Mathematical Problems of Computer Science 56, 48–55, 2021.

UDC 004.056.5

Differential Privacy in Practice: Use Cases

Karen A. Mastoyan

Gavar State University
e-mail: kmastoyan@yandex.com

Abstract

The problem of ensuring privacy is relevant in connection with the development of big data technologies. One of the modern and most promising methods of privacy protection is the differential privacy. In this paper the differential privacy applications developed by big companies are investigated. The libraries' capabilities and tools of Google, IBM, as well as packages in R are analyzed. The differential privacy process for data collected from users implemented by Apple is studied.

Keywords: Big data, Differential privacy, R environment.

1. Introduction

Big Data is an actual research topic because it provides new opportunities with data analysis for businesses and organizations to improve the decision making power. Various large companies, such as Facebook, Apple, Amazon and Google infiltrate users' personal lives and social interactions to accumulate huge databases at any time, which violates people's privacy. Along with the growth of data, it is necessary to develop new methods and means that will allow people to remain confidential.

A number of research articles are devoted to the study of Big data privacy, in the surveys [1] - [3] one can find detailed information and a full list of publications. In the role of information theory in the field of big data privacy is surveyed in [4].

There are various methods of confidentiality that allow for large-scale data analysis, statistical analysis, data (text) excavation, etc., while ensuring the privacy of individual participants. The most reliable approach is the Differential Privacy (DP).

DP is a modern approach to cyber security, where proponents argue that personal data is much better protected than traditional methods. DP is a strict mathematical definition of privacy [5], [6]. In the simplest terms, consider an algorithm that analyzes a dataset and calculates its statistics. It is said that such an algorithm is differentially private, if looking at the output, one cannot say whether anyone's data was included in the original database or not. In other words, the guarantee of a differential algorithm is that its behavior is unlikely to change when an individual joins or leaves a database. Anything that an algorithm can retrieve in a database that contains some individual information is almost as likely to come from a database without that individual information.

Most importantly, this guarantee is reserved for any individual and any dataset. Therefore, no matter how strange someone's details are, and no matter how much someone's details are in the database, the guarantee of differential confidentiality is still maintained. This provides a formal guarantee that individual-level information about database participants will not be leaked.

DP protects an individual's privacy by adding a few random noises to the database when analyzing the data [7]. Simply put, identifying personal information based on the results of the analysis by presenting noise will not work. However, after adding noise, the result of the analysis turns into an approximation, not an accurate result, which would be obtained only if it were conducted on a real database. In addition, it is possible that if the differential private analysis is performed several times, it may yield different results each time the randomness of the noise is presented in the databases.

In this article we discuss the application of DP by big companies such as Apple, Google, IBM. We study the libraries developed by Google, IBM and a package for R developed by Benjamin I. P. Rubinstein, called Brubinstein's diffpriv package, and analyze the capabilities and tools in them.

2. Usage of differential privacy

It is worth noting that DP works better on larger databases. The reason is that as the number of individuals in a database increases, so does the impact of any individual on a given aggregate statistic. DP can be applied to everything from warranty systems and social networks to deployment-based services. Example:

- Apple employs DP to accumulate anonymous usage insights from devices like iPhones, iPads and Mac.
- Amazon uses DP to access user's personalized shopping preferences while covering sensitive information regarding their past purchases.
- Facebook uses it to gather behavioral data for target advertising campaigns without defying any nation's privacy policy.
- There are various variants of differentially private algorithms employed in machine learning, game theory and economic mechanism design, statistical estimation, and many more.

2.1 Apple

Apple has mastered and developed a technique known in academia as local differential privacy to do something very interesting: gain insight into what many Apple users are doing while helping to protect the privacy of individual users. It's a technique that allows Apple to learn about the community of users without knowing about individuals in the community [8]. DP transforms the information shared with Apple before it ever leaves the user's device, so that Apple can never reproduce the actual data. The DP technology used by Apple is rooted in the idea that statistical noise, which is a bit biased, can disguise a user's personal information before it is shared with Apple. If many people share the same data, the added noise may on average exceed a large number of data points, and Apple can see that meaningful information

