Approximate and Probabilistic Differential Privacy Definitions

Sebastian Meiser
University College London, United Kingdom, e-mail: s.meiser@ucl.ac.uk

March 21, 2018

Abstract

This technical report discusses three subtleties related to the widely used notion of differential privacy (DP). First, we discuss how the choice of a distinguisher influences the privacy notion and why we should always have a distinguisher if we consider approximate DP. Secondly, we draw a line between the very intuitive probabilistic differential privacy (with probability $1 - \delta$ we have ε-DP) and the commonly used approximate differential privacy ((ε, δ)-DP). Finally we see that and why probabilistic differential privacy (and similar notions) are not complete under post-processing, which has significant implications for notions used in the literature.

Differential privacy

Differential privacy and its relaxation, approximate differential privacy (ADP) have been used widely in the literature. ADP classically quantifies, in terms of parameters ε and δ, the privacy of a mechanism that releases statistical information on databases with sensitive information in a privacy-preserving way. A prime example is releasing the answers to statistical queries on a database with medical records without revealing information about the individual records in the database. Moreover, differential privacy finds applications in machine learning, smart metering and even anonymous communication. Technically, differential privacy is often defined roughly like this:

Definition 1 (Differential privacy)**.** *A mechanism M is (ε, δ)-differentially private, where $\varepsilon \geq 0$ and $\delta \geq 0$, if for all neighboring databases D_0 and D_1, i.e., for databases differing in only one record, and for all sets $S \subseteq [M]$, where $[M]$ is the range of M, the following in-equation holds:*

$$\Pr[M(D_0) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D_1) \in S] + \delta.$$

We say that a mechanism satisfies *pure differential privacy*, if it satisfies $(\varepsilon, 0)$-differential privacy and we say that it satisfies *approximate differential privacy* (ADP), if it satisfies (ε, δ)-differential privacy for $\delta > 0$.

Differential privacy is quite helpful for defining privacy, particularly since it is preserved under arbitrary post-processing: no matter which functions are applied on the output of a differentially private mechanism, the results do not reduce the privacy guarantee in any way. More formally, if M is (ε, δ)-differentially private and T is any randomized algorithm, then $T(M)$, defined as $T(M)(x) = T(M(x))$ is also (ε, δ)-differentially private.

Background and further reading

For background on the notions, definitions and applications of differential privacy we want to refer to the comprehensive book on differential privacy by Aaron Roth and Cynthia Dwork [2]. As a further read, particularly on the intuition behind the notions, we would recommend some of the great blog posts by Frank McSherry on what differential privacy is and isn't about [3] and his illustrative take on the qualitative difference of ε and δ in (ε, δ)-differential privacy [4].

At this point we don't want to delve into more intuition on why differential privacy is interesting or on the mathematical foundations behind the definition. Instead, we will simply claim that in a perfect world we would typically prefer to have pure $(\varepsilon, 0)$-differential privacy, as this would provide us with a clean and (more or less) easily understandable definition that provides us with deniability. In fact, if would make definitions of differential privacy a bit easier and a common pitfall could be avoided.

Subtlety 1: Who distinguishes the output?

The definition for differential privacy from above holds for all possible sets of outputs of M. Another way to look at this is to consider S to be a (potentially computationally unbounded) distinguisher: a certain subsets of outputs are considered to be in S, while all other outputs are not. Note that if the range of M is large, this distinguisher can potentially solve computationally difficult problems (S can be the set of all outputs that encode instances of DDH), or even solve undecidable problems (S can be the set of all outputs that encode a Turing machine and an input, such that the machine halts on that input). We might not always want to consider such a strong adversary, so in some cases we need to find weaker definitions that allow us to get rid of the all-powerful set S.

