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 the observation that differential privacy has deep connections with a generalization of 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.

1. Introduction
Differential privacy is a rigorous definition of statistical privacy proposed by Dwork, McSherry, Nissim and Smith [12], 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. 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 privately 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 [11].)
- 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 [11].)
- The Exponential mechanism [16], 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]. Although the two concepts are tightly connected, their relationship has been little explored.

2. A view of differential privacy as a form of approximate probabilistic liftings [2, 4], a generalization of probabilistic liftings used in probabilistic process algebra [13].

We elaborate on these points, and then present our contributions.

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 \( R^* \subseteq T A \times T B \), 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 in the study 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 [10].

While research in this area has traditionally focused on 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 tool in probability theory [15, 17]. 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 avenues for applying 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 [7] are the most flexible techniques known for reasoning formally about differentially private computations. Their expressive
power stems from their use of approximate probabilistic liftings, a generalization of probabilistic liftings based on a notion of distance between distributions; 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, provided that one pays sufficient privacy. 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, and provide the key 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, but has been little-studied in the differential privacy literature. The one-sided Laplace mechanism nicely illustrates the benefits of our approach: although it is not differentially private, its properties can be captured formally by approximate probabilistic liftings. These properties can be combined to show privacy for a larger program. These discussions are deferred to the extended version.

An extended probabilistic relational program logic To demonstrate our techniques, we take the relational program logic apRHL [4] as our starting point. 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 is in terms of 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 proof 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. An extended version of this paper [8] is available at http://arxiv.org/abs/1601.05047.

2. Generalized probabilistic liftings
To verify advanced algorithms like AboveT, we will leverage the power of approximate probabilistic liftings. In a sentence, our proofs will replace the sequential composition theorem of differential privacy—which is not strong 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.

2.1 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 1 (Coupling). There is a coupling between two sub-distributions \( \mu_1 \in \text{Distr}(B_1) \) and \( \mu_2 \in \text{Distr}(B_2) \) if there exists a sub-distribution (called the witness) \( \mu \in \text{Distr}(B_1 \times B_2) \) s.t. \( \pi_1(\mu) = \mu_1 \) and \( \pi_2(\mu) = \mu_2 \).

Probabilistic liftings are a special class of couplings.

Definition 2 (Lifting). Two sub-distributions \( \mu_1 \in \text{Distr}(B_1) \) and \( \mu_2 \in \text{Distr}(B_2) \) 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 \( \mu \in \text{Distr}(B_1 \times B_2) \) of \( \mu_1 \) and \( \mu_2 \) such that \( \text{supp}(\mu) \subseteq \Psi \).

Probabilistic liftings have many useful consequences. For example, \( \mu_1 \models^\sharp \mu_2 \) holds exactly when the sub-distributions \( \mu_1 \) and \( \mu_2 \) are equal. Less trivially, liftings can bind 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 \( E_1 \subseteq B_1 \), \( E_2 \subseteq B_2 \), \( \mu_1 \in \text{Distr}(B_1) \) and \( \mu_2 \in \text{Distr}(B_2) \). 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

\[ \Pr_{x_1 \sim \mu_1} [x_1 \in E_1] \leq \Pr_{x_2 \sim \mu_2} [x_2 \in E_2] \]

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 \( \mu_1, \mu_2 \in \text{SDistr}(B) \). 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 \( \mu_1, \mu_2 \in \text{Distr}(B) \). 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 $\mu_1 \Psi^\delta \mu_2$ for all $b \in B$, then $\mu_1 = \mu_2$.

2.2 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 3** (Barthe et al. [4]). Let $\varepsilon \geq 0$. The $\varepsilon$-DP divergence $\Delta_{\varepsilon}(\mu_1, \mu_2)$ between two sub-distributions $\mu_1 \in \text{Distr}(B)$ and $\mu_2 \in \text{Distr}(B)$ is defined as

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

The following proposition relates $\varepsilon$-DP divergence with $(\varepsilon, \delta)$-differential privacy.

**Proposition 4** (Barthe et al. [4]). A probabilistic computation $M : A \rightarrow \text{Distr}(B)$ is $(\varepsilon, \delta)$-differentially private w.r.t. an adjacency relation $\Phi$ iff for every two adjacent inputs $a$ and $a'$ there holds $a \Phi a'$.

We can use DP-divergence to define an approximate version of probabilistic liftings (which we will sometimes call probabilistic liftings). 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 $M : A \rightarrow \text{Distr}(B)$ is $(\varepsilon, \delta)$-differentially private w.r.t. an adjacency relation $\Phi$ iff there exists $(\delta_b)_{b \in B} \in \mathbb{R}^{\geq 0}$ such that $\sum_{b \in B} \delta_b \leq \delta$, and $M(a) \Psi^{\varepsilon, \delta_b} M(a')$ for every $b \in B$ and every two adjacent inputs $a$ and $a'$, where $\Psi_b = \{(x_1, x_2) \in B \times B \mid x_1 = b \Rightarrow x_2 = b\}$.

2.3 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 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 $\mathcal{L}_{\varepsilon}(v_1) \approx_{(k, \varepsilon, 0)} \mathcal{L}_{\varepsilon}(v_2)$.