is emerging. DP is used as the first step in a data analysis system that includes strong privacy protection at every stage. The system is opt-in and designed to ensure transparency to the user. The first step is to privatize information using local differential privacy on the user's device. The purpose of the privatization is to ensure that Apple's servers do not receive clear data. Device specifications are removed from the data and transmitted to Apple via an encrypted channel. Apple Analytics system ingests differential private contributions, droping IP addresses and other metadata. The final step is consolidation, where customized protocols are developed to calculate relevant statistics, and the consolidated statistics are then shared with Apple's respective teams. Both ingestion and consolidation phases take place in a restricted environment, so even privatized data is not widely available to Apple employees. Apple's implementation of DP includes the idea of a perdonation privacy budget (measured by parameter ϵ), and imposes a strict limit on the amount of data transmitted by a user to maintain their privacy. The fact is that the slightly biased noise used in DP tends to outperform a large number of investments on average, which theoretically allows us to determine user activity information over a large number of views per user (although it is important to note that Apple does not associate any characteristic with information collected through DP). Apple uses local differential privacy to help protect the privacy of users' activities over a period of time, while gaining insight that improves intelligence and usability of features such as:

- QuickType suggestions,
- Emoji suggestions,
- Lookup Hints,
- Safari Energy Draining Domains,
- Safari Autoplay Intent Detection (macOS High Sierra),
- Safari Crashing Domains (iOS 11),
- Health Type Usage (iOS 10.2).

For each feature, Apple seeks to reduce the privacy budget while collecting enough data for Apple to improve the features. Apple stores the collected data for a maximum of three months. The sent data does not include any identifiers and the IP addresses are not stored. For Lookup Hints, Apple uses a privacy budget with $\epsilon = 4$ and limits the sending of user data to twice a day. For Emoji, Apple uses a privacy budget with $\epsilon = 4$ and requires a one-time daily data submission. For QuickType, Apple uses a privacy budget with $\epsilon = 8$ and collects data to twice a day. For Health Type Usage, Apple uses a privacy budget of $\epsilon = 2$ and limits the sending of user data to once a day. The submitted data does not include the health information itself, but what types of health data are edited by the users. For Safari, Apple limits the transfer of user data twice a day. For Safari domains, which are known to cause high power consumption or crashes, Apple uses a single privacy budget with $\epsilon = 4$. For Safari Auto-play intentional detection, Apple uses a privacy budget of $\epsilon = 8$.

Apple uses the Count Mean Sketch technique for DP, with which the initial information that is processed for sharing with Apple is encrypted using a number of hash functions, making easy the data representation in different sizes of a fixed matrix.

The data is encrypted using the SHA-256 account variants, followed by the privatization step, and then written to a chart matrix, the values of which originate from zero.

The noise injection step works as follows: after encoding as a vector function, each coordinate of the vector is then bent (written as an incorrect value) with a probability of

$$1/(1 + e^{\epsilon/2}),$$

where ϵ is the privacy parameter. This ensures that the analysis of the collected data cannot distinguish real values from deviated values, helping to ensure the confidentiality of the shared information. To stay within the privacy budget, the Apple OS does not send the entire chart matrix to the server, but only a random array of matrices. When the information encoded in the graphical matrix is sent to Apple, the Apple server presents the responses of all the devices that share the information and subtracts the average value for each element of the array. Although each presentation contains a lot of random elements, the average value of a large number of presentations gives Apple meaningful data.

The Hadamard Count Mean-based Sketch technology uses the noise injection method, which is similar to the method used in the Count Mean Sketch method, but with one important difference. It uses a type of mathematical operation called converting the Hadamard base to hashed encoding before performing the privatization step. Also, it only sends 1 bit randomly instead of the whole series, as in the Count Mean Sketch technique. This reduces communication costs by 1 bit due to some accuracy.

For each feature, Apple seeks to make the privacy budget small while still collecting enough data to enable Apple to improve the features. Apple retains the collected data for a maximum of three months. The donations do not include any identifier, and IP addresses are not stored.

2.2 Google

Google DP repository contains libraries to generate ϵ- and (ϵ, δ)-differentially private statistics over datasets. It contains the following tools.