To soften up the definition, we can introduce a distinguisher/adversary $\mathcal{A}$ that has to distinguish outputs of M on input D_0 and outputs of M on input D_1, which is very close to the notion of IND-CDP introduced by Mironov, Pandey, Reingold, and Vadhan [5]:

Definition 2 (Computational differential privacy)**.** *A mechanism M is (ε, δ)-differentially private, where $\varepsilon \geq 0$ and $\delta \geq 0$, if for all neighboring databases D_0 and D_1, i.e., for databases differing in only one record, and for all probabilistic polynomial time Turing machines $\mathcal{A}$, the following in-equation holds:*

$$\Pr[0 = \mathcal{A}(M(D_0))] \leq e^{\varepsilon} \Pr[0 = \mathcal{A}(M(D_1))] + \delta.$$

Such a computational definition allows us to include, say, the encryption of a secret into the output of M, without unreasonably concluding that the encryption results in a privacy violation. If we had used sets S instead, our definition would have to hold even for sets that can distinguish the encryption of one secret from the encryption of another secret. Note that we additionally need to be careful about where the encryption key comes from. If the key is considered static, it can be hard-coded into any distinguisher and if it is a secret input into M we need to be careful about who chooses this input. Thus, the next subtlety we run into is deciding on who chooses the input databases. We might ask ourselves: is a property that holds for all pairs or neighboring databases equivalent to having the adversary choose two databases? Fortunately, for stateless mechanisms M and fixed inputs it is, as any pair of two databases (D_0, D_1) can be hard-coded in the description of $\mathcal{A}$. If the databases depend on the security parameter, things get a little more tricky. We either consider a non-uniform adversary (that can have two relevant databases of a polynomial size hard-coded) or we show that our mechanism is not as complex as it could be; e.g., if we simply add some noise to the result of a counting query and additionally send an encryption with it, there really is no need for complicated solutions: we can split the analysis of the (ε, δ)-differentially private output and of the encryption (leading to a negligible adversarial advantage δ') and then deduce that we achieve $(\varepsilon, \delta + \delta')$-differential privacy.

Bad idea: no distinguisher at all An alternative path of getting rid of the set S is simply avoiding it, which brings us to our first common example of an unhelpful definition.

Definition 3 (How <u>not</u> to define differential privacy)**.** *A mechanism M is (ε, δ)-differentially private, where $\varepsilon \geq 0$ and $\delta \geq 0$, if for all neighboring databases D_0 and D_1, i.e., for databases differing in only one record, and for all potential outputs $y \in [M]$, where $[M]$ is the range of M, the following in-equation holds:*

$$\Pr[y = M(D_0)] \leq e^{\varepsilon} \Pr[y = M(D_1)] + \delta.$$

The two definitions Definitions 1 and 3 are equivalent if $\delta = 0$, so stumbling upon a definition on individual elements in not always problematic. However, if δ happens to be non-zero and particularly if M has a rather large range, things can become tricky. Let's consider what happens if $|[M]| > \frac{1}{\delta}$: This can occur even for a value of δ negligible in a security parameter λ, if the mechanism is, e.g., allowed to output strings linear in λ. Imagine the following mechanism M_{leaky}. On input D, the mechanism outputs whatever private information we don't want it to output and attaches a random bit string of length λ. Since each such random bit string has a chance of $\frac{1}{2^{\lambda}}$ of being chosen, the probability for any $y \in [M_{\text{leaky}}]$ to be the output of M_{leaky} is bounded by $\frac{1}{2^{\lambda}}$ as well. Consequently, we have a negligible $\delta(\lambda) = \frac{1}{2^{\lambda}}$ and for any two inputs D_0 and D_1 (including neighboring ones), we have

$$\Pr[y = M_{\text{leaky}}(D_0)] \leq \delta(\lambda) \leq e^{\varepsilon} \cdot \Pr[y = M_{\text{leaky}}(D_1)] + \delta(\lambda).$$

In summary, we should not define ADP without any distinguisher, even if δ is negligible.

