Advanced Probabilistic Couplings for Differential Privacy

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

Advanced Probabilistic Couplings for Differential Privacy*

Gilles Barthe
IMDEA Software Institute
Madrid, Spain

Noémie Fong
ENS
Paris, France

Marco Gaboardi†
University at Buffalo, SUNY
Buffalo, USA

Benjamin Grégoire
Inria
Sophia-Antipolis, France

Justin Hsu‡
University of Pennsylvania
Philadelphia, USA

Pierre-Yves Strub
IMDEA Software Institute
Madrid, Spain

ABSTRACT

Differential privacy is a promising formal approach to data privacy, which provides a quantitative bound on the privacy cost of an algorithm that operates on sensitive information. Several tools have been developed for the formal verification of differentially private algorithms, including program logics and type systems. However, these tools do not capture fundamental techniques that have emerged in recent years, and cannot be used for reasoning about cutting-edge differentially private algorithms. Existing techniques fail to handle three broad classes of algorithms: 1) algorithms where privacy depends on accuracy guarantees, 2) algorithms that are analyzed with the advanced composition theorem, which shows slower growth in the privacy cost, 3) algorithms that interactively accept adaptive inputs.

We address these limitations with a new formalism extending apRHL [6], a relational program logic that has been used for proving differential privacy of non-interactive algorithms, and incorporating aHL [11], a (non-relational) program logic for accuracy properties. We illustrate our approach through a single running example, which exemplifies the three classes of algorithms and explores new variants of the Sparse Vector technique, a well-studied algorithm from the privacy literature. We implement our logic in EasyCrypt, and formally verify privacy. We also introduce a novel coupling technique called *optimal subset coupling* that may be of independent interest.

1. INTRODUCTION

Differential privacy, a rigorous and quantitative notion of statistical privacy, is one of the most promising formal definitions of privacy to date. Since its initial formulation by Dwork et al. [19], differential privacy has attracted substantial attention throughout computer science, including areas like databases, machine learning, and optimization, and more.

There are several reasons for this success. For one, differential privacy allows a formal trade-off between privacy and accuracy: differentially private algorithms come with a *privacy guarantee* expressed in terms of two parameters ϵ (expressing the privacy cost) and δ (expressing the probability of violating the privacy cost). For both parameters, smaller values offer stronger privacy guarantees.

Another important advantage differential privacy is that it *composes* well: differentially private algorithms can be easily combined to build new private algorithms. The differential privacy literature offers several *composition theorems*, differing in how the privacy parameter of the larger algorithm depends on the parameters of the components. These composition properties can also be used in *interactive and adaptive* scenarios where an adversary can decide which algorithm to run depending on the outputs of previous algorithms.

Differential privacy's clean composition properties also make it an attractive target for an unusually diverse array of formal verification techniques. By now, there are tools that formally guarantee differential privacy via relational program logics [2, 6], linear type systems [25, 29, 42], interactive automata [50, 51], product programs [7], satisfiability modulo theories [28], refinement type systems [9], and more. While these systems formalize privacy through a wide variety techniques, most of these approaches analyze a composition of private algorithms using the *sequential composition* theorem of differential privacy, which guarantees that the resulting algorithms have parameter ϵ and δ equal to the *sum* of the parameters of their components.

Recently, Barthe et al. [10] highlighted a close connection between *approximate couplings* and differential privacy which enables formal verification beyond sequential composition. Barthe et al. [10] work with the relational program logic apRHL [6], extended with a new composition principle called *pointwise privacy*. The idea is to first prove a restricted case of privacy—corresponding roughly to privacy for a single output—and then combine the results to prove the full differential privacy property. Combined with the composition principle for approximate couplings, which generalizes the sequential composition theorem, apRHL can express simple, compositional proofs of algorithms like the one-side Laplace implementation of the Exponential mechanism [17] and the Above Threshold algorithm [17] while abstracting away reasoning about probabilistic events. Existing privacy proofs for these algorithms, even on paper, involve ad hoc computation of probabilities.

While apRHL substantially expands the range of formal verification in differential privacy, there are still private algorithms that apRHL cannot verify. Roughly, there are three missing pieces:

*The full version of this paper is available at https://arxiv.org/abs/1606.07143.

†Partially supported by NSF grants CNS-1237235 and CNS-1565365, and by EPSRC grant EP/M022358/1.

‡Partially supported by NSF grants TC-1065060 and TWC-1513694, and a grant from the Simons Foundation (#360368 to Justin Hsu).

CCS'16, October 24 - 28, 2016, Vienna, Austria

ISBN 978-1-4503-4139-4/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2976749.2978391

- *Accuracy-dependent privacy*. Some algorithms are only private if an accuracy property holds.
- *Advanced composition*. This principle shows slower growth in the privacy cost, in exchange for a small probability of violating privacy. The proof involves an technical martingale concentration argument.
- *Interactive privacy*. Some private algorithms are *interactive*, receiving a continuous sequence of *adaptive* inputs while producing intermediate outputs.

These three missing pieces correspond to three fundamental principles of differential privacy. While there are many algorithms from the privacy literature that use one (or more) of these three features, to structure our presentation we will work with a variant of the Sparse Vector technique based on the Between Thresholds algorithm [13], a single unifying example that uses all three features (§ 2). After reviewing some technical preliminaries about differential privacy, approximate couplings, and the logic apRHL (§ 3), we describe extensions to apRHL to verify privacy for new classes of algorithms.

- New proof rules that allow reasoning within apRHL while assuming an accuracy property, incorporating accuracy proofs from the Hoare logic aHL [11] (§ 4). We demonstrate these rules on a classic example of accuracy-dependent privacy: the Propose-test-release framework [16, 48].
- A proof rule that analyzes loops using the advanced composition principle; soundness relies on a novel generalization of advanced composition to approximate couplings (§ 5).
- New proof rules for *adversaries*, external procedure calls that model an adaptive source of inputs (§ 6).
- Orthogonal to reasoning about accuracy, advanced composition, and adversarial inputs, we introduce a general construction that may be of independent interest called the *optimal subset coupling*. This construction gives an approximate lifting relating subsets that yields the best possible ϵ, and we use this construction to give a new interval coupling rule for the Laplace distribution (§ 7).

We then show how to combine these ingredients to verify our main running example, the Between Thresholds algorithm (§ 8). We finish with related work (§ 9) and some concluding thoughts (§ 10).

2. MOTIVATING EXAMPLE

Before diving into the technical details we'll first present our motivating example, which involves accuracy-dependent privacy, advanced composition, and interactive privacy. We first review the definition of differential privacy, a relational property about probabilistic programs proposed by Dwork, McSherry, Nissim and Smith.

Definition 1. *Let the* adjacency relation *be* $\Phi \subseteq A \times A$*, and* $\epsilon, \delta > 0$*. A program* $M : A \to \mathbf{Distr}(B)$ *satisfies* (ϵ, δ)-differential privacy *with respect to* Φ *if for every pair of inputs* $a, a' \in A$ *such that* $\Phi(a, a')$ *and every subset of outputs* $S \subseteq B$*, we have*

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

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

Intuitively, Φ relates inputs that differ in a single individual's data. Then, differential privacy requires that the two resulting distributions on outputs should be close.

$\mathsf{ASV}_{\mathrm{bt}}(a, b, M, N, d) :=$
$i \leftarrow 0; l \leftarrow [];$
$u \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(0);$
$A \leftarrow a - u; B \leftarrow b + u;$
while $i < N \wedge |l| < M$ do
  $q \leftarrow \mathcal{A}(l);$
  $S \xleftarrow{\$} \mathcal{L}_{\epsilon'/3}(\mathsf{evalQ}(q, d));$
  if $(A \leq S \leq B)$ then $l \leftarrow i :: l;$
  $i \leftarrow i + 1;$
return l

Figure 1: Sparse Vector for Between Thresholds

Our motivating example is *Adaptive Sparse Vector for Between Thresholds* ($\mathsf{ASV}_{\mathrm{bt}}$), a variant of the Sparse Vector algorithm. Our algorithm takes a stream of numeric queries as input, and answers only the queries that take a value within some range. The main benefit of Sparse Vector is that queries that take a value outside the range do not increase the privacy cost, even though testing whether whether the query is (approximately) in the range involves private data. Sparse Vector is an appealing example, because of its popularity and its difficulty. In particular, the privacy proof of Above Threshold is non-compositional and notoriously tricky, and several variants[1] of the algorithm were supposedly proved to be private but were later shown to be non-private (Lyu et al. [34] provide a comprehensive survey).

The code of $\mathsf{ASV}_{\mathrm{bt}}$ is shown in Fig. 1. At a high level, the algorithm accepts a stream of adversarially chosen queries and produces a list of queries whose answer lies (approximately) between two threshold parameters a and b. The algorithm computes noisy versions A and B of a and b using the Laplace mechanism $\mathcal{L}_\epsilon$, which we review in the next section, and then performs an interactive loop for a fixed number (N) of iterations. Each step, a stateful adversary $\mathcal{A}$ receives the current list l of queries whose answer on input database d lies between $[A, B]$ and selects a new query q. If its noisy answer S lies in $[A, B]$ and there have been fewer than M queries between threshold, the algorithm adds q to the list l. Our algorithm differs from standard presentations of Adaptive Sparse Vector [17] in two significant respects:

- we use BetweenThresholds rather than AboveThreshold for deciding whether to add a query to the list;
- we do not rerandomize the noise on the thresholds each time a query is added to l; therefore, our algorithm adds less noise.

$\mathsf{ASV}_{\mathrm{bt}}$ satisfies the following privacy guarantee.

Theorem 2. *Let* ϵ *and* δ *both be in* $(0, 1)$*. Set*

$$\epsilon' \triangleq \frac{\epsilon}{4\sqrt{2M \ln(2/\delta)}},$$

and suppose a *and* b *are such that*

$$b - a \geq \frac{6}{\epsilon'} \ln(4/\epsilon') + \frac{4}{\epsilon} \ln(2/\delta).$$

[1]There exist multiple versions of Sparse Vector. The earliest reference seems to be Dwork et al. [20]; several refinements were proposed by Hardt and Rothblum [30], Roth and Roughgarden [44]. Applications often use their own variants, e.g. Shokri and Shmatikov [46]. The most canonical version of the algorithm is the version by Dwork and Roth [17].

