guardtime.com

Privacy Series: Differential Privacy in a Nutshell

by Guardtime Research

Whitepaper
November 2019

+7181412527

+3BF25278

+121412527

Introduction

Data privacy has become a prominent issue with companies ingesting and sharing troves of sensitive user data. There are several privacy-preserving technologies that range from the simpler data anonymization techniques like pseudonymization, data masking, data shuffling, etc to complex cryptographic techniques such as multi-party computation (MPC), homomorphic encryption (HE), etc.

It's well known that data anonymization does not necessarily prevent privacy leaks if/when auxiliary datasets become available in the future. One of the most widely known examples of such de-anonymization attacks was introduced in [3] where the Netflix Prize dataset containing movie ratings of 500,000 subscribers was augmented with data from IMDB as background knowledge, thereby allowing the identification of individual Netflix subscribers and uncovering their potentially sensitive information (e.g. political preferences, etc).

While technologies like secure multi-party computation and homomorphic encryption offer high computational security, they do not guarantee output privacy i.e. whether results from the secure MPC leak sensitive information that can be traced back to particular individuals.

Differential Privacy (DP) is a mathematically justified approach that applies random noise to a dataset in such a way that it preserves statistical utility, but makes it impossible to distinguish if an individual has contributed to the results of analysis on a given dataset or not, thereby preventing privacy targeted attacks that use background information.

Introduction to Differential Privacy

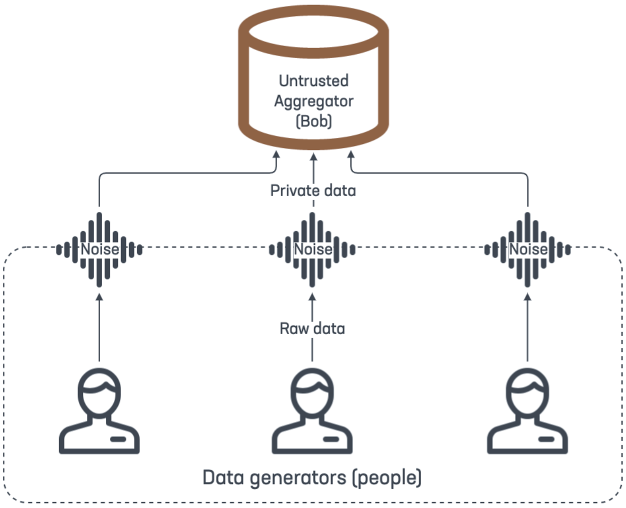
DP is a privacy preserving mechanism based on adding randomized statistical noise, where the probability of producing a computation result on two data sets – namely one with a given individual's information in it, and the other without – is nearly the same. Differential Privacy provides mathematical guarantees about individual Personally Identifiable Information (PII) and hence plays a vital role when it comes to data sharing. Sharing is better facilitated when individuals have guarantees that the result is nearly the same irrespective of whether their data is included in the set (e.g. for statistical analysis or algorithm training).

However, DP does not guarantee that an attacker will not learn about an individual even if their data is not part of the dataset being protected. DP only hides differences between datasets that differ by one individual, not whole groups. For example, assuming an individual answers a survey in a particular manner, applying DP techniques to protect the dataset will not allow an attacker to guess the answer an individual provided. However, if the individual's behavior is the exact same as their cohort, and if the attacker knows that some people in the group answered the survey in a certain manner, the attacker can guess how the said individual would have answered the survey. Essentially DP doesn't prevent an attacker from drawing conclusions about an individual based on known cohorts or general population.

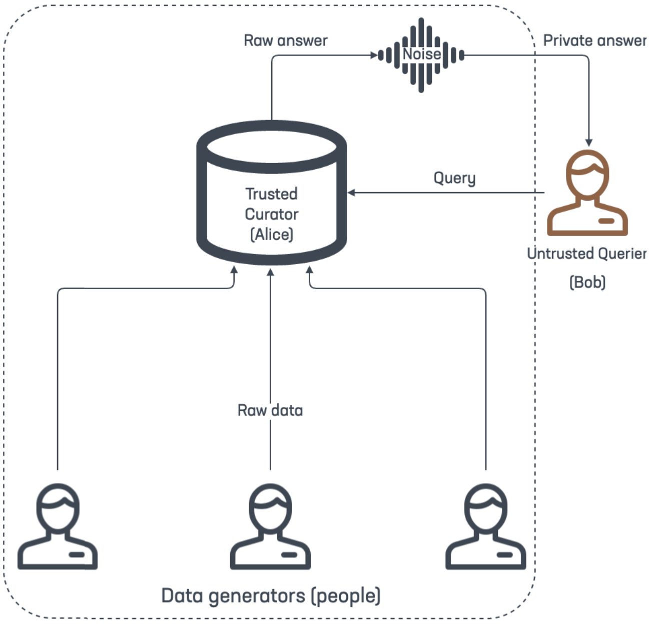
There are two common flavors of differential privacy - local and global.