Subtlety 2: Probabilistic differential privacy is not equivalent to approximate differential privacy

ADP is widely used, and rightfully so, since many actual implementations of differential privacy do not satisfy ε-differential privacy, but only (ε, δ)-ADP. Here is a common interpretation for what it means if a mechanism is (ε, δ)-ADP: with probability $1 - \delta$ the output of the mechanism enjoys (pure) $(\varepsilon, 0)$-differential privacy, i.e., the ratio between the probability to observe this output on either input is bounded by e^{ε}. This intuition is useful for reasoning about the otherwise quite mathematical notion in terms we can mostly understand: Assuming we have developed an understanding of the probability ratio of (pure) differential privacy, we can easily reason about small chances of things going wrong. However, this intuition unfortunately is wrong: there can be events that occur with a probability significantly higher than δ for which the probability ratio is not quite bounded by e^{ε}. To illustrate this discrepancy between ADP and probabilistic DP, we create a simple example.

A simple example We construct a mechanism M on just one bit of input and one bit of output. Let

$$M(0) = \begin{cases} 0 & \text{with probability } \frac{e^{\varepsilon}}{1+e^{\varepsilon}} \\ 1 & \text{with probability } \frac{1}{1+e^{\varepsilon}} \end{cases},$$

for some small value $\varepsilon > 0$ and, conversely,

$$M(1) = \begin{cases} 0 & \text{with probability } \frac{1}{1+e^{\varepsilon}} \\ 1 & \text{with probability } \frac{e^{\varepsilon}}{1+e^{\varepsilon}} \end{cases}.$$

We trivially see that we have (pure) ε-DP:

$$\forall x. \quad e^{-\varepsilon} \leq \frac{M(0)}{M(1)} = \frac{M(1)}{M(0)} \leq e^{\varepsilon}.$$

Since this ratio holds for all individual elements (all two of them), it also holds for all sets of elements (all four of them). Moreover, we can calculate the total variation distance (the L^1 norm) between $M(0)$ and $M(1)$ as $\frac{e^{\varepsilon}-1}{1+e^{\varepsilon}}$. In terms of distinguishing between them, the best distinguisher simply outputs the bit it sees. Thus, we can show that M is $\left(0, \frac{e^{\varepsilon}-1}{1+e^{\varepsilon}}\right)$-ADP as well (here, the worst-case sets are $S = \{0\}$ and $S = \{1\}$). The latter should already tell us that (ε, δ)-ADP cannot be the same as probabilistic differential privacy: we have a result for $\varepsilon = 0$ with a small value $\delta = \frac{e^{\varepsilon}-1}{1+e^{\varepsilon}} \approx \frac{\varepsilon}{2}$, but in no way are we guaranteed to have 0-DP with a probability anywhere near $1 - \frac{\varepsilon}{2}$. Rather, more likely than not (with probability $\frac{e^{\varepsilon}}{e^{\varepsilon}+1} > \frac{1}{2}$) we will observe an event for which the ratio is exactly e^{ε}, which means that an observer learns exactly this: the chance that the input was b is, assuming an unbiased prior, more likely by a factor of e^{ε} than that the input was $1 - b$.

A more general note We can apply a very similar technique counter to our intuition of probabilistic differential privacy: for any M that is $(\varepsilon, 0)$-DP and for any $0 \leq \varepsilon' \leq \varepsilon$, we can easily show that M satisfies $(\varepsilon', e^{\varepsilon} - e^{\varepsilon'})$-ADP. That is, for all $S \subseteq [M]$ and for all neighboring D_0, D_1 we have

$$\begin{aligned} \Pr[M(D_0) \in S] &\leq e^{\varepsilon} \Pr[M(D_1) \in S] \\ &= e^{\varepsilon'} \Pr[M(D_1) \in S] + (e^{\varepsilon} - e^{\varepsilon'}) \Pr[M(D_1) \in S] \\ &\leq e^{\varepsilon'} \Pr[M(D_1) \in S] + (e^{\varepsilon} - e^{\varepsilon'}). \end{aligned}$$

As this calculation in no way requires any information about M or the two inputs D_0 and D_1, it in fact is always possible and leads us to what we can call an epsilon-delta-graph: For every chosen value of ε', we can derive a bound on δ so that the distributions are (ε', δ)-ADP. When doing just that, we see that δ in fact is unrelated to the actual probabilities of any real events and is merely a definitional measure for how much of a probability mass is outside the ε' curve, by which we mean the area between the output distributions of $M(D_0)$ and $e^{\varepsilon} \cdot M(D_1)$, which is the output distribution of $M(D_1)$ multiplied with e^{ε} (see Figures 1 and 2).