If all adversarial queries q are 1-sensitive (i.e. $|\mathsf{evalQ}(q,d) - \mathsf{evalQ}(q,d')| \leq 1$ for every adjacent databases d and d'), then $\mathsf{ASV}_{\mathrm{bt}}$ is (ϵ,δ)-differentially private.

The formal proof of this theorem, which we have verified in an implementation of our logic within the EasyCrypt system, involves several features:

- reasoning principles for mixing accuracy and privacy guarantees, using a combination of relational logics [6, 10] and non-relational logics [11];
- a generalization of the advanced composition theorem for handling the body of the loop;
- an adversary rule for handling interactive inputs in the loop; and
- a new reasoning principle, called *optimal subset coupling*, for handling the Laplace mechanism in the loop.

We stress that the use of pointwise equality, which is required for proving privacy of between thresholds, makes the proof significantly more challenging than other examples involving solely adaptive adversaries, advanced composition, and accuracy-dependent privacy.

We remark that Bun et al. [13] proposed Between Threshold and proved its privacy. Their proof does not use advanced composition, and follows from a somewhat complicated calculations about the probabilities of certain events. Our proof demonstrates the power of approximate liftings: somewhat surprisingly, we arrive at an elegant privacy proof without probabilistic reasoning.

3. BACKGROUND

Before presenting our new extensions, we first review some preliminaries about differential privacy, the connection to approximate liftings, the program logic apRHL [6] and its extension apRHL$^+$ [10], and the union bound logic aHL [11].

3.1 Mathematical preliminaries

To avoid measure-theoretic issues, we base our technical development on distributions over discrete sets B. A function $\mu : B \to \mathbb{R}^{\geq 0}$ is 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. We write $\mathbf{Distr}(B)$ for the set of discrete distributions over B. Equality of distributions is defined as pointwise equality of functions.

We will also use *marginal distributions*. Formally, the first and second marginals of a distribution $\mu \in \mathbf{Distr}(B_1 \times B_2)$ are simply the projections: the distributions $\pi_1(\mu) \in \mathbf{Distr}(B_1)$ and $\pi_2(\mu) \in \mathbf{Distr}(B_2)$ given by

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

3.2 Differential privacy

We will need several tools from differential privacy; readers should consult the textbook by Dwork and Roth [17] for a more comprehensive introduction. Most differentially private algorithms are constructed from private primitive operations. The most famous primitive is the Laplace mechanism.

Definition 3 (Laplace mechanism [19])**.** *Let $\epsilon > 0$. The* (discrete) Laplace mechanism $\mathcal{L}_\epsilon : \mathbb{Z} \to \mathbf{Distr}(\mathbb{Z})$ *is defined by $\mathcal{L}_\epsilon(t) = t + \nu$, where $\nu \in \mathbb{Z}$ with probability proportional to*

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

The level of privacy depends on the sensitivity of the query, which measures how far the function may differ on two related inputs. Roughly, adding the same level of Laplace noise to a higher sensitivity query will be less private.

Definition 4 (Sensitivity)**.** *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 $\Phi(a_1, a_2)$.*

The Laplace mechanism satisfies an accuracy specification.

Proposition 5 (Laplace accuracy)**.** *Let $\epsilon, \beta > 0$, and suppose x is the result from running $\mathcal{L}_\epsilon(t)$. Then $|x - t| \leq \frac{1}{\epsilon} \ln \frac{1}{\beta}$ with probability at least $1 - \beta$.*

Besides private primitives, the other main tools for constructing private programs are the *composition theorems*. These results describe the privacy level for a combination of private programs—say, calls to the Laplace mechanism. We will use a bit of notation for compositions. Let $\{f_i\}$ be a set of n functions of type $A \to D \to \mathbf{Distr}(A)$. Denote the n-fold composition $f^n : A \to D \to \mathbf{Distr}(A)$ by

$$f^k(a,d) = \begin{cases} \mathsf{unit}\ a & : k = 0 \\ \mathsf{bind}\ f^{k-1}(a,d)\ f_k(-,d) & : k \geq 1. \end{cases}$$

Here, $\mathsf{unit} : A \to \mathbf{Distr}(A)$ and $\mathsf{bind} : \mathbf{Distr}(A) \to (A \to \mathbf{Distr}(B)) \to \mathbf{Distr}(B)$ are the monadic operations for distributions. They satisfy the following equalities:

$$\mathsf{unit}(a)(b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

and for $f : A \to \mathbf{Distr}(B)$ and $F : A \to B \to \mathbf{Distr}(C)$,

$$(\mathsf{bind}\ f\ F)(a)(c) = \sum_b f(a)(b)\ F(a)(b)(c).$$

We will also use this composition notation when the functions $\{f_i\}$ have type $A \to \mathbf{Distr}(A)$, simply dropping the parameter d above when defining $f^n : A \to \mathbf{Distr}(A)$.

Then, the most basic composition theorem in differential privacy is *sequential composition*.

Theorem 6 (Sequential composition)**.** *Let $f_i : A \to D \to \mathbf{Distr}(A)$ be a sequence of n functions, such that for every fixed $a \in A$, the functions $f_i(a) : D \to \mathbf{Distr}(A)$ are (ϵ_i, δ_i)-differentially private for some adjacency relation on D. Then for every initial value $a \in A$, the composition $f^n(a) : D \to \mathbf{Distr}(A)$ is (ϵ^*, δ^*)-differentially private for*

$$\epsilon^* = \sum_{i=1}^n \epsilon_i \quad \textit{and} \quad \delta^* = \sum_{i=1}^n \delta_i.$$

That is, the epsilons and deltas sum up through composition.

The sequential composition theorem is quite useful, and is the main principle supporting modular verification of differential privacy. However, there is another composition theorem, known as *advanced composition* [22]. Instead of summing up the privacy costs, this theorem gives slower growth of ϵ in exchange for increasing the δ parameter. Advanced composition is an extremely common tool for analyzing differentially private algorithms, but it is not supported by most formal verification systems today.

Theorem 7 (Advanced composition)**.** *Let $f_i : A \to D \to \mathbf{Distr}(A)$ be a sequence of n functions, such that for every fixed $a \in A$, the functions $f_i(a) : D \to \mathbf{Distr}(A)$ are (ϵ, δ)-differentially*

private for some adjacency relation on D. Then, for every $a \in A$ and $\omega \in (0,1)$, the composed function $f^n(a) : D \to \mathbf{Distr}(A)$ is (ϵ^*, δ^*)-differentially private for

$$\epsilon^* = \left(\sqrt{2n\ln(1/\omega)}\right)\epsilon + n\epsilon(e^\epsilon - 1) \quad \textit{and} \quad \delta^* = n\delta + \omega.$$

In particular, if we have $\epsilon' \in (0,1)$, $\omega \in (0, 1/2)$, and

$$\epsilon = \frac{\epsilon'}{2\sqrt{2n\ln(1/\omega)}},$$

a short calculation shows that the function f^n is (ϵ', δ^*)-differentially private.

Kairouz et al. [32] propose sharper versions of this composition theorem, including a provably optimal version and a version for the heterogeneous case when the privacy level (ϵ_i, δ_i) may depend on i. We will use Theorem 7 for simplicity, but our techniques enable other composition theorems to be easily plugged in.

3.3 Approximate liftings

While the definition of differential privacy seems to be a straightforward property about probabilities in two distributions, a recent line of work initiated by Barthe et al. [6] and subsequently developed [2, 10] shows that differential privacy is a consequence of an approximate version of probabilistic coupling, called *approximate liftings*. Couplings are a long-standing tool in probability theory for analyzing pairs of distributions, but the relation between differential privacy and approximate couplings is still being explored.

Unlike couplings, where there is a single accepted definition, several incomparable notions of approximate liftings have been proposed. The first definition is by Barthe et al. [6] but has some technical shortcomings; we will use a more recent definition by Barthe and Olmedo [2]. We begin by defining a distance on distributions, closely related to (ϵ, δ)-differential privacy.

Definition 8 (Barthe and Olmedo [2])**.** *Let $\epsilon \geq 0$. The ϵ-DP divergence $\Delta_\epsilon(\mu_1, \mu_2)$ between two distributions $\mu_1 \in \mathbf{Distr}(A)$ and $\mu_2 \in \mathbf{Distr}(A)$ is defined as*

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

For the connection to differential privacy, it is not hard to see that if $M : D \to \mathbf{Distr}(A)$, then M is (ϵ, δ)-differentially private iff for every pair of adjacent inputs d, d', we have $\Delta_\epsilon(M(d), M(d')) \leq \delta$. This distance is also central to the definition of *approximate liftings*.

Definition 9 (Barthe and Olmedo [2])**.** *Two distributions $\mu_1 \in \mathbf{Distr}(A_1)$ and $\mu_2 \in \mathbf{Distr}(A_2)$ are related by the (ϵ, δ)-*lifting *of $\Psi \subseteq A_1 \times A_2$, written $\mu_1 \; \Psi^{\sharp(\epsilon,\delta)} \; \mu_2$, if there exist two* witness *distributions $\mu_L \in \mathbf{Distr}(A_1 \times A_2)$ and $\mu_R \in \mathbf{Distr}(A_1 \times A_2)$ 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. *$\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$.*

Approximate liftings generalize several concepts for relating distributions. When $\mu_L = \mu_R$, we have a $(0,0)$-lifting, sometimes called an exact *probabilistic lifting*. Such a lifting, with any Ψ, implies a *probabilistic coupling* between (μ_1, μ_2).

Approximate liftings satisfy the following property, also known as the fundamental lemma of approximate liftings.

Lemma 10 (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

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

Using this lemma, one can prove that differential privacy is equivalent to a particular form of approximate lifting.

Proposition 11 (Barthe and Olmedo [2])**.** *A probabilistic computation $M : D \to \mathbf{Distr}(A)$ is (ϵ, δ)-differentially private for adjacency relation Φ iff*

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

for every two adjacent inputs a and a'.

Approximate liftings form the basis of the program logic apRHL, to which we turn next.