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

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

where

$$\Psi = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{Z} \mid x_1 + k = x_2\}.$$
Assertions and judgments Assertions in the logic are first-order formulae over generalized expressions. The latter are expressions built from tagged variables \( x(1) \) and \( x(2) \), 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, \( x(1) = x(2) + 1 \) 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 \( \Phi \). The interpretation is standard (besides the use of tagged variables) and is omitted. By abuse of notation, we write \( e(1) \) or \( e(2) \), where \( e \) is a program expression, to denote the generalized expression built according to \( e \), but in which all variables are tagged with a \( (1) \) or \( (2) \), respectively. Judgments in both \( \text{apRHL} \) and \( \text{apRHL}^+ \) are of the form

\[
\vdash c_1 \sim_{(\epsilon, \delta)} c_2 : \Phi \rightarrow \Psi
\]

where \( c_1 \) and \( c_2 \) are statements, the precondition \( \Phi \) and postcondition \( \Psi \) are relational assertions, and \( \epsilon \) and \( \delta \) are non-negative reals. 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 \( \Phi \) are related by the \( (\epsilon, \delta) \)-lifting of the postcondition \( \Psi \). Formally, the judgment

\[
\vdash c_1 \sim_{(\epsilon, \delta)} c_2 : \Phi \rightarrow \Psi
\]

is valid iff for every two memories \( m_1 \) and \( m_2 \), such that \( m_1 \models [\Phi] m_2 \), we have

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

Proof system We defer the presentation of the proof system of \( \text{apRHL} \) to the extended version.

Figure 1 collects the new rules in \( \text{apRHL}^+ \), which are all derived from the new proof principles we saw in the previous section. The first rule \( \text{FORALL-Eq} \) allows proving differential privacy via pointwise privacy; this rule reflects Proposition 6.

The next pair of rules, \( \text{LapGen} \) and \( \text{LapNull} \), 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 \( \text{LapNull} \)—otherwise, the sample may change \( c_1 \) and \( c_2 \).

4. 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 theorems. 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

\[
r(1) = i \Rightarrow r(2) = i.
\]

The coupling proof goes as follows. We start by coupling the samplings of the noisy thresholds so that \( T(1) + 1 = T(2) \); 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 \( \text{LapNull} \). By 1-sensitivity of the queries and adjacency of the two databases, we know

\[
\text{eval}Q(Q[j], d) \leq 1,
\]

so

\[
S(1) < T(1) \Rightarrow S(2) < T(2).
\]

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

\[
r(1) = |Q| + 1 \Rightarrow r(2) = |Q| + 1 \land \forall r(1) < i,
\]

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

When we reach the \( i \)th iteration and \( i < |Q| + 1 \), we couple the samplings of \( S \) so that \( S(1) + 1 = S(2) \); the cost of this coupling is \((\epsilon/2, 0)\). Because \( T(1) + 1 = T(2) \) and \( S(1) + 1 = S(2) \), 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

\[
r(1) = i \Rightarrow r(2) = i.
\]

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

\[
\vdash \text{AboveT} \sim_{(\epsilon, 0)} \text{AboveT} : \Phi \rightarrow r(1) = r(2),
\]

where \( \Phi \) denotes the precondition

\[
\text{adj}(d(1), d(2)) \land t(1) = t(2) \land Q(1) = Q(2) \land \forall j. |\text{eval}Q(Q[j], d(1)) - \text{eval}Q(Q[j], d(2))| \leq 1.
\]

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 \( \text{FORALL-Eq} \), it suffices to prove

\[
\vdash \text{AboveT} \sim_{(\epsilon, 0)} \text{AboveT} : \Phi \rightarrow (r(1) = i) \Rightarrow (r(2) = i).
\]

for every \( i \in \mathbb{Z} \).

We begin with the three initializations:

\[
\begin{align*}
& j \leftarrow 1; \\
& r \leftarrow |Q| + 1; \\
& T \leftarrow \text{Lap}(t);
\end{align*}
\]

This command \( \text{c0} \) computes a noisy version of the threshold \( t \). We use the rule \( \text{LapGen} \) with \( \epsilon = \epsilon/2, k = 1 \) and \( k' = k \), noticing that \( t \) is the same value in both sides. This proves the judgment

\[
\vdash \text{c0} \sim_{\epsilon/2} \text{c0} : \Phi \rightarrow T(1) + 1 = T(2).
\]

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 \( \text{c1} \):

\[
\begin{align*}
& \text{while } j < |Q| \text{ do} \\
& \quad S \leftarrow \text{Lap}_{\epsilon/4}(\text{eval}Q(Q[j], d)); \\
& \quad \text{if } (T \leq S \land r = |Q| + 1) \text{ then } r \leftarrow j; \\
& \quad j \leftarrow j + 1;
\end{align*}
\]

and prove the judgment

\[
\vdash \text{c1} \sim_{\epsilon/2} \text{c1} : \Phi \land T(1) + 1 = T(2) \Rightarrow (r(1) = i) \Rightarrow (r(2) = i)
\]

with the \( \text{WHILE-Ext} \) rule.

References


∀i. ⊢ c_1 \sim_{\epsilon, \delta} c_2 : \Phi \implies x(1) = i \Rightarrow x(2) = i \quad \sum_{i \in I} \delta_i \leq \delta \quad \text{[FORALL-Eq]}

\vdash y_1 \triangleq \mathcal{L}_x(e_1) \sim_{(\epsilon', \delta')} y_2 \triangleq \mathcal{L}_x(e_2) : |k + e_1(1) - e_2(2)| \leq k' \implies y_1(1) + k = y_2(2) \quad \text{[LAPGEN]}

\vdash y_1 \notin \text{FV}(e_1) \quad y_2 \notin \text{FV}(e_2) \quad y_1(1) - y_2(2) = e_1(1) - e_2(2) \quad \text{[LAPNULL]}

Figure 1. Selected proof rules from apRHL^+