- Global (or standard) differential privacy is performed on outputs to queries run on an already aggregated group-level dataset . This allows an individual to deny being part of a dataset based on the output of a query. Global DP is performed when the end users provide their raw data (without noise addition) to an aggregator (curator). The aggregator then applies DP techniques (adds noise) to transform this data and publish it. The disadvantage of this model is that the aggregator node needs to be trusted enough for the end users to send their raw data to it (example: Census Bureau).

- Local differential privacy is performed on individual data before any aggregation, similar to the randomised response method proposed by Warner in 1965 [18]. In this model, the aggregator doesn't need to be trusted because it doesn't have access to the raw user data – each end user applies noise to their own data before sending it to the aggregator. Hence, the aggregator can publish all the data collected from the end users.



Local privacy



Global privacy

The “one size fits all” approach of global DP also ignores the reality that data privacy is a personal concept, and that different individuals may have very different privacy expectations for their personal data. To cater to this, personalized differential privacy [16] has been proposed that applies differential privacy techniques at an individual-level instead of using a single, global parameter for all individuals in a database. In other words, a privacy budget is set for each record in a database instead of for the whole database.

Differential Privacy can be executed in an interactive or non-interactive setting. Interactive versions, as the name implies, entail a two-way communication between the data curator and the client which is querying the dataset. In order to respond to the client queries, the curator has to be online and keep track of the privacy budget assigned for each client. Non-interactive DP is a one-way protocol used for releasing the dataset to the public. The data is preprocessed by applying the DP mechanism and revealed to the ‘public’ for independent statistical analysis. In this setting, the data can be used by anyone to compute answers to any queries without the need to interact with the curator. One particular type of non-interactive mechanism is the generation of a synthetic dataset that allows the answers to a certain class of queries to be approximated [5, 6, 7].

Definitions

This section provides definitions for some DP parameters and properties.

- Query sensitivity - The sensitivity of a query or a function shows how much the output is affected by the addition or removal of a record in a dataset. Low sensitivity implies that the output is not severely affected by the change and only a small amount of noise is sufficient to preserve privacy.

- Privacy budget - One of the limits of differential privacy is that each time a differentially private computation takes place, there is a bit of privacy loss. Typically data curators set and enforce a maximum allowed privacy loss parameter which is referred to as the privacy budget. Each query is treated as a privacy expense. When many queries are allowed on the dataset, incremental privacy loss can result in the entire privacy budget being spent. It is important to carefully monitor the sensitivity of queries in order not to exhaust the budget too quickly and to be able to halt access to data (limit number of queries) if the budget is exceeded.

- Epsilon-DP - A function F (wrapping a query and introducing the randomness) provides ϵ-DP if the probability of F producing a given output changes by at most a multiplicative factor of e^{ϵ} when the input is changed by adding or removing a record of one person. The smaller the value ϵ, the better the privacy, but the worse the accuracy of results. In particular, $\epsilon = 0$ means perfect privacy, but unfortunately, the results will be useless because the noise will completely skew the output. On the contrary, larger values of ϵ allow better utility, but lead to lack of privacy. The main problem in DP is finding a balance between the privacy loss and the accuracy.

- Composability - Robustness under composition [1] is the property of DP that when multiple analyses are performed on data describing the same set of individuals, then, as long as each of the analyses satisfies differential privacy, it is guaranteed that all of the information released, when taken together, will still be differentially private. The privacy loss does accumulate, though, so each query may use only a part of the overall budget.

- Noise generation - Differential privacy is achieved by the addition of a certain amount of numerical noise to either the raw data or to the results of the queries performed on the raw data. The noise values are usually drawn from symmetric probability distributions (e.g. Laplacian or Gaussian), where the scaling parameter which affects the size of the picked numbers depends on the query sensitivity and a given budget.