3.4 The relational program logic

The logic apRHL, originally proposed by Barthe et al. [6], is a relational program logic for verifying differential privacy. We take this logic as our point of departure; we briefly recall the main points here.

We consider a simple imperative language with random sampling, oracle calls and adversary calls; the latter two are new to the present work. The set of commands is defined inductively:

$\mathcal{C} ::=$	skip	noop
	$\mid \mathcal{C}; \mathcal{C}$	sequencing
	$\mid \mathcal{X} \leftarrow \mathcal{E}$	deterministic assignment
	$\mid \mathcal{X} \xleftarrow{\$} \mathcal{L}_\epsilon(\mathcal{E})$	Laplace mechanism
	$\mid \mathsf{if}\ \mathcal{E}\ \mathsf{then}\ \mathcal{C}\ \mathsf{else}\ \mathcal{C}$	conditional
	$\mid \mathsf{while}\ \mathcal{E}\ \mathsf{do}\ \mathcal{C}$	while loop
	$\mid (\mathcal{X}, \ldots, \mathcal{X}) \leftarrow \mathcal{A}(\mathcal{E}, \ldots, \mathcal{E})$	adversary call
	$\mid (\mathcal{X}, \ldots, \mathcal{X}) \leftarrow \mathcal{O}(\mathcal{E}, \ldots, \mathcal{E})$	procedure call

where $\mathcal{X}$ is a set of *variables* and $\mathcal{E}$ is a set of *expressions*. Variables and expressions are typed, and range over standard types like booleans, integers, databases, queries, lists, etc. We omit the semantics of expressions, which is standard. Commands are interpreted as maps $\mathsf{State} \to \mathbf{Distr}(\mathsf{State})$; this is also a standard choice (e.g., see Barthe et al. [6]).[2] We will write $[\![c]\!]_m$ to mean the output distribution of command c, executed on input memory m.

An apRHL judgment has the form

$$\vdash c \sim_{\langle\epsilon,\delta\rangle} c' : \Phi \Longrightarrow \Psi.$$

Reminiscent of Hoare logic, Φ represents the pre-condition while Ψ represents the post-condition. Both Φ and Ψ are first order formulas over the program variables. For expressing relational properties, program variables are tagged with either $\langle 1\rangle$ or $\langle 2\rangle$ to indicate whether they belong to c or c' respectively. For instance, we can assert that the variable x differs by at most 1 in the two runs with the assertion $|x\langle 1\rangle - x\langle 2\rangle| \leq 1$.

Crucially, the post-condition Ψ is interpreted as an approximate lifting over the output distributions. More formally, the judgment is *valid* iff for every two memories m_1 and m_2 such that $m_1 \; \Phi \; m_2$, we have

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

[2] We will assume that commands are terminating on all executions. The logic apRHL can also reason about possibly non-terminating programs by working with sub-distributions instead of distributions.

We present selected rules, taken from prior presentations of apRHL [6, 10] in Fig. 2; $FV(\Phi)$ denotes the set of program variables in the assertion Φ, and $MV(c)$ denotes the set of program variables that are modified (i.e., written) by program c. Many of the rules bear a resemblance to the standard Hoare logic rules. The rules [ASSN] and [COND] are relational versions of the assignment and conditional rules; note that [COND] assumes that the two guards are equal in the pre-condition. The rule [SEQ] reflects the composition principle of approximate liftings, where the indices ϵ and δ add; this rule generalizes the standard composition theorem of differential privacy. The rule [WHILE] extends this reasoning to loops with a bound number of iterations, again assuming that the guards are equal in both programs.

The next two rules, [LAPNULL] and [LAPGEN], are for relating two sampling instructions from the Laplace distribution. Intuitively [LAPNULL] models adding identical noise on both sides, so that the distance between the samples $(y_1\langle 1\rangle, y_2\langle 2\rangle)$ is equal to the distance between the means $(e_1\langle 1\rangle, e_2\langle 2\rangle)$. [LAPGEN] is a general rule for assuming that the two samples are shifted and related by $y_1\langle 1\rangle + k = y_2\langle 2\rangle$; the privacy cost depends on how far the means $(e_1\langle 1\rangle + k, e_2\langle 2\rangle)$ are.

The final group of rules are the structural rules. Besides the usual rules for consequence and framing ([CONSEQ] and [FRAME]), the most interesting rule is the *pointwise equality* rule [PW-EQ]. This rule proves differential privacy by showing a pointwise judgment for each possible output value i, and is the key tool for supporting privacy proofs beyond the standard composition theorems.

3.5 The union bound logic

When reasoning about privacy, we will sometimes need to prove probabilistic bounds on accuracy. Since accuracy properties are not relational, we cannot verify them in apRHL. There is a long history of research for formally verifying probabilistic properties, and we are free to choose any of these techniques to interface with our logic. In our favor, we are interested in simple accuracy properties of the form $\Pr[\Psi] < \beta$, where Ψ is an assertion on the program memory. We call such assertions *bad event assertions*, since they state that the probability of some event Ψ—the "bad event"—is at most β. We will prove accuracy assertions in the Hoare logic aHL [11], which is specialized to prove bad event assertions.

We will highlight just the features of aHL needed for our purposes; readers should consult Barthe et al. [11] for a complete presentation. Concretely, aHL judgments have the following form:

$$\vdash_\beta c : \Phi \Longrightarrow \Psi,$$

where Φ and Ψ are (non-probabilistic) assertions over program memories, and $\beta \in [0, 1]$ is a real-valued index. Assertions in aHL are non-relational, and mention program variables from a *single* memory instead of program variables tagged with $\langle 1\rangle$ or $\langle 2\rangle$. To mediate between the non-relational assertions of aHL and the relational assertions of apRHL, from a non-relational assertion Φ we can define relational assertions $\Phi\langle 1\rangle$ and $\Phi\langle 2\rangle$ by interpreting Φ where all program variables are tagged with $\langle 1\rangle$ or $\langle 2\rangle$ respectively.

The semantics of commands is unchanged from apRHL; we interpret commands as maps $\mathsf{State} \to \mathbf{Distr}(\mathsf{State})$. The above judgement means: for any initial memory satisfying Φ, the probability that $\neg\Psi$ holds in the resulting distribution on memories is at most β. For instance, the accuracy specification of the Laplace mechanism (Proposition 5) is given by the following aHL judgment:

$$\vdash_\beta y \xleftarrow{\$} \mathcal{L}_\epsilon(e) : \top \Longrightarrow |y - e| \leq \frac{1}{\epsilon} \log \frac{1}{\beta}$$

for every $\beta \in (0, 1)$.

4. ACCURACY-DEPENDENT PRIVACY

Let us begin with our first class of private algorithms, where privacy follows from an *accuracy* property. For our purposes, these accuracy properties are non-relational probabilistic properties that hold on a single execution of a single program. For instance, the assertion $\Pr[x > 0] < 0.2$, stating that the probability x is positive is at most 0.2, is an accuracy property. Accuracy properties appear in privacy proofs in a variety of ways. For instance, they may imply that the privacy cost ϵ is smaller than expected. Or, privacy may be conditional: if the accuracy property holds then we have differential privacy, otherwise the algorithm fails and there is no guarantee. Programs in the latter case satisfy (ϵ, δ)-differential privacy, where the probability of failure is included in δ.

4.1 Up-to-bad reasoning

To integrate accuracy assertions into apRHL, we will use a technique from cryptographic verification: *up-to-bad* reasoning. Roughly speaking, rather than directly proving the equality lifting corresponding to differential privacy:

$$\mu_1 \ (=)^{\sharp(\epsilon,\delta)} \ \mu_2,$$

we will prove a conditional, *up-to-bad* lifting:

$$\mu_1 \ \{(x_1, x_2) \mid (\neg\Phi(x_1, x_2) \to x_1 = x_2)\}^{\sharp(\epsilon,\delta)} \ \mu_2.$$

Here, Φ is an assertion involving just variables from one side. Roughly speaking, the lifting shows that if the *bad event* Φ does not hold, then we have differential privacy. Then, we conclude the proof with a structural rule that combines the bad event assertion—proved externally in aHL—with the up-to-bad lifting, removing the bad event while adjusting the privacy parameters (ϵ, δ).

To support this reasoning in our program logic, we propose the two rules in Fig. 3. The rules, [UTB-L] and [UTB-R], internalize an approximate version of up-to-bad reasoning. If the assertion Θ holds, then we have the (ϵ, δ)-lifting of equality. So, if we know the probability of $\neg\Theta$ is at most δ', then we can show the $(\epsilon, \delta + \delta')$-differential privacy when Θ is a property of the first run, and $(\epsilon, \delta + e^\epsilon\delta')$-differential privacy when Θ is a property of the second run. The asymmetry in the left and right versions of the rule reflects the asymmetric definition of approximate lifting, which is in turn inspired by the asymmetric definition of differential privacy.

In order to include these rules, we show that they are valid. To prove the equality lifting for privacy, we would like to use the equivalence in Proposition 11. However, there is a catch: we only know that the distributions over e are differentially private—the distributions over the whole memory may not be differentially private. Therefore, we will use a new property of approximate liftings: they are well-behaved when mapping the underlying distribution.

