Identifying the optimal differential private mechanisms for different users*

Ehab ElSalamouny[1,2], Konstantinos Chatzikokolakis[1], and Catuscia Palamidessi[1]

[1] INRIA and LIX, Ecole polytechnique, France

[2] Faculty of Computer and Information Science, Suez Canal University, Egypt

Abstract. The notion of differential privacy has emerged in the area of statistical databases to provide a protection for the sensitive information about participants in these databases. Without concerning the privacy protection, participants' sensitive information can be leaked easily to an attacker by performing selected queries on such databases. Differential privacy is satisfied using a 'randomisation' mechanism which provides the user with a 'noisy' answer for her query instead of the exact answer. The privacy is therefore achieved at the cost of reducing the accuracy (or 'utility') of the user's query answer. A trend of research has recently directed to finding the 'optimal' differentially private mechanisms which provide a trade-off between privacy and utility. The main challenge is that an optimal mechanism for a user depends on both the database query and the user's side information about possible query results, modelled as a 'prior' probability distribution over these results.

In this work we describe, for a general query and privacy level, a randomisation mechanism which satisfies differential privacy and at the same time is optimal for a class of users having various priors. We present the properties of this mechanism in terms of utility and privacy, and also characterise the class of users for whom it is optimal.

1 Overview

Statistical databases are commonly used to provide statistical information about the individuals of a certain population to attain a social benefit. These databases usually store sensitive information about participants. Statistical queries applied to these database are, for instance, the average salary of individuals in a particular organisation, or the number of individuals having certain disease. The results of these queries are useful for e.g. financial planning, or studying diseases. However the availability of these results represent a major threat for the privacy of participants in the databases.

To illustrate this privacy issue, consider a database which contains the salaries of individuals. Suppose it is required to keep the salary of each individual hidden (private), while allowing only queries which yield the sum of all salaries. If an individual i is

* This work has been partially supported by the project ANR-09-BLAN-0169-01 PANDA and by the INRIA Action d'Envergure CAPPRIS.

known to be removed from the database, then i's salary can be easily inferred as the difference between the total salaries before and after her removal. The 'private' salary of i can therefore be disclosed using the sum query.

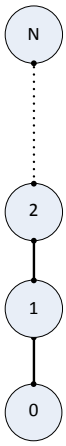
A successful approach to solve the above problem is to supply the user with a 'noisy' answer instead of the exact query answer. The noisy answer is produced by adding 'random' noise to the exact answer. This procedure is represented by a 'randomisation function' $\mathcal{K}$ which processes the database v as an input and produces a noisy output o in some domain O. The privacy is therefore achieved due to the uncertainty of the user in guessing the true query answer from the observed output. The notion of differential privacy introduced by Dwork ([6, 9, 7, 8]) provides a means to quantify the level of privacy guaranteed by a randomisation function. A function $\mathcal{K}$ is ϵ-differentially private (for some $\epsilon > 0$) if the ratio between the probabilities of producing the same answer given two 'adjacent' databases v, v' does not exceed e^{ϵ}. The adjacency relation between v and v' (written as $v \sim v'$) means that they differ in only one entry.

In fact, any randomisation function $\mathcal{K}$ can be described as a cascade of two functions: the query function Q and a randomisation mechanism $\mathcal{H}$. The function Q corresponds to the database query, e.g. count, average, etc, which maps the database to the set of possible query results $\mathcal{R}$. The randomisation mechanism $\mathcal{H}$ adds random noise to the exact query result $r \in \mathcal{R}$ and produces a 'noisy' output $o \in O$ to the user.

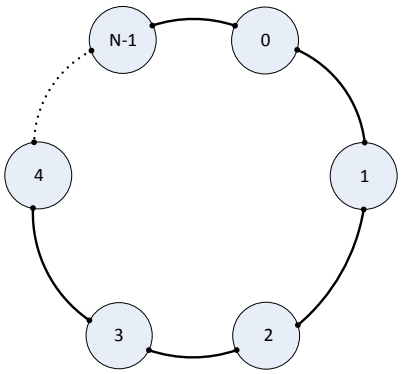
Note that privacy guarantees are provided by the randomisation mechanism $\mathcal{H}$, due to the added noise. In this work we will assume that the mechanism $\mathcal{H}$ is 'oblivious', that is it depends only on the exact query result r regardless of the underlying database. Under this assumption, the mechanism $\mathcal{H}$ can be regarded as a matrix $\boldsymbol{x}$ of probabilities x_{ro} where $r \in \mathcal{R}$ and $o \in O$. With this representation x_{ro} denotes the probability of producing output o when the exact query answer is r. In the following we will refer to the randomisation mechanism by the associated matrix $\boldsymbol{x}$.