- Privacy on Beam is an end-to-end DP framework built on top of Apache Beam. It is intended to be easy to use, even by non-experts.
- Three "DP building block" libraries in C++, Go, and Java implement basic noise addition primitives and differentially private aggregations. Privacy on Beam is implemented using these libraries.
- A stochastic tester is used to help catch regressions that could make the DP property no longer hold.
- A DP accounting library is used for tracking privacy budget.
- Google DP repository includes a command line interface for running differentially private SQL queries with ZetaSQL. You can use the Privacy on Beam laboratory to generate differential private data [9].

Currently, the DP building block libraries support the following algorithms:

Table 1: Google DP library algorithms [9].

Algorithm	C++	Go	Java
Laplace mechanism	Supported	Supported	Supported
Gaussian mechanism	Supported	Supported	Supported
Count	Supported	Supported	Supported
Sum	Supported	Supported	Supported
Mean	Supported	Supported	Supported
Variance	Supported	Supported	Planned
Standard deviation	Supported	Supported	Planned
Quantiles	Supported	Supported	Supported
Automatic bounds approximation	Supported	Planned	Planned
Truncated geometric thresholding	Supported	Supported	Supported
Laplace thresholding	Supported	Supported	Supported
Gaussian thresholding	Planned	Supported	Supported

Implementations of the Laplace and Gaussian mechanism use secure noise generation. These mechanisms can be used to perform computations that aren't covered by the algorithms implemented in our libraries. The DP building block libraries are suitable for research, experimental or production use cases, while the other tools are currently experimental and subject to change [10].

2.3 IBM DP library

IBM differential-privacy-library is comprised of four major components:

1. Mechanisms. These are the building blocks of DP, and are used in all models that implement DP. Mechanisms have little or no default settings, and are intended for use by experts implementing their own models. They can, however, be used outside of models for separate investigations, etc.

2. Models. This module includes machine learning models with DP. Diffprivlib currently has models for clustering, classification, regression, dimensionality reduction and preprocessing.

3. Tools. Diffprivlib comes with a number of generic tools for differentially private data analysis. This includes differentially private histograms, following the same format as Numpy's histogram function.

4. Accountant. The BudgetAccountant class can be used to track the privacy budget and calculate the total privacy loss using advanced composition techniques [11].

2.4 Brubinstein‘s diffpriv package in R (library)

Brubinstein‘s diffpriv R package implements generic mechanisms for DP, along with sensitivity sampler that replaces exact sensitivity bounds with empirical estimates. As a result, diffpriv privatizes a wide range of procedures under random DP, automatically, without

mathematical analysis and in many cases achieving high utility. Diffpriv is available from [12] under an open-source license. Brubinstein's diffpriv package makes privacy-aware data science in R easy. It implements the formal framework of DP: differentially-private mechanisms can safely release computed statistics, models fit, or arbitrary structures derived on privacy-sensitive data to untrusted third parties. Due to the worst-case nature of the framework, mechanism development typically requires involved theoretical analysis. diffpriv offers a turn-key approach to DP by automating this process with sensitivity sampling in place of theoretical sensitivity analysis [13].

