of differential privacy for interactive systems (extended abstract). *Electronic Notes in Theoretical Computer Science*, 276(0):61–79, 2011. ISSN 1571–0661. URL http://arxiv.org/pdf/1101.2819v1.

[32] L. Xu, K. Chatzikokolakis, and H. Lin. Metrics for differential privacy in concurrent systems. In *IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems (FORTE), Berlin, Germany*, volume 8461 of *Lecture Notes in Computer Science*, pages 199–215, June 2014. URL https://hal.inria.fr/hal-00879140.