Basic Definitions of Differential Privacy

Gonzalo Munilla Garrido

Attacks on privacy are an ever-present threat. Traditional privacy protection methods are not sufficient to prevent all risks involved in publishing an analysis of sensitive data because they are usually vulnerable to auxiliary information attacks [12] [9]. To address the limitations of traditional techniques against such attacks, Differential privacy (DP) was introduced in 2006 by Microsoft Research associate Cynthia Dwork [5].

In broad terms, DP guarantees personal protection from any risks caused by being part of a database, irrespective of how much auxiliary information is available. More specifically, DP is a mathematical guarantee that can be satisfied by an algorithm that releases statistical information about a dataset. C. Dwork describes it as "*[...] a promise, made by a data curator to a data subject: You will not be affected, adversely or otherwise, by allowing a practitioner to use one's data in any study, no matter what other studies, data sets, or information from other sources is available.*" [3]. DP maintains this promise by ensuring that the practitioner reaches the same conclusion, i.e., any analysis output is "essentially" equally likely to occur, regardless of the presence or absence of any individual in the dataset. Consequently, the release of information does not reveal anything new about an individual because an adversary analyzing the output cannot tell whether this individual's data was used to compute the output, regardless of any auxiliary information the attacker possesses.
For the adversary, there exists two datasets, one with the victim $\mathcal{D}$, and one without $\mathcal{D}'$, or vice versa. These are called neighboring datasets. One can conclude that every individual in a dataset has "essentially" the same protection level they would have had if they were not in the dataset. Symmetrically, individuals not included in a dataset have "essentially" the same protection as if they were. Put into yet another perspective, the output of DP algorithms is "essentially" the same, including or not including an individual. The term "essentially" is captured by the parameter ε. ε bounds how much more likely finding an individual in a scenario is than in the other. This concept is formally described in Definition 1, which is based on [3].

Definition 1 (*ε-Differential Privacy*). A randomized algorithm $\mathcal{M}$ is ε-differentially private if for any two datasets $\mathcal{D}$ and $\mathcal{D}'$ differing on at most one element (neighboring datasets), and any set of possible outputs $\mathcal{S} \in Range(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\varepsilon} \times \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}].$$

Definition 1 is considered pure DP. A weaker form of DP is (ε,δ)-differential privacy, which is described in Definition 2, based on [2]. This relaxed form of DP introduces a new parameter, δ, which lowers the individuals' probable privacy in exchange for accuracy. [8]. In pure DP (Definition 1), each output is "essentially" equally likely. However, introducing δ, introduces the minimal possibility that some outputs are much more or much less likely depending on whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$, i.e., the output is bounded by ε with a probability of at least $1-\delta$.

Definition 2 (*(ε,δ)-Differential Privacy*). A randomized algorithm $\mathcal{M}$ is (ε,δ)-differentially private if for any two datasets $\mathcal{D}$ and $\mathcal{D}'$ differing on at most one element, and any set of possible outputs $\mathcal{S} \in Range(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\varepsilon} \times \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta.$$

Additionally, one may choose how $\mathcal{D}$ and $\mathcal{D}'$ differ in one individual, i.e., what is the definition of a neighboring dataset; there are two possibilities: Bounded or unbounded DP. In practice, both definitions are correct; however, there might be some algorithm restrictions if one chooses bounded DP, and, additionally, the output results may vary depending on which definition one chooses [**?**]. One can achieve unbounded DP by either removing ($|\mathcal{D}'| = |\mathcal{D}| - 1$), or adding ($|\mathcal{D}'| = |\mathcal{D}| + 1$) one record. On the other hand, bounded DP is achieved when both neighbors contain the same number of individuals ($|\mathcal{D}'| = |\mathcal{D}|$). Nonetheless, formally, in bounded and unbounded DP, $\mathcal{D}$ and $\mathcal{D}'$ hold a Hamming distance of $d_h(\mathcal{D},\mathcal{D}') = 1$, i.e. differ in one individual; if the Hamming distance is higher, then DP ensures group privacy [?].

Definition 3 (*Unbounded DP*). Given two datasets $\mathcal{D}$ and $\mathcal{D}'$, $\mathcal{D}'$ can be obtained by removing or adding one record from $\mathcal{D}$, i.e. while the Hamming distance between datasets is $d_h(\mathcal{D},\mathcal{D}') = 1$, their cardinality is different: $|\mathcal{D}'| = |\mathcal{D}| - 1$ (Removing one record), or $|\mathcal{D}'| = |\mathcal{D}| + 1$ (Adding one record).

Definition 4 (*Bounded DP*). Given two datasets $\mathcal{D}$ and $\mathcal{D}'$, $\mathcal{D}'$ can be obtained by changing one record from $\mathcal{D}$, i.e. while the Hamming distance between datasets is $d_h(\mathcal{D},\mathcal{D}') = 1$, their cardinality is the same: $|\mathcal{D}'| = |\mathcal{D}|$.