Proposition 12. *For a function $f : A \to B$, let $f^\sharp$: $\mathbf{Distr}(A) \to \mathbf{Distr}(B)$ denote function lifted to a map on distributions. If f is surjective, and R is a relation on B, then*

$$\mu_1 \ \{(x_1, x_2) \mid f(x_1) \ R \ f(x_2)\}^{\sharp(\epsilon,\delta)} \ \mu_2$$

if and only if

$$f^\sharp \ (\mu_1) \ \{(y_1, y_2) \mid y_1 \ R \ y_2\}^{\sharp(\epsilon,\delta)} f^\sharp \ (\mu_2).$$

In particular, if we have a set E of equivalence classes of A and the distribution $\mu/E : \mathbf{Distr}(E)$ represents the probability of being in each equivalence class, taking $f : A \to E$ mapping an element to its equivalence class and R to be the equivalence relation gives a result by Barthe and Olmedo [2, Proposition 8]:

$$\mu_1 \ (=_E)^{\sharp(\epsilon,\delta)} \ \mu_2 \iff \mu_1/E \ (=)^{\sharp(\epsilon,\delta)} \ \mu_2/E.$$

$$\textsc{Assn}\ \frac{}{\vdash x_1 \leftarrow e_1 \sim_{\langle 0,0\rangle} x_2 \leftarrow e_2 : \Psi\{e_1\langle 1\rangle, e_2\langle 2\rangle / x_1\langle 1\rangle, x_2\langle 2\rangle\} \Longrightarrow \Psi}$$

$$\textsc{Cond}\ \frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \wedge b_1\langle 1\rangle \Longrightarrow \Psi \qquad \vdash c_1' \sim_{\langle \epsilon,\delta\rangle} c_2' : \Phi \wedge \neg b_1\langle 1\rangle \Longrightarrow \Psi}{\vdash \mathsf{if}\ b_1\ \mathsf{then}\ c_1\ \mathsf{else}\ c_1' \sim_{\langle \epsilon,\delta\rangle} \mathsf{if}\ b_2\ \mathsf{then}\ c_2\ \mathsf{else}\ c_2' : \Phi \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \Longrightarrow \Psi}$$

$$\textsc{Seq}\ \frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi' \qquad \vdash c_1' \sim_{\langle \epsilon',\delta'\rangle} c_2' : \Psi' \Longrightarrow \Psi}{\vdash c_1; c_1' \sim_{\langle \epsilon+\epsilon',\delta+\delta'\rangle} c_2; c_2' : \Phi \Longrightarrow \Psi}$$

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

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

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

$$\textsc{Conseq}\ \frac{\vdash \Phi' \sim_{\langle \epsilon',\delta'\rangle} c_1 : c_2 \Longrightarrow \Psi' \qquad \models \Phi \rightarrow \Phi' \qquad \models \Psi' \rightarrow \Psi \qquad \epsilon' \leq \epsilon \qquad \delta' \leq \delta}{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi}$$

$$\textsc{Frame}\ \frac{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \Longrightarrow \Psi \qquad FV(\Theta) \cap MV(c_1, c_2) = \emptyset}{\vdash c_1 \sim_{\langle \epsilon,\delta\rangle} c_2 : \Phi \wedge \Theta \Longrightarrow \Psi \wedge \Theta}$$

$$\textsc{PW-Eq}\ \frac{\forall i.\ \vdash c_1 \sim_{\langle \epsilon,\delta_i\rangle} c_2 : \Phi \Longrightarrow x\langle 1\rangle = i \rightarrow x\langle 2\rangle = i}{\vdash c_1 \sim_{\langle \epsilon, \sum_{i\in I} \delta_i\rangle} c_2 : \Phi \Longrightarrow x\langle 1\rangle = x\langle 2\rangle}$$

Figure 2: Selected proof rules of apRHL [6, 10]

$$\textsc{UtB-L}\ \frac{\models \Phi \rightarrow \Phi_0\langle 1\rangle \qquad \vdash c \sim_{\langle \epsilon,\delta\rangle} c' : \Phi \Longrightarrow \Theta\langle 1\rangle \rightarrow e\langle 1\rangle = e\langle 2\rangle \qquad \vdash_{\delta'} c : \Phi_0 \Longrightarrow \Theta}{\vdash c \sim_{\langle \epsilon,\delta+\delta'\rangle} c' : \Phi \Longrightarrow e\langle 1\rangle = e\langle 2\rangle}$$

$$\textsc{UtB-R}\ \frac{\models \Phi \rightarrow \Phi_0\langle 2\rangle \qquad \vdash c \sim_{\langle \epsilon,\delta\rangle} c' : \Phi \Longrightarrow \Theta\langle 2\rangle \rightarrow e\langle 1\rangle = e\langle 2\rangle \qquad \vdash_{\delta'} c' : \Phi_0 \Longrightarrow \Theta}{\vdash c \sim_{\langle \epsilon,\delta+e^\epsilon\delta'\rangle} c' : \Phi \Longrightarrow e\langle 1\rangle = e\langle 2\rangle}$$

Figure 3: Up-to-bad rules

This result allows us to prove an approximate lifting for a distribution over memories by proving an approximting lifting for the distribution over a single variable or expression. We defer the details of the proof to the full version. Now, we are ready to show soundness of the up-to-bad rules.

Theorem 13. *The rules* [UTB-L] *and* [UTB-R] *are sound.*

Proof. We will start with [UTB-L]. Take any two memories (m_1, m_2) such that $(m_1, m_2) \models \Phi$, and let μ_1, μ_2 be $[\![c]\!]_{m_1}$ and $[\![c']\!]_{m_2}$ respectively. Note that $m_1 \models \Phi_0$. By validity of the premise, we know

$$\Pr_{m\sim\mu_1}[\neg\Theta] \leq \delta'$$

and we have a pair of witnesses μ_L, μ_R for the relation

$$R \triangleq \Theta\langle 1\rangle \rightarrow e\langle 1\rangle = e\langle 2\rangle,$$

such that $\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$. Our goal is to show that the marginal distributions of $[\![e]\!]$ in μ_1, μ_2 satisfy $(\epsilon, \delta + \delta')$-differential privacy, i.e. for any set S,

$$\Pr_{m\sim\mu_1}[[\![e]\!]_m \in S] \leq e^\epsilon \Pr_{m'\sim\mu_2}[[\![e]\!]_{m'} \in S] + \delta + \delta'.$$

To begin, we know that

$$\begin{aligned}\Pr_{m\sim\mu_1}[[\![e]\!]_m \in S] &= \Pr_{m\sim\mu_1}[[\![e]\!]_m \in S \wedge m \models \Theta] \\ &\quad + \Pr_{m\sim\mu_1}[[\![e]\!]_m \in S \wedge m \models \neg\Theta] \\ &\leq \Pr_{m\sim\mu_1}[[\![e]\!]_m \in S \wedge m \models \Theta] + \delta'\end{aligned}$$

since the probability of $\neg\Theta$ in μ_1 is at most δ'. Now, we can conclude with the coupling:

$$\begin{aligned}&\Pr_{m\sim\mu_1}[[\![e]\!]_m \in S \wedge m \models \Theta] + \delta' \\ &= \Pr_{(m,m')\sim\mu_L}[[\![e]\!]_m \in S \wedge m \models \Theta] + \delta' \\ &\leq e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]_m \in S \wedge m \models \Theta] + \delta + \delta' \\ &\leq e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]'_m \in S] + \delta + \delta' \\ &= e^\epsilon \Pr_{m'\sim\mu_2}[[\![e]\!]'_m \in S] + \delta + \delta',\end{aligned}$$

where the first inequality uses $\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$, while the second inequality uses $(m, m') \in \mathsf{supp}(\mu_R)$. So, the distributions of $[\![e]\!]$ satisfy differential privacy. By Proposition 11 and Proposition 12

with equivalence classes defined by the value of $[\![e]\!]$, we can conclude soundness of [UTB-L].

We can show soundness of [UTB-R] in a similar way. Let μ_1, μ_2 be as above. We can use the coupling as follows:

$$\begin{aligned}
&\Pr_{m\sim\mu_1}[[\![e]\!]_m \in S] \\
&= \Pr_{(m,m')\sim\mu_L}[[\![e]\!]_m \in S] \\
&\leq e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]_m \in S] + \delta \\
&= e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]_m \in S \wedge m' \models \Theta] \\
&+ e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]_m \in S \wedge m' \models \neg\Theta] + \delta \\
&\leq e^\epsilon \Pr_{(m,m')\sim\mu_R}[[\![e]\!]'_m \in S] + e^\epsilon \Pr_{(m,m')\sim\mu_R}[m' \models \neg\Theta] + \delta \\
&= e^\epsilon \Pr_{m'\sim\mu_2}[[\![e]\!]_{m'} \in S] + e^\epsilon \Pr_{m'\sim\mu_2}[m' \models \neg\Theta] + \delta \\
&\leq e^\epsilon \Pr_{m'\sim\mu_2}[[\![e]\!]_{m'} \in S] + \delta + e^\epsilon\delta'
\end{aligned}$$

The first inequality uses the bound on the distance between the witnesses: $\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$. The second inequality uses the support of μ_R. The final inequality uses the fact that $\Pr_{m'\sim\mu_2}[m' \models \neg\Theta] \leq \delta'$. Since the distributions of $[\![e]\!]$ in μ_1, μ_2 satisfy $(\epsilon, \delta + e^\epsilon\delta')$-differential privacy, we can again use Proposition 11 and Proposition 12 to show [UTB-R] is sound. □

4.2 Propose-Test-Release

To give a small example of up-to-bad reasoning, we can prove privacy for the *Propose-Test-Release (PTR)* framework [16, 48], a classic example of privacy depending on an accuracy guarantee. The goal is to release the answer to a function f. Rather than adding noise directly to the answer (which may be non-numeric), PTR estimates the *distance to instability* of the database d with respect to f, denoted $DistToInst_f(d)$. This quantity measures the distance from d to the closest database d' such that $f(d) \neq f(d')$, where adjacent databases are at distance 1. We will use the following two properties of $DistToInst$:

$$DistToInst_f(d) > 1 \to \forall d'.\ (Adj(d, d') \to f(d) = f(d'))$$
$$Adj(d, d') \to |DistToInst_f(d) - DistToInst_f(d')| \leq 1$$

Since the distance itself is private information, PTR first adds noise to the distance, calling the noisy result $dist$. If it is large enough, then PTR returns the value of f with no noise. Otherwise, PTR returns a default value $\perp$. In code:

$dist \xleftarrow{\$} \mathcal{L}_\epsilon(DistToInst_f(d))$;
if $dist > \ln(1/\delta)/\epsilon + 1$ then
  $r \leftarrow f(d)$;
else
  $r \leftarrow \perp$;
return r

The key to the privacy of PTR is that if the noise when estimating $dist$ is not too large, then there are two cases. If $dist$ is large, then there is no privacy cost for releasing the value of q when $dist$ is large. Otherwise, we return a default value $\perp$ revealing nothing. In either case, we just have privacy cost ϵ from computing the noisy distance. If the noise when estimating $dist$ is too large (happening with probability at most δ), we may violate privacy. So, we get an (ϵ, δ)-private algorithm.