- Approximate-DP - Epsilon-DP (also called pure DP) imposes certain complexity and inflexibility due to its strictness. The definition was relaxed with the concept of approximate differential privacy that introduced an additional parameter δ where the privacy guarantee needs to be satisfied only for events whose probability is at least $\approx \delta$ (i.e. it is ϵ-DP "except with probability δ") and even for very small δ, the complexity could be reduced significantly [9, 19].

What types of attacks does it help against?

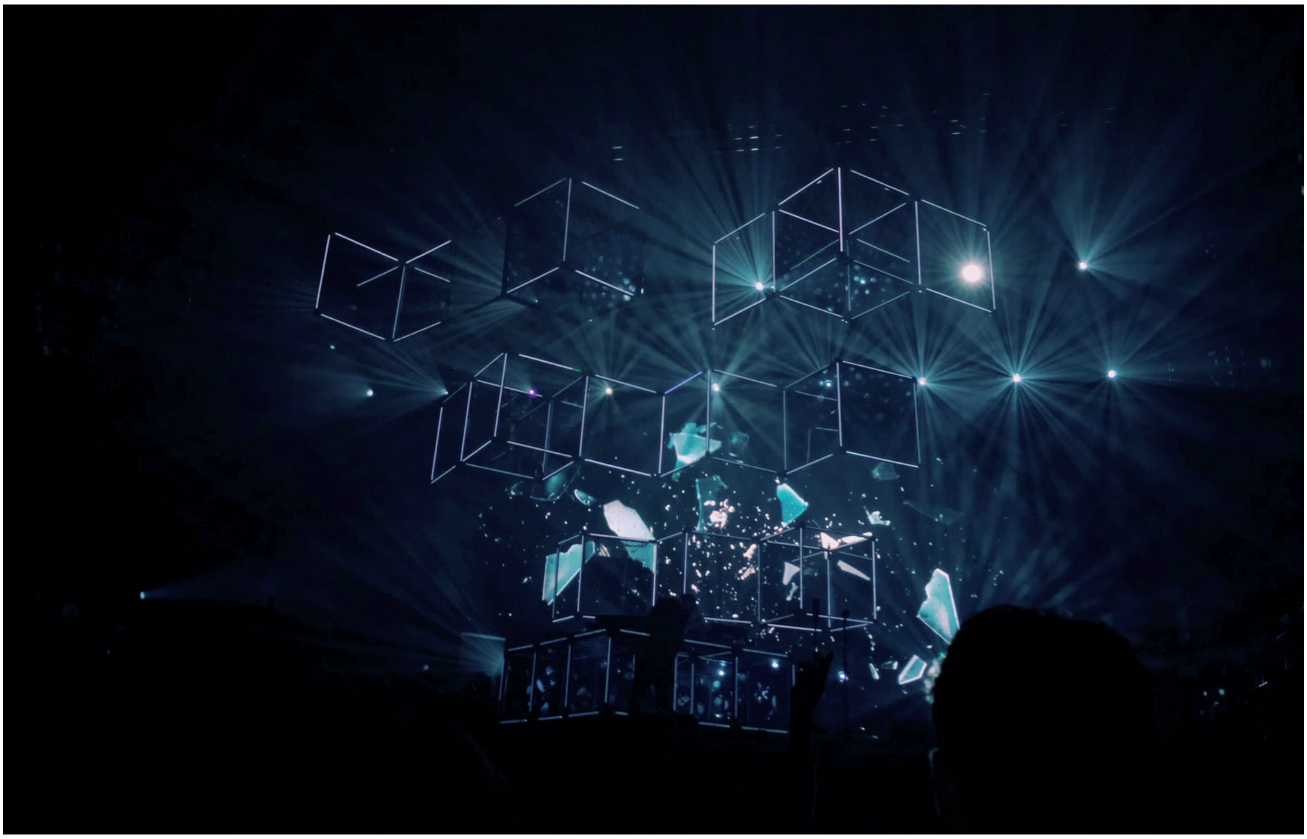
-

This section outlines some of the types of attacks that Differential Privacy helps protect against.

- Database reconstruction attacks (DRAs) [4,17] - Assuming the database to be a collection of rows, one per individual, with each row having a lot of non-private data and one secret bit per individual, the goal in a reconstruction attack is to determine the secret bits for nearly all individuals in the dataset.