Definitions 1 to 4 lay the theory, but to comply with them, a randomized algorithm $\mathcal{M}$ needs to add noise sampled from a random variable to the true output of an algorithm $\mathcal{W}$:

$$\mathcal{M}(\mathcal{D}) = \mathcal{W}(\mathcal{D}) + Noise.$$

However, the standard deviation of the probability density function (pdf) of the random variable needs to be proportional to the output difference between the truthful algorithm $\mathcal{W}$ executed on any possible $\mathcal{D}$ and on any possible $\mathcal{D}'$. Additionally, $\mathcal{D}'$ must not contain the individual from $\mathcal{D}$ that generates the largest difference in output possible so that any other individual is protected. Formally, this maximum difference is the algorithm's ℓ_1-sensitivity, also known as *global sensitivity*, which is defined in Definition 2 based on [8], without using their histogram notation.

Definition 5 (ℓ_1-sensitivity). The ℓ_1-sensitivity of an algorithm $\mathcal{W} : \mathbb{R}^{k'} \to \mathbb{R}^{k''}$, executed over datasets $\mathcal{D}$, $\mathcal{D}' \in \mathbb{R}^k$ at a Hamming distance of $d_h(\mathcal{D},\mathcal{D}') = 1$, i.e. differing in at most one record, is defined to be:

$$\Delta f_{GS} = \max_{\substack{\mathcal{D},\mathcal{D}' \in \mathbb{R}^k \\ d_h(\mathcal{D},\mathcal{D}')=1}} \|\mathcal{W}(\mathcal{D}) - \mathcal{W}(\mathcal{D}')\|_1.$$

The value of the ℓ_1-sensitivity depends on whether DP is defined to be bounded or unbounded, e.g., the sensitivity of an unbounded DP count is 1, while of a bounded DP count is 2. Removing or adding one value reduces or increases the count of an attribute by 1. On the other hand, changing an attribute may increase the count of an attribute by 1, and decrease the count of another by 1, generating a total difference in counts of 2.

Furthermore, if instead of defining the ℓ_1-sensitivity with any possible datasets, we fix $\mathcal{D}$ as the dataset queried, then, the definition describes *local sensitivity* instead. The *Local sensitivity* of an algorithm is smaller in value than its *global sensitivity* (Upper bound), as it does not consider all the possible values the neighboring datasets could have. *Local sensitivity* ensures privacy while adding less noise to the algorithm's output; however, because *local sensitivity* depends on the algorithm's input $\mathcal{D}$, the output may reveal information from the input. *Local sensitivity* is described in Definition 6.

$$\Delta f_{LS} = \max_{\substack{\mathcal{D}' \in \mathbb{R}^k \\ d_h(\mathcal{D},\mathcal{D}')=1}} \|\mathcal{W}(\mathcal{D}) - \mathcal{W}(\mathcal{D}')\|_1 \leq \Delta f_{GS}.$$

Moreover, there are multiple implementations of algorithms complying with Definition 1 and 2, e.g., the Gaussian mechanism [8], or the exponential mechanism [11]. The simplest and the one devised since the inception of DP is the Laplace mechanism, which is defined in Definition 7 based on [4]. The definitions and proofs for the exponential and Gaussian mechanism may be found in Chapter 3, and the Appendix of [8], respectively.

Definition 7 (Laplace mechanism). For an algorithm $\mathcal{W}$ executed over a dataset $\mathcal{D}$, the differentially private version $\mathcal{M}$, adds Laplace noise proportional to the sensitivity of $\mathcal{W}$ and inversely proportional to ε:

$$\mathcal{M}(\mathcal{D}) = \mathcal{W}(\mathcal{D}) + Lap(\Delta f/\varepsilon),$$

with

$$Lap(\mathcal{W}(\mathcal{D}) = x \mid \mu = 0, b = \frac{\Delta f}{\varepsilon}) = \frac{\varepsilon}{2\Delta f} e^{-\dfrac{\varepsilon|x|}{\Delta f}}.$$

These expressions mean that the lower the ε, the larger the noise, and thus, the more privacy we ensure. Moreover, note that the magnitude of the ℓ_1-sensitivity depends on the type of algorithm. The larger the Δf, the more noise algorithm $\mathcal{W}$ needs to be private, e.g., a count query needs less noise than a mean query. Practitioners may calculate Δf empirically, but for large datasets, it is practically infeasible. Thus, there are empirical estimations and analytical formulas. Furthermore, as it has been explained, using *local sensitivity* rather than *global sensitivity* reduces the variance of the noise sample, but at the expense of revealing information about the input. Lastly, note that the noise is independent of the dataset's size, and, therefore, the larger a dataset is, the more relatively accurate the results are.