Theorem 14. *PTR is (ϵ, δ)-differentially private for $\epsilon, \delta > 0$.*

Proof sketch. To prove differential privacy, we will use the rule [UTB-L]. We can take the event Θ to hold exactly when the noise is not too large:

$$\Theta \triangleq |dist - DistToInst_f(d)| < \ln(1/\delta)/\epsilon.$$

Then, we couple the noise to be the same, so that we take the same branch in both runs. Then, under assumption Θ, we know that if we estimate that the distance to instability is large and we take the first branch, then in fact $f(d\langle 1\rangle) = f(d\langle 2\rangle)$ as desired. Finally, using a tail bound for the Laplace distribution in aHL gives

$$\vdash_\delta c : \Phi\langle 1\rangle \Longrightarrow \Theta,$$

and rule [UTB-L] gives (ϵ, δ)-differential privacy. □

5. ADVANCED COMPOSITION

Advanced composition is a key tool for giving a more precise privacy analysis of a composition of private programs. In this section, we extend apRHL with a new loop rule that supports advanced composition for arbitrary invariants and we show how the rule can be applied for proving privacy of a non-interactive variant of $\mathsf{ASV}_{\mathrm{bt}}$.

Before presenting our solution, we stress that there are some technical challenges in extending apRHL with advanced composition. On the one hand, apRHL derives its expressiveness from its ability to reason about *arbitrary* approximate liftings, and not simply about approximate liftings for the equality relation; as a consequence, an advanced composition rule for apRHL should support approximate liftings in order to be useful (in fact, an advanced composition rule for just approximate liftings of equality would be too weak for verifying our running example). On the other hand, the proof of advanced composition is substantially more technical than the proof for the sequential composition theorem, using sophisticated results from fields such as martingale theory and hypothesis testing. We overcome these obstacles by showing approximate liftings from a version of differential privacy; as we saw before, differential privacy is also a consequence of an approximate lifting of equality. The key observation is that the two witnesses μ_L and μ_R used in the definition of an approximate lifting define a mechanism $\mu : \mathbb{B} \to \mathbf{Distr}(A_1 \times A_2)$ such that $\mu(\mathsf{true}) = \mu_L$ and $\mu(\mathsf{false}) = \mu_R$ where $\Delta_\epsilon(\mu_L, \mu_R) \leq \delta$ iff μ is (ϵ, δ)-differentially private. Next, we show how to take advantage of this observation.

Since the (ϵ, δ) parameters from approximate lifting are defined by the ϵ-distance $\Delta_\epsilon(\mu_1, \mu_2)$ between the two witnesses, we will first show an advanced composition theorem for ϵ-distance.

Proposition 15 (Advanced composition for ϵ-distance)**.** *Let $f_i, g_i : A \to \mathbf{Distr}(A)$ such that $\Delta_\epsilon(f_i(a), g_i(a)) \leq \delta$ for every $a \in A$. For any $\omega > 0$, let:*

$$\epsilon^* \triangleq \left(\sqrt{2n\ln(1/\omega)}\right)\epsilon + n\epsilon(e^\epsilon - 1) \quad \textit{and} \quad \delta^* \triangleq n\delta + \omega.$$

Then for every $n \in \mathbb{N}$ and $a \in A$, $\Delta_{\epsilon^*}(f^n(a), g^n(a)) \leq \delta^*$.

Proof. Let $h_i : A \to \mathbb{B} \to \mathbf{Distr}(A)$ be such that for every $a \in A$, $h_i(a, \mathsf{true}) = f_i(a)$ and $h_i(a, \mathsf{false}) = g_i(a)$. Then $\Delta_\epsilon(f_i(a), g_i(a)) \leq \delta$ iff $h_i(a) : \mathbb{B} \to \mathbf{Distr}(A)$ is (ϵ, δ)-differentially private for every $a \in A$.

By directly applying the advanced composition theorem of differential privacy (Theorem 7), the function $h^n(a) : \mathbb{B} \to \mathbf{Distr}(A)$ is (ϵ^*, δ^*)-differentially private for each $a \in A$. So for every $b, b' \in \mathbb{B}$ and $a \in A$, $\Delta_{\epsilon^*}(h^n(a, b), h^n(a, b')) \leq \delta^*$. Now for every $a \in A$, $h^n(a, \mathsf{true}) = f^n(a)$ and $h^n(a, \mathsf{false}) = g^n(a)$. Therefore, $\Delta_{\epsilon^*}(f^n(a), g^n(a)) \leq \delta^*$. □

Now that we have an advanced composition for ϵ-distance, it is a simple matter to extend our result to approximate liftings. Note here that we apply advanced composition not to the distributions on A—which are related by an approximate lifting, but perhaps *not* related by differential privacy—but rather to the two *witnesses* of the lifting, distributions on pairs in $A \times A$.

Proposition 16 (Advanced composition for lifting)**.** *Let $f_i, f_i' : A \to \mathbf{Distr}(A)$ and $\Phi \subseteq A \times A$ such that for every $a, a' \in A$, $(a, a') \models \Phi$ implies*

$$f_i(a)\ \Phi^{\sharp(\epsilon,\delta)}\ f_i'(a').$$

Let $n \in \mathbb{N}$ and let (ϵ', δ') be as in Theorem 7: For any $\omega > 0$, let

$$\epsilon^* \triangleq \left(\sqrt{2n\ln(1/\omega)}\right)\epsilon + n\epsilon(e^\epsilon - 1) \quad \textit{and} \quad \delta^* \triangleq n\delta + \omega.$$

Then for every $a, a' \in A$ such that $(a, a') \models \Phi$, we have

$$f^n(a)\ \Phi^{\sharp(\epsilon^*,\delta^*)}\ f'^n(a').$$

Proof. We can map any pair $(a, a') \in \Phi$ to the left and right witnesses of the approximate lifting. That is, there exists $h_i^l, h_i^r : (A \times A) \to \mathbf{Distr}(A \times A)$ such that for every $(a, a') \models \Phi$:

- $\pi_1(h_i^l(a, a')) = f_i(a)$ and $\pi_2(h_i^r(a, a')) = f_i'(a')$
- $\mathsf{supp}(h_i^l(a, a')) \subseteq \Phi$ and $\mathsf{supp}(h_i^r(a, a')) \subseteq \Phi$
- $\Delta_\epsilon(h_i^l(a, a'), h_i^r(a, a')) \leq \delta$

Without loss of generality, we can assume that $h_i^l(a, a') = h_i^r(a, a') = 0$ if $(a, a') \models \neg\Phi$. By Proposition 15, we also have $\Delta_{\epsilon^*}((h^l)^n(a, a'), (h^r)^n(a, a')) \leq \delta^*$ for every $(a, a') \models \Phi$. By induction on n, for every $(a, a') \models \Phi$ we have $\mathsf{supp}((h^l)^n(a, a')) \subseteq \Phi$ and $\mathsf{supp}((h^r)^n(a, a')) \subseteq \Phi$, $\pi_1((h^l)^n(a, a')) = f^n(a)$ and $\pi_2((h^r)^n(a, a')) = f'^n(a')$. □

We remark that our connection between the witnesses of liftings and differential privacy allows us to directly import other composition theorems of differential privacy and their proofs without change. For instance, Kairouz et al. [32] consider two variants of advanced composition: an optimal variant that provably gives the best bound on ϵ and δ, and a heterogeneous variant that allows ϵ and δ be different for the different mechanisms. In unpublished work, Rogers et al. [43] consider a version of the advanced composition theorem where the privacy level ϵ_i and δ_i for the i-th mechanism may be chosen *adaptively*, i.e., depending on the results from the first $i - 1$ mechanisms. These composition theorems are quite tricky to prove, involving sophisticated tools from martingale theory and hypothesis testing. We expect that we can internalize all of these composition theorems—and directly generalize to liftings—with minimal effort.

Based on the previous result, we introduce a new rule [AC-WHILE] that formalizes advanced composition for loops. The soundness for the new rule, which is given in Fig. 4 follows immediately from the results of the previous section.

Theorem 17. *The rule* [AC-WHILE] *is sound.*

6. INTERACTIVE PRIVACY

So far, we have seen how to incorporate composition theorems and accuracy proofs into our logic. Now, we consider the last piece needed to verify $\mathsf{ASV}_{\mathrm{bt}}$: proving privacy for *interactive* algorithms. To date, privacy has only been formally verified for algorithms where the entire input is available in a single piece; such algorithms are called *offline* algorithms. In contrast, *interactive* or *online* algorithms accept input piece by piece, in a finite stream of input, and must produce an intermediate outputs as inputs arrive.

The differential literature proposes several interactive algorithms; examples include private counters [14, 21], the Sparse Vector mechanism, and other algorithms using these mechanisms [30]. The main difficulty in verifying privacy is to model *adaptivity*: later inputs can depend on earlier outputs. Indeed, differential privacy behaves well under adaptivity, a highly useful property enabling applications to adaptive data analysis and statistics [23].

We can view adaptive inputs as controlled by an *adversary*, who receives earlier outputs and selects the next input. We draw on techniques for formally verifying cryptographic proofs, which often also involve an adversary who is trying to break the protocol. We take inspiration from the treatment of adversaries in the logic pRHL, an exact version of apRHL that has been used for verifying cryptographic proofs [4]. Specifically, we extend apRHL with a rule [ADV] for the adversary. The rule, displayed in Figure 5, generalizes the adversary rule from pRHL; let Φ an assertion that does not contain any adversary variable, and assume that the adversary $\mathcal{A}$ has access to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$ and that each oracle guarantees equality of outputs and an invariant Φ, provided it is called on equal inputs that satisfy Φ. Then, $\mathcal{A}$ guarantees equality of outputs and an invariant Φ, provided it is called on equal inputs that satisfy Φ. Moreover, the privacy cost of calling the adversary $\mathcal{A}$ is equal to $\langle \sum_{k=1}^n q_k\epsilon_k, \sum_{k=1}^n q_k\delta_k \rangle$ where $\langle \epsilon_i, \delta_i \rangle$ is the cost of calling once the oracle $\mathcal{O}_i$, and q_i is the maximal number of adversarial queries for oracle $\mathcal{O}_i$. One can prove the soundness of the adversary rule by induction on the code of the adversary.