The adjacency relation between databases induces an analogous adjacency relation between query results. We call two query results r, r' adjacent if they differentiate between two adjacent databases, that is, if there exists two adjacent databases v, v' such that $Q(v) = r$ and $Q(v') = r'$. Using this notion of adjacency between query results, a graph notation can be used to model the query results along with the adjacency relationship. More precisely, for a given query, the set of nodes in the corresponding graph represents the exact query results $\mathcal{R}$, while edges represent the adjacency relationship between pairs of adjacent query results. It is worth noting that graph structure of queries have been used also in [1, 5] to analyse the differentially private mechanisms. Figure 1 shows examples of the graph structures of different queries. In these examples $count(v, p)$ refers to a counting query which returns the number of records in the database v which satisfy a certain property p. Other queries in the figure are expressed using the *count* function.

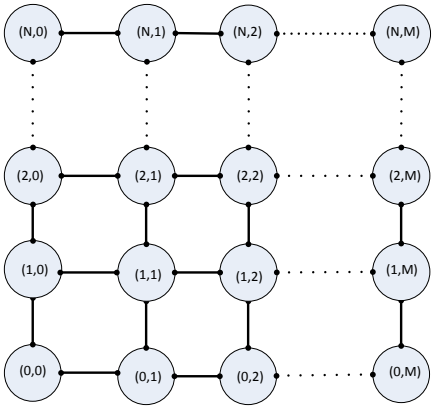
With the graph structure of a query, the 'distance' between two query results i, h, denoted by $d(i, h)$, is the graph distance between them. Similar to the adjacency relation, it is easy to see that the distance relation is symmetric. It also coincides with the adjacency relation when $d(i, h) = 1$. Using this notion of distance, the formulation of differential privacy can be lifted from a condition on pairs of adjacent databases to a condition on any pair of query results (nodes) according to the following theorem.



A count query $Q(v) = count(v, p)$



Query $Q(v) = count(v, p) \bmod N$



A multiple-count query $Q(v) = (count(v, p_1), count(v, p_1))$

Fig. 1. Examples for the graph structures of different queries

Theorem 1. *A randomisation mechanism* $\boldsymbol{x}$ *satisfies* ϵ*-differential privacy if and only if for all query results* i, h *and all mechanism outputs* o *it holds*

$$x_{io} \leq e^{\epsilon\, d(i,h)} \cdot x_{ho}$$

That is, the ratio between the probability of producing an answer o given that the query result is r and the probability of giving the same output o given that the query result is h does not exceed $e^{\epsilon\, d(i,h)}$.

The objective of a randomisation mechanism $\boldsymbol{x}$ is to guarantee the differential privacy of the database, while providing the user with 'useful' information about the exact query result. That is to satisfy a trade off between the privacy and utility.

The utility achieved by a mechanism refers to how informative, on average, is the reported answers to the user. The utility depends therefore on a numeric gain function $g(r, o)$ which defines the gain for the user when the real query result is r and the reported output is o. While the domain of query results $\mathcal{R}$ is, in general, independent of the domain of outputs $\mathcal{O}$, we assume for simplicity that $\mathcal{R} = \mathcal{O}$. That is the randomisation mechanism reports to the user an output drawn from the domain of query results. Under this assumption, the gain function can simply quantify the accuracy of the reported output compared to the real query result. That is, the closer are exact query result and the reported output, the higher is the gain. In the current analysis we choose the binary gain function where the $g(r, o) = 1$ when $r = o$ and 0 otherwise. The utility of the randomisation mechanism is defined as the expected value of the gain function.

Note that in addition to the mechanism $\boldsymbol{x}$, the utility depends on the gain function and also the probability distribution over query results, known as the 'prior' distribution. This prior is in fact relative to the user and models the side-information which the user may have about the database. This is because using such information, the user can estimate the likelihood of each possible value of her query. Suppose for example that a user Alice knows that Bob's salary is 20 K, while others have less salaries, and there is a record for Bob in the salaries database. Thus Alice knows that the sum of the salaries of participants in the database must be larger than Bob's salary. This is reflected on Alice's prior for the sum query results such that the total probability mass is distributed on the range of values > 20 K, while assigning 0 probability to values to low values.