Combining Definition 1 with the pdf of Laplace:

$$\frac{Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}]}{Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}]} = \frac{Lap(x|\Delta f/\varepsilon)}{Lap(x+\Delta f|\Delta f/\varepsilon)} \leq e^{\varepsilon}.$$

The complete proof may be found at the end of this document, which is based on [8].

Furthermore, DP processes have the property to be cumulative, i.e. if two randomized algorithms $\mathcal{M}_1$ and $\mathcal{M}_2$, add noise to the resulting query over the same dataset $\mathcal{D}$ with ε_1 and ε_2, the total ε used is $\varepsilon_1 + \varepsilon_2$; this is called *sequential composition* [8]. The DP community has coined the final ε as the *privacy budget*, and practitioners set the privacy budget before executing a set of randomized algorithms. On the other hand, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are applied to disjoint datasets, i.e., not containing the same underlying data, then the privacy budget of the analysis results is $max(\varepsilon_1, \varepsilon_2)$; this has the name of *parallel composition*. These two types of composition play a significant role in designing privacy budget strategies.

Privacy budgeting is tightly related to the concept of user-level [7] and event-level privacy [6] [10]. The original intention behind DP was to provide user-level privacy [7], where all the records belonging to a single individual are either present or absent. However, in streaming data, an individual's budget of, for example, 1 or even 100 would quickly render the query results meaningless given the sheer amount of data points an individual generates daily. There is not a clear consensus in literature of what should be the right amount of privacy budget; thus, some authors have defined event-level privacy [6] [10], where all the records belonging to an event or group of events are either missing or absent.

An essential aspect of DP that contributes to privacy budgeting is its quality of being post-processing proof [8]. Once a noisy result has been disclosed, any operation executed after the fact cannot revert the DP process. To reveal the underlying data, adversaries instead can only update their posterior beliefs about the possible instances of such data, which depending on their size and nature, the resources needed are inordinate.

Another advantage of DP is its versatility. To provide an example, the best DP version of a deep neural network uses DP stochastic gradient descent (DP-SGD), proposed by M. Abadi et al. [1]. DP-SGD (i) randomly samples a lot from the examples, (ii) computes the gradient of each example, (iii) clips the ℓ_2 norm of the gradients to prevent outliers from skewing the gradients, (iv) computes the mean of the gradient adding noise form a Gaussian random variable, (v) takes a step in the opposite direction of the mean noisy gradient, and (vi) computes the privacy loss for the entire process across layers. DP-SGD is a testimony of the versatility but a complex adaption of DP to existing processes. Other machine learning algorithms have been transformed into DP compliant, e.g., K-means, PCA, linear regression, among others. Moreover, even collecting data from cell phones can be converted into a DP compliant process.

Lastly, in terms of adaptability, DP can be deployed locally or globally. One may use DP on single data points, e.g., extracting information from someone's cell phone by adding noise with a randomized response. Alternatively, a trusted data curator may collect data from all available cell phones and apply queries with Laplacian noise to the combined dataset. Local DP is more private but less accurate, while global DP is more accurate but less private as there needs to be an intermediary. There could be a sweet spot in the middle, but the DP community should further research the matter.

In summary, DP provides a strong, mathematically proven guarantee that the achieved privacy holds under any circumstances transparently, as it does not need to conceal details of the implementation secret because of its post-processing quality. Moreover, DP adapts to many use cases, provides a factor ε to assess privacy loss and risk, and can make sensitive data widely available for researchers and practitioners worldwide, e.g., use cancer-related data to improve diagnosis and cures. However, despite the excellent properties of DP, it also brings challenges: (i) there are a limited utility, (ii) its implementation is complex and usually tailored per use case, and (iii) practitioners and product owners must change the way they work to consider DP.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, Oct. 2016. arXiv: 1607.00133.

[2] F. M. I. M. C. Dwork, K. Kenthapadi and M. Naor. Our data, our- selves: Privacy via distributed noise generation. *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2006.

[3] C. Dwork. Differential privacy. *International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[4] C. Dwork. Differential Privacy: A Survey of Results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation*, volume 4978, pages 1–19. Springer Berlin Heidelberg,

Berlin, Heidelberg, 2008. Series Title: Lecture Notes in Computer Science.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. page 20.

[6] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, page 715, Cambridge, Massachusetts, USA, 2010. ACM Press.

[7] C. Dwork, M. Naor, T. Pitassi, S. Yekhanin, and G. N. Rothblum. Pan-Private Streaming Algorithms. page 32.

[8] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[9] X. Gao, B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang, and J. Lindqvist. Elastic pathing: your speed is enough to track you. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 975–986, Seattle Washington, Sept. 2014. ACM.

[10] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment*, 7(12):1155–1166, Aug. 2014.

[11] F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. page 10.

[12] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, Oakland, CA, USA, May 2008. IEEE. ISSN: 1081-6011.