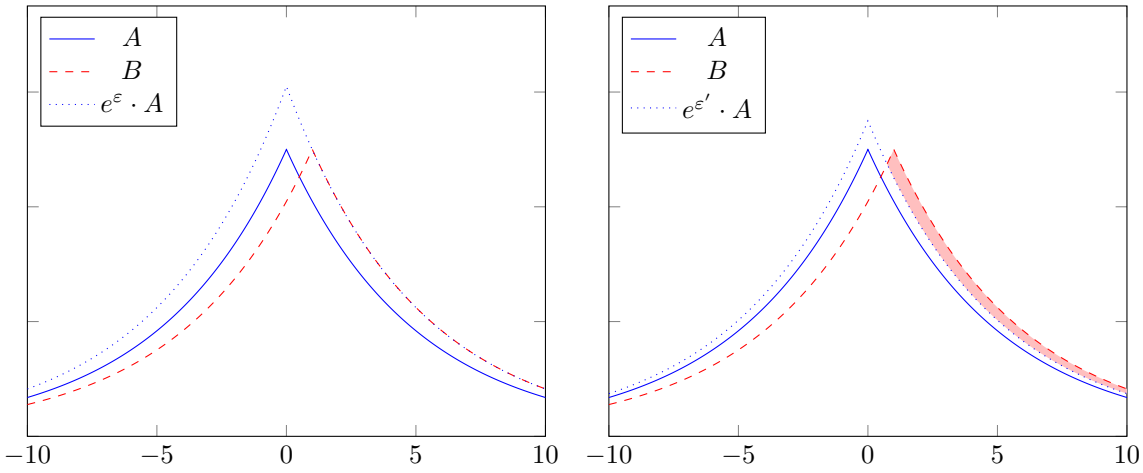

Figure 1: Each graph depicts two Laplace probability distributions A and B, as well as a "scaled up" version of A (every point is multiplied with e^{ε} / $e^{\varepsilon'}$). Note that $e^{\varepsilon} \cdot A$ is not a probability distribution. On the left, we scale up A with an exactly large enough factor e^{ε} to achieve pure ε-DP (every point of B fits under $e^{\varepsilon} \cdot A$). On the right side we chose a factor $e^{\varepsilon'}$ that is too low (not every point of B fits under $e^{\varepsilon'} \cdot A$). We still achieve (ε', δ)-ADP, where δ is the volume of the red area, i.e., the area not captured under the scaled up curve.