Proposition 18. *The rule* [ADV] *is sound.*

Note that the proof of sparse vector only makes a restricted use of the [ADV] rule: as $\mathcal{A}$ does not have access to any oracle, the pRHL rule suffices for the proof. However, the following, oracle based, presentation of sparse vector uses the full power of the [ADV] rule:

$$\begin{array}{l} l \leftarrow []; \\ u \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(0); \\ A \leftarrow a - u; \\ B \leftarrow b + u; \\ x \leftarrow \mathcal{A}^{\mathcal{O}}(); \\ \mathsf{return}\ l \end{array}$$

where $\mathcal{O}$ is an oracle that takes a query and checks whether it is between thresholds and updates a public list l, and $\mathcal{A}$ is allowed to query $\mathcal{O}$ up to N times. In addition, we note that our new rule can also be useful for cryptographic proofs which involve reasoning about statistical distance.

7. OPTIMAL SUBSET COUPLING

The privacy proof of $\mathsf{ASV}_{\mathrm{bt}}$ relies on a new interval coupling rule [LAPINT] for Laplace sampling, Fig. 6. This rule allows us to relate a larger interval with a smaller interval nested inside. That is, we can assume that the sample $y_1\langle 1 \rangle$ lies in $[p, q]$ if and only if the sample $y_2\langle 2 \rangle$ lies in $[r, s]$ contained in $[p, q]$. The privacy cost depends on two things: the difference in sizes of the two intervals $(q - p) - (s - r)$, and the size of the inner interval $s - r$. Roughly, a larger inner interval and smaller outer interval yield a smaller privacy cost.

To show that this rule is sound, we will first prove a general construction for liftings of the form

$$\mu\ (y_1 \in P \leftrightarrow y_2 \in Q)^{\sharp(\epsilon,0)}\ \mu$$

for $Q \subseteq P$. We call such such liftings *subset couplings*, since they relate a set of outputs to a subset of outputs. Our construction applies

$$\text{AC-While}\ \frac{\models \Theta \wedge e\langle 1\rangle \le 0 \to \neg b_1\langle 1\rangle \qquad \epsilon^* \triangleq \left(\sqrt{2n\ln(1/\omega)}\right)\epsilon + n\epsilon(e^\epsilon - 1) \qquad \delta^* \triangleq n\delta + \omega \qquad \omega > 0}{\begin{array}{c}\vdash c_1 \sim_{\langle\epsilon,\delta\rangle} c_2 : \Theta \wedge b_1\langle 1\rangle \wedge b_2\langle 2\rangle \wedge e\langle 1\rangle = k \implies \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle < k \\ \hline \vdash \mathsf{while}\ b_1\ \mathsf{do}\ c_1 \sim_{\langle\epsilon^*,\delta^*\rangle} \mathsf{while}\ b_2\ \mathsf{do}\ c_2 : \Theta \wedge b_1\langle 1\rangle = b_2\langle 2\rangle \wedge e\langle 1\rangle \le n \implies \Theta \wedge \neg b_1\langle 1\rangle \wedge \neg b_2\langle 2\rangle\end{array}}$$

Figure 4: Advanced composition rule

$$\text{Adv}\ \frac{\forall i, \vec{y}, \vec{z}.\ \vdash \vec{y} \leftarrow \mathcal{O}_i(\vec{z}) \sim_{\langle\epsilon_i,\delta_i\rangle} \vec{y} \leftarrow \mathcal{O}_i(\vec{z}) : \vec{z}\langle 1\rangle = \vec{z}\langle 2\rangle \wedge \Phi \implies \vec{y}\langle 1\rangle = \vec{y}\langle 2\rangle \wedge \Phi}{\vdash \vec{x} \leftarrow \mathcal{A}(\vec{e}) \sim_{\langle \sum_{k=1}^n q_k\epsilon_k, \sum_{k=1}^n q_k\delta_k\rangle} \vec{x} \leftarrow \mathcal{A}(\vec{e}) : \vec{e}\langle 1\rangle = \vec{e}\langle 2\rangle \wedge \mathbf{x}_{\mathcal{A}}\langle 1\rangle = \mathbf{x}_{\mathcal{A}}\langle 2\rangle \wedge \Phi \implies \vec{x}\langle 1\rangle = \vec{x}\langle 2\rangle \wedge \mathbf{x}_{\mathcal{A}}\langle 1\rangle = \mathbf{x}_{\mathcal{A}}\langle 2\rangle \wedge \Phi}$$

where $q_1, \ldots, q_n$ are the maximal number of queries that $\mathcal{A}$ can make to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$ and $\mathbf{x}_{\mathcal{A}}$ is the state of the adversary.

Figure 5: Adversary rule

to all discrete distributions, and is *optimal* in a precise sense: among all liftings of the relation, our construction gives the smallest (i.e., the most precise) ϵ.

Theorem 19 (Optimal subset coupling)**.** *Let μ be a distribution over S, and consider two proper subsets P, Q of S such that $Q \subseteq P$. Then $\mu(P) \le \alpha\mu(Q)$ if and only if the following lifting holds:*

$$\mu\ R^{\sharp(\ln\alpha, 0)}\ \mu$$

where the relation R is defined by the following clause:

$$(y_1, y_2) \in R \triangleq y_1 \in P \leftrightarrow y_2 \in Q$$

Proof. The reverse direction follows from the fundamental lemma of approximate liftings [2]. For the forward implication, we construct two witnesses. Note that the theorem is trivial if $\mu(P \setminus Q) = 0$ since we can just take the identity coupling, so we will assume otherwise. Define the following witnesses:

$$\mu_L(x,y) = \begin{cases} \mu(x) & \text{if } x \notin P \setminus Q \wedge x = y \\ \frac{\mu(x)\mu(y)}{\mu(Q)} & \text{if } x \in P \setminus Q \wedge y \in Q \\ 0 & \text{otherwise.} \end{cases}$$

$$\mu_R(x,y) = \begin{cases} \mu(y) & \text{if } x = y \wedge y \notin P \\ \lambda \cdot \mu(y) & \text{if } x = y \wedge y \in Q \\ \frac{(1-\lambda)\mu(x)\mu(y)}{\mu(P\setminus Q)} & \text{if } x \in P \setminus Q \wedge y \in Q \\ \mu(y) & \text{if } x = x_0 \in S \setminus P \wedge y \in P \setminus Q \\ 0 & \text{otherwise.} \end{cases}$$

Here, x_0 is an arbitrary element of $S \setminus P$. We set $\lambda = \mu(Q)/\mu(P)$. Note that λ satisfies:

$$\lambda = (1-\lambda)\frac{\mu(Q)}{\mu(P \setminus Q)}.$$

For the witnesses, it is not hard to see that the marginal conditions are satisfied, and that $x\ R\ y$ for all pairs (x, y) in the supports of μ_L and μ_R. Furthermore, for all $x \in P$ and $y \in Q$, we have $\mu_L(x,y) = (1/\lambda) \cdot \mu_R(x,y)$ by our choice of λ. By the condition on marginals and the support, we have a $(\ln(1/\lambda), 0)$-lifting. This immediately implies the $(\ln\alpha, 0)$-lifting for any $\alpha \ge 1/\lambda = \mu(P)/\mu(Q)$. □

This result is very much in the spirit of the optimal or *maximal* coupling for exact probabilistic couplings (see, e.g., [49]). By computing the total variation distance between two distributions, the maximal coupling construction shows how to create a coupling that exactly realizes the total variation distance.

Similarly, by the optimal subset coupling, we can construct a subset coupling and calculate the distance ϵ in the lifting by simply proving a property of the discrete Laplace distribution. Formally, let $\mathcal{L}_\epsilon(v)$ for $v \in \mathbb{Z}$ have distribution over $\mathbb{Z}$ with probability proportional to

$$\Pr[r] \propto \exp(-\epsilon|r - v|).$$

We write $\mathcal{L}_\epsilon$ to mean $\mathcal{L}_\epsilon(0)$. We will use the following property, a discrete version of Bun et al. [13, Claim 5.13].

Lemma 20. *Let r be a draw from $\mathcal{L}_\epsilon$, and take $a, a', b, b' \in \mathbb{Z}$ such that $a < b$ and $[a, b] \subseteq [a', b']$. Then,*

$$\Pr[r \in [a', b']] \le \alpha \Pr[r \in [a, b]]$$

with

$$\alpha = \frac{\exp(\eta\epsilon)}{1 - \exp(-(b - a + 2)\epsilon/2)}, \qquad \eta = (b' - a') - (b - a).$$

The proof follows by a small calculation; we defer the details to the full version. With this property, we can now prove soundness of our subset rule for sampling from the Laplace distribution.

Theorem 21. *The rule* [LapInt] *is sound.*

Proof. Suppose that $[\![e\langle 1\rangle]\!] = v_1, [\![e\langle 2\rangle]\!] = v_2$, and $|v_1 - v_2| = \Delta \le k$. Let the noises be $w_1 = x\langle 1\rangle - v_1, w_2 = x\langle 2\rangle = v_2$; note that both samples are distributed as $\mathcal{L}_\epsilon(0)$. Note that $x\langle 1\rangle \in [p, q]$ exactly when $w_1 \in [p - v_1, q - v_1]$, and similarly for x_2 and w_2. By Proposition 12, to show a lifting on memories, it suffices to find a lifting on the distribution over the sampled variable y, taking f to be the function that maps a memory m to the value $m(y)$. So, it suffices to show

$$\mathcal{L}_\epsilon(0)\ \{w_1 \in I_1 \leftrightarrow w_2 \in I_2\}^{\sharp(\epsilon', 0)}\ \mathcal{L}_\epsilon(0)$$

where $I_1 \triangleq [p - v_1, q - v_1]$ and $I_2 \triangleq [r - v_2, s - v_2]$. Since $p + k \le r$ and $s \le q - k$ and $|v_1 - v_2| \le k$, we know $I_2 \subseteq I_1$. Then, we can directly apply Lemma 20 on these two intervals and we are done. □

8. PROVING PRIVACY FOR $\mathsf{ASV}_{\mathrm{bt}}$

We verify differential privacy for the full version of $\mathsf{ASV}_{\mathrm{bt}}$ with an adaptive adversary that chooses its queries interactively. We recall the theorem.