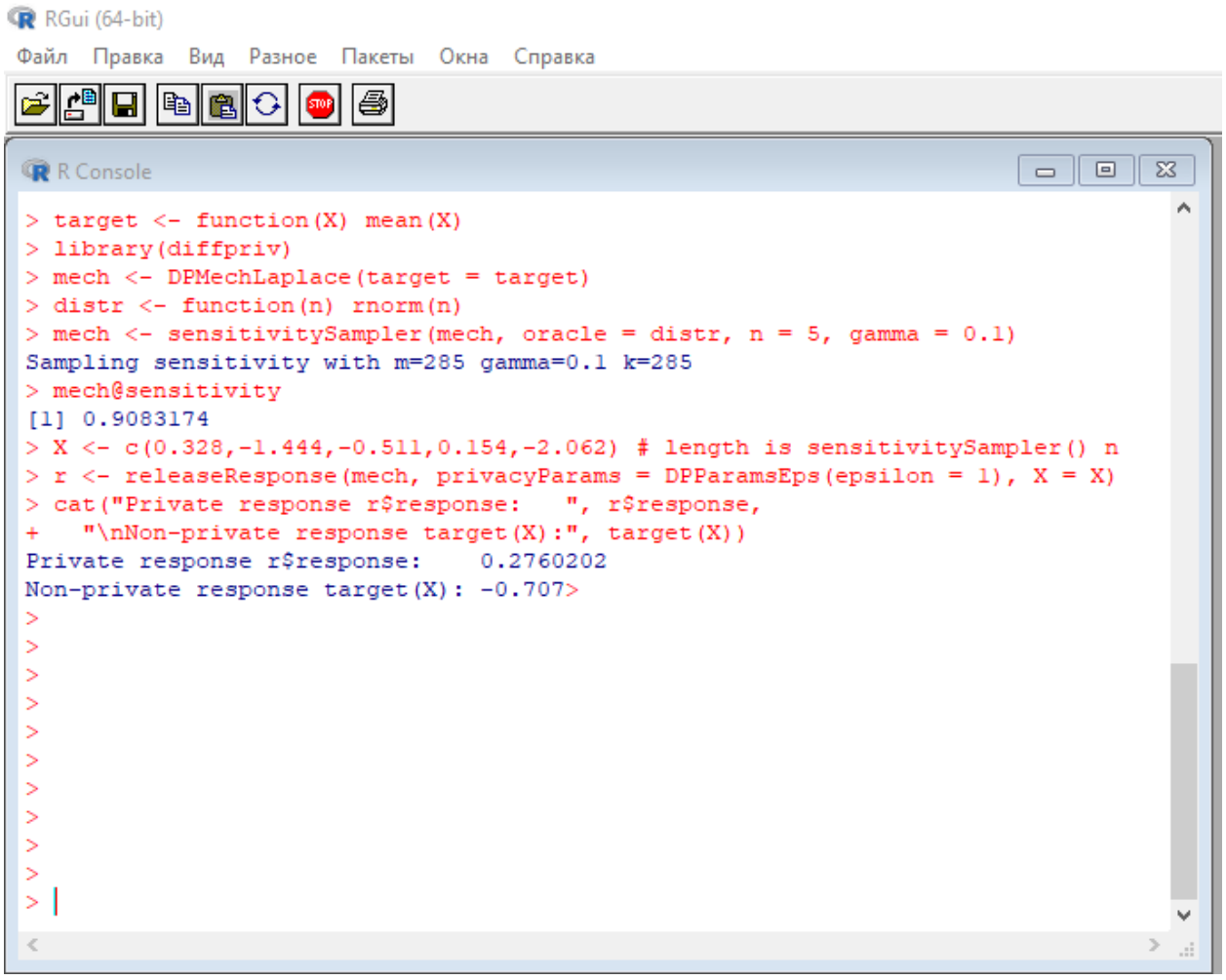

Fig. 1. Privately releasing a simple target function of privacy-sensitive input data X.

3. Conclusion

We investigated the implementation of DP by Apple, Google, IBM, as well as a package developed by Benjamin I. P. Rubinstein for R. As a result we can mention that Apple actively uses differential privacy in its systems, but does not provide open source libraries. The Google library is convenient for use with scripts written in C ++, Go, Java programming languages and allows the implementation of a number of algorithms, such as Laplace mechanism, Gaussian mechanism, Variance, Standard deviation, etc. IBM differential privacy library is written in python language and is more convenient to use in a python environment.

Brubinstein's diffpriv allows the implementation of the differential privacy process in R, which is open for various environments and algorithms. This research is useful for developing new DP applications and libraries.

References

[1] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua and S. Guo, "Protection of Big Data privacy", *IEEE Access*, vol. 4, pp. 1821–1834, 2016, doi: 10.1109/ACCESS.2016.2558446.

[2] L. Xu, C. Jiang, J. Wang, J. Yuan and Y. Ren, "Information security in Big Data: Privacy and data mining" *IEEE Access*, vol. 2, pp. 1149–1176, 2014, doi: 10.1109/ACCESS.2014.2362522.

[3] S. Yu, "Big Privacy: Challenges and opportunities of privacy study in the age of Big Data", *IEEE Acces*, vol. 4, pp. 2751–2763, 2016, doi: 10.1109/ACCESS.2016.2577036.