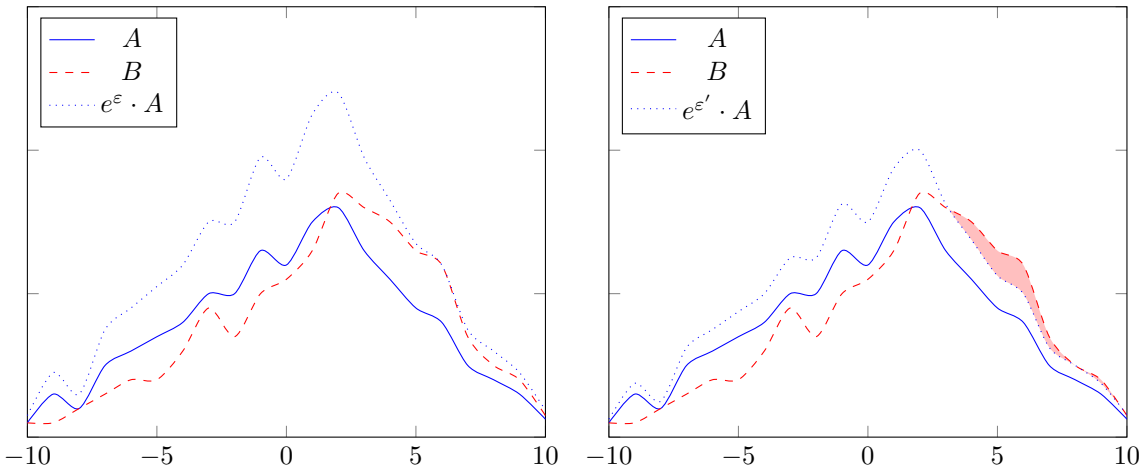

Figure 2: Each graph depicts two less regular probability distributions A and B, as well as a "scaled up" version of A (every point is multiplied with e^{ε} / $e^{\varepsilon'}$). Note that $e^{\varepsilon} \cdot A$ is not a probability distribution. On the left, we scale up A with an exactly large enough factor e^{ε} to achieve pure ε-DP (every point of B fits under $e^{\varepsilon} \cdot A$). On the right side we chose a factor $e^{\varepsilon'}$ that is too low (not every point of B fits under $e^{\varepsilon'} \cdot A$). We still achieve (ε', δ)-ADP, where δ is the volume of the red area, i.e., the area not captured under the scaled up curve.

Subtlety 3: Probabilistic differential privacy is not preserved under post-processing

One might ask why we even bother to work with ADP in the first place, if there is a more intuitive form of differential privacy. Consider the following definition for probabilistic differential privacy:

Definition 4. *A mechanism M is (ε, δ)-probabilistically differentially private, where $\varepsilon \geq 0$ and $\delta \geq 0$, if for all neighboring databases D_0 and D_1, i.e., for databases differing in only one record there are sets $S_0^\delta, S_1^\delta \subseteq [M]$ with $\Pr\left[M(D_0) \in S_0^\delta\right] \leq \delta$ and $\Pr\left[M(D_1) \in S_1^\delta\right] \leq \delta$, s.t., for all sets $S \subseteq [M]$, where $[M]$ is the range of M, the following in-equations hold:*

$$\begin{aligned}\Pr\left[M(D_0) \in S \setminus S_0^\delta\right] &\leq e^\varepsilon \cdot \Pr\left[M(D_1) \in S \setminus S_0^\delta\right] \\ \wedge \Pr\left[M(D_1) \in S \setminus S_1^\delta\right] &\leq e^\varepsilon \cdot \Pr\left[M(D_1) \in S \setminus S_1^\delta\right].\end{aligned}$$

The definition fits to our intuition in that there may be some events that occur with a probability of less than δ for which we don't make a statement, but all other events enjoy the property of (pure) $(\varepsilon, 0)$-DP. One of the main advantages of differential privacy, however, is that the property is preserved under arbitrary post-processing. Indeed, as long as our post-processing only expands each output/event this is still true for PDP. However, we can define a post-processing step that destroys the PDP property.

Theorem 1. *Let M be any (ε, δ)-PDP mechanism and let D_0 and D_1 be two neighboring inputs for which the PDP property barely holds, i.e., for which we have sets $S_0^\delta \subseteq [M]$ and $S_\varepsilon \subseteq [M] \setminus S_0^\delta$ with the following properties (that are in line with PDP):*

- $\Pr\left[M(D_0) \in S_0^\delta\right] \leq \delta$,
- $\Pr\left[M(D_0) \in S_0^\delta \cup S_\varepsilon\right] > e^\varepsilon \Pr\left[M(D_1) \in S_0^\delta \cup S_\varepsilon\right]$,
- $\Pr\left[M(D_0) \in S_0^\delta \cup S_\varepsilon\right] > \delta$.

Then, there is a deterministic transformation T such that $T \circ M$ is not PDP.

Proof. Let M be any (ε, δ)-PDP mechanism and let D_0 and D_1 be two neighboring inputs for which we have sets $S_0^\delta, S_\varepsilon \subseteq [M]$ with the properties mentioned in the lemma.