Theorem 22. *Let ϵ and δ both be in $(0, 1)$. Set*

$$\epsilon' \triangleq \frac{\epsilon}{4\sqrt{2M\ln(2/\delta)}}.$$

$$\textsc{LapInt}\ \frac{\epsilon' \triangleq \ln\left(\frac{\exp(\eta\epsilon)}{1-\exp(-\sigma\epsilon/2)}\right) \qquad \Phi \triangleq |e\langle 1\rangle - e\langle 2\rangle| \le k \wedge (p+k \le r < s \le q-k) \wedge (q-p)-(s-r) \le \eta \wedge 0 < \sigma \le (s-r)+2}{\vdash y_1 \xleftarrow{\$} \mathcal{L}_\epsilon(e) \sim_{\langle\epsilon',0\rangle} y_2 \xleftarrow{\$} \mathcal{L}_\epsilon(e) : \Phi \Longrightarrow y_1\langle 1\rangle \in [p,q] \leftrightarrow y_2\langle 2\rangle \in [r,s]}$$

Figure 6: Interval coupling for Laplace

If all adversarial queries q are 1-sensitive (i.e. $|\mathsf{evalQ}(q,d) - \mathsf{evalQ}(q,d')| \le 1$ for every adjacent databases d and d'), then $\mathsf{ASV}_{\mathrm{bt}}$ is (ϵ,δ)-differentially private. Formally,

$$\vdash \mathsf{ASV}_{\mathrm{bt}} \sim_{\langle\epsilon,\delta\rangle} \mathsf{ASV}_{\mathrm{bt}} : \Phi \Longrightarrow r\langle 1\rangle = r\langle 2\rangle$$

where Φ is defined as

$$=_{\{a,b,N,qs\}} \wedge \mathsf{Adj}(d\langle 1\rangle, d\langle 2\rangle) \wedge b\langle 1\rangle - a\langle 1\rangle \ge \gamma$$

for

$$\gamma \triangleq \frac{6}{\epsilon'}\ln(4/\epsilon') + \frac{4}{\epsilon}\ln(2/\delta).$$

Proof sketch. Now, we put everything together to prove $\mathsf{ASV}_{\mathrm{bt}}$ is (ϵ,δ)-private algorithm. We present just the key points of the proof here; the full proof is formalized in the EasyCrypt system. We first transform the algorithm to an equivalent algorithm:[3]

$\mathsf{ASV}_{\mathrm{bt}}(a,b,M,N,d) :=$
$i \leftarrow 0; l \leftarrow [];$
$u \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(0);$
$A \leftarrow a-u; B \leftarrow b+u;$
while $i < N \wedge |l| < M$ **do**
  $i' \leftarrow i; hd \leftarrow -1;$
  while $i' < N$ **do**
    if $(hd = -1)$
      $q \leftarrow \mathcal{A}(l);$
      $S \xleftarrow{\$} \mathcal{L}_{\epsilon'/3}(\mathsf{evalQ}(q,d));$
      if $(A \le S \le B)$ **then** $hd \leftarrow i;$
      $i \leftarrow i+1;$
    $i' \leftarrow i'+1;$
  if $(hd \ne -1)$ **then** $l \leftarrow hd :: l;$
return l

The main change is that the loop iterations in Fig. 1 are grouped into blocks of queries, each handled by an inner loop. Each outer iteration now corresponds to finding a single query between the thresholds. The inner iterations loop through the queries until there is a between threshold query. To allow the inner loop to be analyzed in a synchronized fashion, the inner loop always continues up to iteration N, doing nothing for all iterations beyond the first between threshold query.

This transformation allows us to use both advanced composition and pointwise equality. At a high level, we follow four steps. First, we set up the threshold noise, so that the noisy interval $[A,B]$ is smaller in the first run than in the second run; this costs a bit of privacy. Then, we handle the loop working inside to out: we apply the adversary rule, the subset coupling, and pointwise equality to the inner loop to show privacy for each block of queries that stops as soon as we see a between threshold query, assuming that the noisy intervals $[A,B]$ are sufficiently large. Next, we apply advanced composition to bound the privacy cost of the outer loop, still assuming $[A,B]$ is sufficiently large. Finally, we use up-to-bad reasoning to remove this assumption, increasing δ slightly for the final (ϵ,δ)-privacy bound.

We now detail each step. To reduce notation, we will suppress the adjacency predicate $Adj(d\langle 1\rangle, d\langle 2\rangle)$ which is preserved throughout the computation and the proof.

Threshold coupling. Let c_t be the initialization command, including all commands before the loop. First, we can prove:

$$\vdash c_t \sim_{\langle \frac{\epsilon}{2},0\rangle} c_t : \Phi \Longrightarrow \Phi'$$

where $\Phi' \triangleq A\langle 1\rangle + 1 = A\langle 2\rangle \wedge B\langle 1\rangle = B\langle 2\rangle + 1$, by applying the rule [LapGen] to ensure $u\langle 1\rangle = u\langle 2\rangle + 1$ as a post-condition. This step costs $(\epsilon/2, 0)$ privacy.

The inner loop. For the inner loop, we will assume the threshold condition Φ' and $l\langle 1\rangle = l\langle 2\rangle$ initially; both conditions are preserved by the inner loop. We will also assume that the noisy interval $[A,B]$ is sufficiently large:[4]

$$\Psi \triangleq B - A \ge \frac{6}{\epsilon'}\ln(4/\epsilon').$$

Let c_i be the inner loop. We will first prove the pointwise judgment:

$$\vdash c_i \sim_{\langle\epsilon',0\rangle} c_i : \Phi' \wedge \Psi\langle 1\rangle \wedge l\langle 1\rangle = l\langle 2\rangle \Longrightarrow (hd\langle 1\rangle = v) \to (hd\langle 2\rangle = v) \tag{1}$$

for each index v.

We focus on the case where $0 \le v \le N$, as other cases are easy. First, we apply the [While] rule, with $\epsilon_i = 0$ except for $i = v$, where we set $\epsilon_v = \frac{\epsilon}{2}$. Whenever we call the adversary for the next query, we know that $l\langle 1\rangle = l\langle 2\rangle$, so we may apply adversary rule with cost $(0,0)$ to ensure that $q\langle 1\rangle = q\langle 2\rangle$ throughout.

Then, the proof for the rest of the loop body goes as follows:

- For the iterations $i < v$ and $i > v$, we use the rule [LapNull] to couple the noisy query answers $S\langle 1\rangle, S\langle 2\rangle$. This has no privacy cost and preserves the invariant.
- For the iteration $i = v$, suppose that $S\langle 1\rangle \in [A\langle 1\rangle, B\langle 1\rangle]$ (otherwise we are done). We can apply a coupling for the Laplace distribution, [LapInt], to ensure that $S\langle 2\rangle \in [A\langle 2\rangle, B\langle 2\rangle]$ as well. Under $\Psi\langle 1\rangle$ and the coupling on the noisy thresholds Φ', the inner interval $[A\langle 2\rangle, B\langle 2\rangle]$ has size at least $(6/\epsilon')\ln(4/\epsilon') - 2$. Taking $(p,q,r,s) = (A\langle 1\rangle, B\langle 1\rangle, A\langle 2\rangle, B\langle 2\rangle)$, $\eta = 2$, $\sigma = (6/\epsilon')\ln(4/\epsilon')$, and $k = 1$, a calculation shows that [LapInt] gives a $(\epsilon', 0)$-lifting so the critical iteration has privacy cost ϵ'.

This establishes Eq. (1). By the pointwise equality rule [PW-Eq], we have:

$$\vdash c_i \sim_{\langle\epsilon',0\rangle} c_i : \Phi' \wedge \Psi\langle 1\rangle \wedge l\langle 1\rangle = l\langle 2\rangle \Longrightarrow hd\langle 1\rangle = hd\langle 2\rangle$$

[3] We have formally verified equivalence of this program with the program in Fig. 1 by using a recently-proposed asynchronous loop rule [3].

[4] While Ψ does not have tagged variables, we will later interpret A and B as coming from the first run.

By [FRAME] and some manipulations, we can assume that $l\langle 1\rangle = l\langle 2\rangle$ at the end of each iteration of the outer loop.

The outer loop. For the outer loop, we apply advanced composition. Letting c_o be the outer loop, our choice of ϵ' and corresponds to the setting in Theorem 7, so we have the following judgment by [AC-WHILE]:

$$\vdash c_o \sim_{\langle \epsilon/2,0\rangle} c_o : l\langle 1\rangle = l\langle 2\rangle \wedge \Phi' \wedge \Psi\langle 1\rangle \Longrightarrow l\langle 1\rangle = l\langle 2\rangle.$$

Since c_o does not modify the thresholds and preserves $\Psi\langle 1\rangle$, [FRAME] and some manipulations allows us to move this assertion into the post-condition:

$$\vdash c_o \sim_{\langle \epsilon/2,0\rangle} c_o : l\langle 1\rangle = l\langle 2\rangle \wedge \Phi' \Longrightarrow \Psi\langle 1\rangle \rightarrow l\langle 1\rangle = l\langle 2\rangle.$$

Applying up-to-bad reasoning. Finally, we can apply [SEQ] with our judgement for the initialization c_i and the outer loop c_o, giving:

$$\vdash \mathsf{ASV}_{\mathrm{bt}} \sim_{\langle \epsilon/2,0\rangle} \mathsf{ASV}_{\mathrm{bt}} : \Phi_0 \Longrightarrow \Psi\langle 1\rangle \rightarrow l\langle 1\rangle = l\langle 2\rangle$$

for

$$\Phi_0 \triangleq =_{\{a,b,N,qs\}} \wedge \mathsf{Adj}(d\langle 1\rangle, d\langle 2\rangle) \wedge b\langle 1\rangle - a\langle 1\rangle \geq \gamma.$$

To conclude the proof, all that remains is to remove the assertion $\Psi\langle 1\rangle$. We will bound the probability that $\Psi\langle 1\rangle$ does not hold. The accuracy rule for the Laplace mechanism gives

$$\vdash_\delta u \xleftarrow{\$} \mathcal{L}_{\epsilon/2}(0) : b - a \geq \gamma \Longrightarrow |u| \leq \frac{2}{\epsilon}\log(1/\delta),$$

from which we can conclude

$$\vdash_\delta \mathsf{ASV}_{\mathrm{bt}} : b - a \geq \gamma \Longrightarrow \Psi.$$

Finally, applying [UTB-L] yields the desired judgment:

$$\vdash \mathsf{ASV}_{\mathrm{bt}} \sim_{\langle \epsilon/2,\delta\rangle} \mathsf{ASV}_{\mathrm{bt}} : \Phi_0 \Longrightarrow l\langle 1\rangle = l\langle 2\rangle \qquad \square$$

9. RELATED WORKS

Differential privacy [19] has been an area of intensive research in the last decade. We refer readers interested in a more comprehensive treatment of the algorithmic aspects of differential privacy to the excellent monograph by Dwork and Roth [17]. Several tools have been developed to support the development of differentially private data analysis. PINQ [36] internalizes the use of standard composition in the form of a privacy budget management platform, Airavat [45] uses differential privacy combined with the map-reduce approach, GUPT [40] implements the general idea of sample and aggregate [41]. Other tools implement algorithms targeting specific applications like: location data [35], genomic data [27, 47], mobility data [37], and browser error reports [26].