- Re-identification attacks via record linkage or inference - This refers to the identification of one or more individuals in a de-identified dataset by uniquely linking a record in a de-identified dataset with records in a publicly available dataset. Sometimes individuals can be identified by inference, based on attributes specific to them. Differential privacy effectively hides the influence of an individual, or groups of individuals on queries performed on a dataset thereby providing protection against such attacks.

- Side channel attacks - This includes different types of attacks like timing attacks or privacy budget attack. When a certain condition is met, by intentionally pausing for a long time in the query code, the privacy mechanism reveals one bit (yes/no). The privacy budget attack checks how much the given privacy budget has decreased when the outer query returns. [2]



When does Differential Privacy fail?

-

Collusion

Suppose a DP system allocates a certain privacy budget ϵ to data analyst A and the same budget ϵ to data analyst B, for querying a dataset. Both analysts can make the same set of queries independently thereby using up their privacy budgets. Even with composably secure DP, if A and B decide to collaborate and share their answers, the total privacy loss might become 2ϵ. Collusion is typically only an issue for global DP model wherein a single party answers queries from different analysts each of whom might have their own privacy budget.

Correlation

DP might be vulnerable if it assumes all rows in a dataset (each from a different individual) are independent. It is very natural to have dependence due to social interactions between people (e.g. friendship relationships in social network graphs). Paper [15] demonstrates a Bayesian attack on DP using a real-world dataset exploiting the correlation between location and social information. In such a case, DP underestimates the amount of noise required to achieve the desired privacy bound, thereby enabling an adversary to perform sensitive inferences.

Is Differential Privacy universally applicable?

-

Unfortunately, DP isn't universally applicable. It must be understood that DP does not provide a general framework suitable for all domains and applications. To apply DP techniques, one needs to analyse the setting under which the data is being released or analysed, sensitivity of data and the kinds of queries required for the analysis. In other words, for each application domain, a differential privacy implementation should be carefully "handcrafted".

Differential Privacy at Guardtime

-

Guardtime is exploring potential use cases around applications for differential privacy. Some of these include:

- Data Collaboration Platform - There are instances where the data set is large and the custodian (e.g. a biobank) needs to give access to third parties to run analytics. Differential privacy can be applied when the data custodian needs to control the number and type of queries run on the data, to ensure no/acceptable privacy leaks.

- Supply Chain Shortages - In a supply chain, to determine inventory shortages, secure aggregation protocols are typically run on individual entities' inventory data to gather the total inventory for a given product without revealing individual data. Secure multi-party computation is commonly used to provide computational privacy on data aggregation done between mutually distrustful parties. However, it may need to be augmented with differential privacy to provide mathematical guarantees on privacy of the output. [8]

- Federated Learning - Federated learning trains AI models on end devices, and then transfers those learnings back to a global model without the need for data to leave the device. In order to handle potential privacy leaks from the updates that are being sent back to the global model, DP techniques are typically applied, thereby providing a privacy-preserving mechanism to effectively leverage the compute resources inside end devices to train machine learning models. The goal is to ensure that the learned model does not reveal:

    - whether a certain data point was part of the training data [11], or

    - if a client/end device contributed to the training data [10]

- Healthcare - Considering the example of an AI-in-healthcare setting, if multiple hospitals participate in training a centralized model, then differential privacy techniques can be applied to ensure that information about a specific patient stays hidden, or information about a specific hospital stays hidden. The impact of number of clients (hospitals in this case) on the accuracy and model performance is an area to be explored.

- Business Process Modeling - Differential privacy techniques (local or global as the case demands) can be applied to business process modeling to guarantee the privacy of intermediate data shared between different flows in a process. Differential Privacy tells us whether a given intermediate result or a final output of a process reveals information about a given input. This can be applied to use cases modeled by any general purpose workflow engine.

Conclusion

-

While there have been several prior efforts using anonymity, encryption, access control on queries etc to address privacy problems around the need for a data curator to release statistics over their dataset without revealing information about a particular value itself, differential privacy has proven far more successful owing to the rigorous definitions and mathematical guarantees it provides. Differential privacy aids in quantifying and bounding the amount of information leaked about individual records by the output of computations performed on a dataset.

In practice, however, there are challenges. There are no stringent guidelines on choosing privacy parameters. As analysts have shown, companies that employ differential privacy, Apple [13] and Google [14] in particular cut corners and implement weaker privacy techniques than they claim in their publications [12].

When deploying differential privacy techniques, there is always a trade-off between statistical accuracy and privacy loss. While there are several metrics to assess the quality of a published dataset, challenges remain in binding these metrics to legally defined risk factors. Also, the parameters need to be tuned carefully based on the complexity of the data and also of the queries allowed by the system.