Let T be the following transformation and let $\perp$ be a value that is not in $[M]$:

$$T(x) = \begin{cases} \perp & \text{if } x \in S_0^\delta \cup S_\varepsilon \\ x & \text{otherwise.} \end{cases}$$

Note that $\Pr\left[T(M(D_0)) = \perp\right] = \Pr\left[M(D_0) \in S_0^\delta \cup S_\varepsilon\right] > \delta$. Consequently, one cannot add the event $\perp$ to a δ-set for $T \circ M$: For every set S_T^δ with $\Pr\left[T(M(D_0)) \in S_T^\delta\right] \leq \delta$ we know that $\perp \notin S_T^\delta$. To satisfy the definition of PDP, we thus need the $(\varepsilon, 0)$-DP inequality to hold for this event $\perp$, but it will not. Formally, since $S_\varepsilon \subseteq [M] \setminus S_0^\delta$, we have for $S_\perp = \{\perp\}$

$$\begin{aligned}\Pr\left[T(M(D_0)) \in S_\perp\right] &= \Pr\left[M(D_0) \in S_0^\delta \cup S_\varepsilon\right] \\ &> e^\varepsilon \cdot \Pr\left[M(D_1) \in S_0^\delta \cup S_\varepsilon\right] \\ &= e^\varepsilon \Pr\left[T(M(D_1)) \in S_\perp\right],\end{aligned}$$

which violates the definition of (ε, δ)-PDP. Thus, $T \circ M$ does not (ε, δ)-PDP. □

Consider, as an example, the following distributions:

$$M(0) = \begin{cases} 0 & \text{with probability } \delta \\ 1 & \text{with probability } \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 2 & \text{with probability } \frac{1}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 3 & \text{with probability } 0 \end{cases} \qquad M(1) = \begin{cases} 0 & \text{with probability } 0 \\ 1 & \text{with probability } \frac{1}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 2 & \text{with probability } \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 3 & \text{with probability } \delta \end{cases}.$$

They clearly satisfy (ε, δ)-PDP, but $T(M)$ will look like this:

$$M(0) = \begin{cases} \perp & \text{with probability } \delta + \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 2 & \text{with probability } \frac{1}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 3 & \text{with probability } 0 \end{cases} \qquad M(1) = \begin{cases} \perp & \text{with probability } \frac{1}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 2 & \text{with probability } \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) \\ 3 & \text{with probability } \delta \end{cases},$$

and we have $\Pr[M(0) = \perp] = \delta + \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) > \frac{e^\varepsilon}{(1+e^\varepsilon)} \cdot (1-\delta) = e^\varepsilon \cdot \Pr[M(1) = \perp]$.

The very same strategy can be applied to other "probabilistic" relaxations of (differential) privacy notions, such as δ-approximate zero-concentrated differential privacy and to a wide range of approximate probabilistic properties, really. To see that this is true, we generalize the proof idea from above.

Generalization: no non-trivial δ-approximate probabilistic property is preserved under post-processing. While there of course are properties that are preserved under post-processing (for example $(\varepsilon, 0)$-DP and (ε, δ)-ADP), we now characterize which properties are not preserved. They boil down to properties that exclude a "bad" event (defined as a set of potential outputs; if this event occurs, the property does not have to hold), by placing a bound δ on the probability that the "bad" event occurs and by forcing the property to hold otherwise. Moreover, the property can't just be trivially true (e.g., even if the "bad" event occurs), so we require that there is some set of "borderline" events that, when combined with the "bad" events do not fulfill the property. To disallow the property from cheating, we finally require that the combined probability of the "bad" and the "borderline" events must be larger than δ.