[4] M. Haroutunian and K. Mastoyan, "The role of information theory in the field of Big Data privacy, *Mathematical Problems of Computer Science*, vol. 55, pp. 45 - 53, 2021.

[5] C. Dwork, M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (eds) Automata, "Differential Privacy", *Languages and Programming. ICALP*, Lecture Notes in Computer Science, vol 4052. Springer, Berlin, Heidelberg, 2006. https://doi.org/10.1007/11787006

[6] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", *Foundations and Trends in Theoretical Computer Science*, vol. 9, no.3-4, pp. 211407, 2014.

[7] K. M. P. Shrivastva, M. A. Rizvi and S. Singh, "Big Data privacy based on differential privacy a hope for Big Data," *Proc. Intern. Conf. on Computational Intelligence and Communication Networks*, Bhopal, India, pp. 776–781, 2014. doi: 10.1109/CICN.2014.167.

[8] Differential Privacy Team, Apple, Learning with Privacy at Scale, [Online]. Available: https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf

[9] End-to-end differential privacy solution, [Online]. Available: https://github.com/google/differential-privacy/tree/main/privacy-on-beam

[10] Google Developers, Google, Enabling developers and organizations to use differential privacy, [Online]. Available: https://developers.googleblog.com/2019/09/enabling-developers-and-organizations.html

[11] Naoise Holohan, Stefano Braghin, Pol Mac Aonghusa and Killian Levacher. Diffprivlib: The IBM Dierential Privacy Library, 2019 https://arxiv.org/pdf/1907.02444.pdf

[12] Differential privacy package using R, [Online]. Available: https://github.com/brubinstein/diffpriv

[13] B. I. P. Rubinstein and A. Francesco, "diffpriv: An R package for easy differential privacy", *Journal of Machine Learning Research*, vol. 18, pp. 1-5, 2017.

Submitted 16.08.2021, accepted 03.11.2021.

Դիֆերենցիալ գաղտնիություն գործնականում. կիրառման դեպքեր

Կարեն Ա. Մաստոյան

Գավառի պետական համալսարան
e-mail: kmastoyan@yandex.com

Ամփոփում

Գաղտնիության ապահովման խնդիրն արդիական է մեծ տվյալների տեխնոլոգիաների զարգացման հետ կապված: Գաղտնիության պաշտպանության ժամանակակից և ամենախոստումնալից մեթոդներից մեկը դիֆերենցիալ գաղտնիությունն է: Այս հոդվածում ուսումնասիրվում են խոշոր ընկերությունների կողմից մշակված գաղտնիության իրականացման տարբեր հավելվածներ: Վերլուծվում են Google-ի, IBM-ի գրադարանների և գործիքների հնարավորությունները, ինչպես նաև՝ դիֆերենցիալ գաղտնիությունը R փաթեթում: Ուսումնասիրվում է օգտատերերից հավաքագրված տվյալների դիֆերենցիալ գաղտնիության գործընթացը՝ ներդրված Apple-ի կողմից: Այս հետազոտությունը օգտակար է գաղտնիության ապահովման նոր հավելվածներ և գրադարաններ մշակելու համար:

Բանալի բառեր՝ Մեծ տվյալներ, դիֆերենցիալ R միջավայր:

Дифференциальная конфиденциальность на практике: варианты применения

Карен А. Мастоян

Гаварский государственный университет
e-mail: kmastoyan@yandex.com

Аннотация

Проблема обеспечения конфиденциальности актуальна в связи с развитием технологий больших данных. Одним из современных и наиболее перспективных методов защиты конфиденциальности является дифференциальная конфиденциальность. В этой статье исследуются приложения дифференциальной конфиденциальности, разработанные крупными компаниями. Анализируются возможности библиотек и инструментов Google, IBM, а также пакетов в R. Изучается процесс дифференциальной конфиденциальности, реализованный Apple, для данных, собранных от пользователей. Это исследование полезно для разработки новых приложений и библиотек дифференциальной конфиденциальности.

Ключевые слова: Большие данные, дифференциальная конфиденциальность, среда R.