Despite mature academic research in the area, industry adoption of differential privacy has been slow. However, with the increasing trends and visibility in the use of differential privacy in various real life applications, industry practitioners can use the lessons learnt from prior initiatives to successfully apply differential privacy techniques to address privacy breaches while overcoming practical challenges.

References

-

+ F. D. McSherry, “Privacy integrated queries: an extensible platform for privacy-preserving data analysis,” in Proceedings of the 2009 SIGMOD International Conference on Management of Data, pp. 19-30, ACM, 2009

+ P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, “GUPT: privacy preserving data analysis made easy,” in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 349-360, ACM, 2012

+ A. Narayanan, V. Shmatikov, “How to break anonymity of the netflix prize dataset,” arXiv preprint cs/0610105, 2006

+ S. Garfinkel, J. M. Abowd, and C. Martindale, “Understanding database reconstruction attacks on public data,” in Communications of the ACM, 62(3):46-53, ACM, 2019

+ A. Beimel, K. Nissim, and E. Omri, “Distributed private data analysis: on simultaneously solving how and what,” in Proceedings of Advances in Cryptology – CRYPTO 2008, pp. 451-468, Springer, 2008

+ Y. D. Li, Z. Zhang, M. Winslett, and Y. Yang, “Compressive mechanism: utilizing sparse representation in differential privacy,” in Proceedings of the 10th annual ACM workshop on privacy in the electronic society, pp. 177-182, ACM, 2011

+ N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, “Privacy preserving synthetic data release using deep learning,” in Proceedings of Machine Learning and Knowledge Discovery in Databases – ECML PKDD 2018, pp. 510-526, Springer, 2019

+ M. Pettai, P. Laud, “Combining differential privacy and secure multiparty computation,” in Proceedings of the 31st Annual Computer Security Applications Conference, pp. 421-430, ACM, 2015

+ A. Beimel, K. Nissim, and U. Stemmer, “Private learning and sanitization: pure vs. approximate differential privacy,” in Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques – APPROX 2013 and RANDOM 2013, pp. 363-378, Springer, 2013

+ M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308-318, ACM, 2016

+ R. C. Geyer, T. Klein, and M. Nabi, “Differentially private federated learning: a client level perspective,” arXiv preprint arXiv:1712.07557, 2017

+ B. Cyphers, “Understanding differential privacy and why it matters for digital rights,” Access Now, 2017, https://www.accessnow.org/understanding-differential-privacy-matters-digital-rights/

+ “Apple differential privacy technical overview,” Apple white paper, 2017, https://images.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

+ “Learning statistics with privacy, aided by the flip of a coin,” Google AI Blog, 2014, https://ai.googleblog.com/2014/10/learning-statistics-with-privacy-aided.html

- C. Liu, S. Chakraborty, and P. Mittal, “Dependence makes you vulnerable: differential privacy under dependent tuples,” in Proceedings of the Network and Distributed System Security Symposium – NDSS’16, Internet Society, 2016

- Z. Jorgensen, T. Yu, and G. Cormode, “Conservative or liberal? Personalized differential privacy,” in Proceedings of 2015 IEEE 31st International Conference on Data Engineering, IEEE, 2015, http://dimacs.rutgers.edu/~graham/pubs/papers/pdp.pdf

- C. Dwork, A. Smith, T. Steinke, and J. Ullman, “Exposed! A survey of attacks on private data:, in Annual Review of Statistics and Its Application, 4:61-84, 2017

- S. L. Warner, “Randomized response: a survey technique for eliminating evasive answer bias,” Journal of the American Statistical Association, 60(309):63-69,1965

- I. Mironov, “Rényi differential privacy,” in Proceedings of the IEEE 30th Computer Security Foundations Symposium – CSF 2017, pp. 263-275, IEEE, 2017

- P. Moulin, “Universal fingerprinting: capacity and random-coding exponents”, arXiv preprint arXiv:0801.3837, 2008

Appendix

-

Open source libraries

+ https://github.com/tensorflow/privacy
+ https://github.com/google/rappor
+ https://github.com/IBM/differential-privacy-library
+ https://github.com/uber/sql-differential-privacy

Commercial platforms

+ https://www.immuta.com
+ https://www.infosum.com
+ https://www.privitar.com
+ https://www.regdata.ch