Definition 5 (non-trivial δ-approximate probabilistic property)**.** *We call any property Υ a δ-approximate non-trivial probabilistic property on two distributions X and Y if there are sets $S_{bad} \subseteq [Y], S_{borderline} \subseteq [Y] \setminus S_{bad}$ with the following properties:*

1. $\Pr[Y \in S_{bad}] \leq \delta$,
2. $\forall S \subseteq [Y] \setminus S_{bad} : \Upsilon(Y|_S, Z|_S) = 1$
3. $\Upsilon(Y|_{S_{borderline} \cup S_{bad}}, Z|_{S_{borderline} \cup S_{bad}}) = 0$
4. $\Pr[Y \in S_{borderline} \cup S_{bad}] > \delta$,

where $X|_S$ restricts the distribution X on the elements in the set S. If $X = M(D)$ corresponds to a randomized mechanism on some input D, the restriction applies to the outputs of the mechanism on this input. We say that two distributions are weakly δ*-approximate Υ, if only the properties 1. and 2. are satisfied.*

Theorem 2. *Let Υ any δ-approximate non-trivial property on two distributions Y and Z. Then there is a deterministic transformation T such that $T(Y)$ and $T(Z)$ are not even weakly δ-approximate Υ.*

Proof. Let Υ any δ-approximate non-trivial property on two distributions Y and Z with the sets $S_{\text{bad}} \subseteq [Y]$ and $S_{\text{borderline}} \subseteq [Y] \setminus S_{\text{bad}}$ defined as in Definition 5. Let T be the following transformation, where $\perp$ is value that is not in $[Y]$:

$$T(x) = \begin{cases} \perp & \text{if } x \in S_{\text{bad}} \cup S_{\text{borderline}} \\ x & \text{otherwise.} \end{cases}$$

Assume for contradiction that there was a set T_{bad} s.t.,

- $\Pr[T(Y) \in T_{\text{bad}}] \leq \delta$,
- $\forall S \subseteq [T(Y)] \setminus T_{\text{bad}} : \Upsilon(\Pr[T(Y) \in S], \Pr[T(Z) \in S]) = 1$

Note that since Y and Z are δ-approximate non-trivial probabilistic Υ, by property 4. we have $\Pr[T(Y) = \perp] = \Pr[Y \in S_{\text{bad}} \cup S_{\text{borderline}}] > \delta$. Consequently, $\perp$ cannot be in T_{bad}. We define the set $S_\perp = \{\perp\} \subseteq [T(Y)] \setminus T_{\text{bad}}$ and get

$$\Upsilon(T(Y)|_{S_\perp}, T(Z)|_{S_\perp}) = \Upsilon(Y|_{S_{\text{borderline}} \cup S_{\text{bad}}}, Z|_{S_{\text{borderline}} \cup S_{\text{bad}}}) = 0,$$

which violates property 2. for $T(Y)$ and $T(Z)$. Since $S_\perp \subseteq [T(Y)] \setminus T_{\text{bad}}$, this is a contradiction to our assumption that $T(Y)$ and $T(Z)$ are weakly δ-approximate Υ. □

Example: δ-approximate zero-concentrated differential privacy (δ-zCDP) We now instantiate the general property from above to show that δ-zCDP is not preserved under post-processing. To see this, we first review the definition of δ-zCDP.

Definition 6 (δ-approximate zero-concentrated differential privacy [1])**.** *A randomized mechanism M is δ-approximately (ξ, ρ)-zCDP if, for all neighboring D_0, D_1 there exist events E and E' such that $\Pr[E] \geq 1 - \delta \wedge \Pr[E'] \geq 1 - \delta$, and*

$$\forall \alpha \in (1, \infty) \quad D_\alpha\left(M(D_0)|_E \mid\mid M(D_1)|_{E'}\right)) \leq \xi + \rho \cdot \alpha \quad \wedge \quad D_\alpha\left(M(D_1)|_{E'} \mid\mid M(D_0)|_E\right)) \leq \xi + \rho \cdot \alpha,$$

where for two random variables Y and Z over the same universe U, $D_\alpha(Y||Z)$ is the Rényi divergence

$$D_\alpha(Y||Z) = \frac{1}{\alpha - 1} \log\left(\sum_{x \in U} \frac{Y(x)}{Z(x)}\right),$$

which is not defined if $Z(x) = 0$ for any $x \in U$.