Several tools have been proposed for providing formal verification of the differential privacy guarantee, using a wide variety of verification approaches: dynamic checking [24, 36], relational program logic [2, 6] and relational refinement type systems [9], linear (dependent) type systems [29, 42], product programs [7], methods based on computing bisimulations families for probabilistic automata [50, 51], and methods based on counting variants of satisfiability modulo theories [28]. None of these techniques can handle advanced composition, interactive online algorithms and privacy depending on accuracy. Barthe et al. [5] present a system for reasoning about *computational differential privacy* [39] a relaxation of differential privacy where the adversary are computationally-bound.

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 [31], who use couplings to reason about generalized Feistel networks, Mironov [38] first used this technique in his analysis of RC4. There are seemingly few applications of coupling in formal verification, despite considerable research on probabilistic bisimulation (first introduced by Larsen and Skou [33]) and probabilistic relational program logics (first introduced by Barthe et al. [4]). The connection between liftings and couplings was recently noted by Barthe et al. [8] and explored for differential privacy by Barthe et al. [10]. The latter uses a coupling argument to prove differentially private the sparse vector algorithm that we also consider in this work. The additional challenges that we face are: first, the integration of advanced composition, providing a much better privacy bound; second, the proof that sparse vector is differentially private also in the interactive model, which requires additionally to have a logic that permits to reason about the adversary. Moreover, Barthe et al. [10] do not provide methods to prove privacy using accuracy.

In promising recent work, Zhang and Kifer [52] design a system to automatically verify differentially privacy for examples where the privacy proof uses tools beyond the standard composition theorem, including the Sparse Vector technique. Their proof strategy is morally similar to couplings, but their work uses a combination of product programs and lightweight refinement types backed by novel type-inference techniques, rather than a relational program logic like we consider. Their system can also optimize the privacy cost, something that we do not consider. While their work is highly automated, their system is limited to pure, $(\epsilon, 0)$ differential privacy, so it cannot verify the algorithms we consider, where privacy follows from accuracy or the advanced composition theorem. Their techniques also seem limited to couplings from bijections; in particular, it is not clear how to prove privacy for examples that use more advanced couplings like the optimal subset coupling.

10. CONCLUDING REMARKS

We have presented an extension of the logic apRHL [6] that can express three classes of privacy proofs beyond current state-of-the-art techniques for privacy verification: privacy depending on accuracy, privacy from advanced composition, and privacy for interactive algorithm. We have formalized a generalization of the adaptive Sparse Vector algorithm, known as Between Thresholds [13]. This and other possible generalizations of sparse vector could bring interesting results in domains like geo-indistinguishability [1].

For the future, it would be interesting to explore generalizations of differential privacy like the recent notion of *concentrated differential privacy* [12, 18]. This generalization features a simple composition principle that internalizes the advanced composition principle of standard differential privacy. However, it is currently unclear whether the definition of concentrated differential privacy, which involves Rényi divergences, can be modeled using apRHL.

Additionally, there is still room for improving the expressivity of apRHL for differential privacy. One interesting example combining accuracy and privacy is the *large margin mechanism* [15]. The privacy proof for this algorithm requires careful reasoning about the size of the support when applying pointwise equality, and sophisticated facts about the accuracy Sparse Vector. This example seems beyond the reach of our techniques, but we believe it could be handled by generalizing the existing rules.

Finally, it would be interesting to explore a tighter integration of accuracy and privacy proofs. We currently use two systems, aHL and apRHL, to verify privacy. This can lead to awkward proofs since the two logics can only interact in specific places in the proof (i.e., the up-to-bad rules). A combined version of the logics could allow more natural proofs.

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *ACM SIGSAC Conference on Computer and Communications Security (CCS), Berlin, Germany*, pages 901–914, 2013.

[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.

[3] G. Barthe, B. Grégoire, J. Hsu, and P.-Y. Strub. Coupling proofs are probabilistic product programs.

[4] 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.

[5] G. Barthe, G. Danezis, B. Grégoire, C. Kunz, and S. Z. Béguelin. Verified computational differential privacy with applications to smart metering. In *IEEE Computer Security Foundations Symposium (CSF), New Orleans, Louisiana*, pages 287–301, 2013.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] G. Barthe, M. Gaboardi, B. Grégoire, J. Hsu, and P.-Y. Strub. Proving differential privacy via probabilistic couplings. In *IEEE Symposium on Logic in Computer Science (LICS), New York, New York*, 2016.

[11] G. Barthe, M. Gaboardi, B. Grégoire, J. Hsu, and P.-Y. Strub. A program logic for union bounds. In *International Colloquium on Automata, Languages and Programming (ICALP), Rome, Italy*, 2016.

[12] M. Bun and T. Steinke. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. May 2016.

[13] M. Bun, T. Steinke, and J. Ullman. Make Up Your Mind: The Price of Online Queries in Differential Privacy. Apr. 2016.

[14] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14(3):26, 2011.

[15] K. Chaudhuri, D. J. Hsu, and S. Song. The large margin mechanism for differentially private maximization. In *Conference on Neural Information Processing Systems (NIPS), Montréal, Québec*, pages 1287–1295, 2014.

[16] C. Dwork and J. Lei. Differential privacy and robust statistics. In *ACM SIGACT Symposium on Theory of Computing (STOC), Bethesda, Maryland*, pages 371–380, 2009.

[17] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[18] C. Dwork and G. N. Rothblum. Concentrated Differential Privacy. Mar. 2016.

[19] 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.

[20] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *ACM SIGACT Symposium on Theory of Computing (STOC), Bethesda, Maryland*, pages 381–390, 2009.

[21] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *ACM SIGACT Symposium on Theory of Computing (STOC), Cambridge, Massachusetts*, pages 715–724, 2010.

[22] C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *IEEE Symposium on Foundations of Computer Science (FOCS), Las Vegas, Nevada*, pages 51—60, 2010.

[23] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[24] 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.

[25] F. Eigner and M. Maffei. Differential privacy by typing in security protocols. In *IEEE Computer Security Foundations Symposium (CSF), New Orleans, Louisiana*, pages 272–286, 2013.

[26] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC Conference on Computer and Communications Security (CCS), Scottsdale, Arizona*, pages 1054–1067, 2014.

[27] S. E. Fienberg, A. B. Slavkovic, and C. Uhler. Privacy preserving GWAS data sharing. In *IEEE International Conference on Data Mining Workshops (ICDMW), Vancouver, British Colombia*, pages 628–635, 2011.

[28] M. Fredrikson and S. Jha. Satisfiability modulo counting: a new approach for analyzing privacy properties. In *IEEE Symposium on Logic in Computer Science (LICS), Vienna, Austria*, pages 42:1–42:10, 2014.

[29] 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.

[30] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *IEEE Symposium on Foundations of Computer Science (FOCS), Las Vegas, Nevada*, pages 61–70, 2010.

[31] 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.

[32] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. 2015.

[33] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *ACM Symposium on Principles of Programming Languages (POPL), Austin, Texas*, pages 344–352, 1989.

[34] M. Lyu, D. Su, and N. Li. Understanding the sparse vector technique for differential privacy. 2016.

[35] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *International Conference on Data Engineering (ICDE), Cancún, México*, pages 277–286, 2008.

[36] F. McSherry. Privacy integrated queries. In *ACM SIGMOD International Conference on Management of Data (SIGMOD), Providence, Rhode Island*, 2009.

[37] D. J. Mir, S. Isaacman, R. Cáceres, M. Martonosi, and R. N. Wright. DP-WHERE: differentially private modeling of human mobility. In *IEEE International Conference on Big Data (ICBD), Santa Clara, California*, pages 580–588, 2013.

[38] 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.

[39] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. Computational differential privacy. In *IACR International Cryptology Conference (CRYPTO), Santa Barbara, California*, pages 126–142, 2009.

[40] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. E. Culler. GUPT: privacy preserving data analysis made easy. In *ACM SIGMOD International Conference on Management of Data (SIGMOD), Scottsdale, Arizona*, pages 349–360, 2012.

[41] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *ACM SIGACT Symposium on Theory of Computing (STOC), San Diego, California*, 2007.

[42] 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.

[43] R. Rogers, A. Roth, J. Ullman, and S. Vadhan. Privacy odometers and filters: Pay-as-you-go composition. 2016.

[44] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *ACM SIGACT Symposium on Theory of Computing (STOC), Cambridge, Massachusetts*, pages 765–774, 2010.

[45] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for MapReduce. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI), San Jose, California*, pages 297–312, 2010.

[46] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS), Boulder, Colorado*, pages 1310–1321, 2015.

[47] S. Simmons, C. Sahinalp, and B. Berger. Enabling privacy-preserving GWAS in heterogenous human populations. In *RECOMB*, 2016.

[48] A. Thakurta and A. Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Computational Learning Theory (CoLT), Princeton, New Jersey*, pages 819–850, 2013.

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

[50] 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.

[51] 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.

[52] D. Zhang and D. Kifer. AutoPriv: Automating differential privacy proofs. 2016.