We concentrate on only one direction and set $\Upsilon(Y, Z) = \forall \alpha \in (1, \infty) : \quad D_\alpha\left(Y \mid\mid Z\right)) \leq \xi + \rho$. Assume two distributions Y and Z are non-trivial δ-approximate Υ, i.e., there are events occurring with probability δ for which the property doesn't hold and at least one additional event for which it barely holds. Then, Υ is not preserved if we merge these events by applying T. In particular, if a mechanism M is δ-approximately (ξ, ρ)-zCDP in a non-trivial way, then $T(M)$ is not δ-approximately (ξ, ρ)-zCDP.[1]

The mechanism in the middle The difference between ADP and PDP is a bit subtle. In fact Dwork and Roth [2](Lemma 3.17) show that if and only if M satisfies (ε, δ)-ADP, then we can for any two neighboring inputs D_0 and D_1 find distributions M'_{D_0} and M'_{D_1} such that $\Delta(M(D_0), M'_{D_0}) \leq \frac{\delta}{1+e^\varepsilon}$ and $\Delta(M(D_1), M'_{D_1}) \leq \frac{\delta}{1+e^\varepsilon}$ and such that M'_{D_0} and M'_{D_1} are related as by pure $(\varepsilon, 0)$-DP, where Δ is the statistical distance. Thus, we can say that (ε, δ)-ADP means that M on two inputs is about δ-indistinguishable to some other distributions that have a (now only multiplicative) privacy loss of at most e^ε.

What do we learn?

Pure differential privacy seems like a useful notion that can be defined and even (to some degree) be understood intuitively: ε captures how much more likely any event based on the output of the mechanism can be. These events can range from any guesses about the input based on the output to computations and even actions taken based on the observed output. However, whenever we use computers (that often make use of imperfect randomness) or we combine our outputs with cryptography or we have any other ever so slight chance that the mechanism behaves unexpectedly, we simply cannot expect to get pure $(\varepsilon, 0)$-differential privacy.

Several weaker definitions have been proposed and discussed and many of them seem to work fine (although they run a bit counter to our intuition), but it seems we need to be careful with any δ-approximate (non-trivial) variants of privacy notions. Whenever we try to confine bad events by banishing them into a "small probability of failure" it seems that very simple transformations can, at least formally, invalidate the privacy notion.

Lessons learned in terms of post-processing We see two reasons against using δ-approximate non-trivial privacy properties, such as probabilistic differential privacy or δ-approximate zero-concentrated differential privacy. First, proofs for (differential) privacy notions of any kind very often use preservation under post-processing as a fundamental component. If the preservation under post-processing is not satisfied, we might be faced with problems and errors in proofs. Second, a property that isn't preserved under post-processing may not be helpful in understanding privacy: Nothing prevents the adversary from processing the observations it makes, in which case the property does not hold anymore and we might only end up with a different privacy notion, such as ADP. To make reasonable statements about what the adversary can learn, we then need to understand which privacy notion still holds after post-processing. Thus, we reason that we should use that notion in the first place instead of the original notion.

[1] Note that we require a sort of "tightness" of the bounds to make such strong statements. A mechanism that, for example, **always** outputs 0 is of course δ-approximately (ξ, ρ)-zCDP for arbitrary $\delta, \xi, \rho \geq 0$ and post-processing will not change that.

Acknowledgement

This work has been partially supported by the European Commission through H2020-DS-2014-653497 PANORAMIX and the EPSRC Grant EP/M013-286/1. Thanks to Esfandiar Mohammadi for the discussions on approximate differential privacy and for the idea behind the graphical depiction of delta.

References

[1] M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

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

[3] F. McSherry and J. Klucar. Lunchtime for data privacy. Blog, available under `https://github.com/frankmcsherry/blog/blob/master/posts/2016-08-16.md`, 2016.

[4] F. McSherry and J. Klucar. How many secrets do you have? Blog, available under `https://github.com/frankmcsherry/blog/blob/master/posts/2017-02-08.md`, 2017.

[5] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In *Advances in Cryptology-CRYPTO 2009*, pages 126–142. Springer, 2009.