The 'optimal' ϵ-differentially private mechanism for a given prior is defined as the mechanism $\boldsymbol{x}$ which maximises the utility function, while satisfying ϵ-differential privacy. A strong result for the counting queries was obtained by Gosh et al ([10]), that the truncated geometric mechanism is optimal for all users if each user remaps the output of the mechanism to her best guess according to her prior. In this sense, this mechanism is called universally optimal for the counting query. An carious question immediately arise whether a similar result holds for other queries. Gupte and Sundararajan answered this question negatively in [11] showing the impossibility of a universally optimal mechanism for arbitrary query. Therefore it remains to find, for a general query, a privacy-preserving mechanism which is not necessarily optimal for all users, but is, at least, optimal for a class of different users.

In this work we study, for arbitrary queries and privacy parameter ϵ, a particular mechanism called the 'tight-constrained' mechanism which is determined by the graph

structure of the given query and ϵ. This mechanism is instantiated to the truncated geometric mechanism ([10]) when one considers the counting queries. It is also instantiated to the optimal mechanism constructed in [1], for the uniform prior and the queries whose graph structures have certain symmetry properties. It turns out that the tight-constrained mechanism is optimal for a variety of priors, including the uniform prior. In the following section we summarise the properties of this mechanism in terms of the privacy, utility, and also information leakage.

2 Summary of results

The first and essential property of a tight-constrained mechanism is its differential privacy. For any query there is always a value for the differential privacy parameter $\epsilon > 0$ such that the tight-constrained mechanism exists and satisfies ϵ-differential privacy. The values of ϵ allowing for the tight-constrained mechanism depend on the query.

As a second property of the tight-constrained mechanism expresses its optimality for a family of users. We find that the tight-constrained mechanism maximises the utility (based on the binary gain function) for range of priors corresponding to different users. This range of priors depends essentially on the graph structure of the query and the privacy parameter ϵ.

Now we consider the special and important case of the uniform prior, where all real query results are equally likely. Note that this prior corresponds to users having unbiased information about the possible query results (all results can be yielded with the same probability). In this case, it turns out that the tight-constrained mechanism is optimal for a given query and a privacy parameter value ϵ.

The last property of the tight-constrained mechanism is information-theoretic and related to the line of research pursued to quantify the leakage of information channels ([14, 4, 12, 2]). Regarding any randomisation mechanism as an information channel, we find that, for a given ϵ, the tight-constrained mechanism leaks the maximum information compared to other ϵ-differentially private mechanisms. Here we follow [1, 3] in adopting the min-entropy leakage as a measure for the channel's information leakage. This measure is based on Rényi's notion of min-entropy [13] as a measure for uncertainty.

3 Future work

In the presented work any user is assumed to take the output of the mechanism as the best approximation (guess) for the exact query result. In general, the user can have a better 'guess' for the exact result, based on the observed output and her own side information (prior). In other words she can map the mechanism output to another value to improve the utility. We are investigating the impact of this remapping on enlarging the class of users (priors) for which the tight-constrained mechanism is optimal.

References

1. Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between Differential Privacy and Quantitative Information Flow. In Jiri Sgall Luca Aceto, Monika Henzinger, editor, *38th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 6756 of *Lecture Notes in Computer Science*, pages 60–76, Zurich, Switzerland, 2011. Springer.
2. Miguel E. Andrés, Catuscia Palamidessi, Peter van Rossum, and Geoffrey Smith. Computing the leakage of information-hiding systems. In Javier Esparza and Rupak Majumdar, editors, *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010)*, volume 6015 of *Lecture Notes in Computer Science*, pages 373–389. Springer, 2010.
3. Gilles Barthe and Boris Köpf. Information-theoretic bounds for differentially private mechanisms. In *Proceedings of CSF*, 2011. To appear.
4. Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proceedings of the 25th Conf. on Mathematical Foundations of Programming Semantics*, volume 249 of *Electronic Notes in Theoretical Computer Science*, pages 75–91. Elsevier B.V., 2009.
5. Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 71–80. IEEE Computer Society, October 23-26 2010.
6. Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *33rd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.
7. Cynthia Dwork. Differential privacy in new settings. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 174–183. SIAM, 2010.
8. Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–96, 2011.
9. Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 371–380, Bethesda, MD, USA, May 31 - June 2 2009. ACM.
10. Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proc. of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 351–360. ACM, 2009.
11. Mangesh Gupte and Mukund Sundararajan. Universally optimal privacy mechanisms for minimax agents. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '10, pages 135–146, New York, NY, USA, 2010. ACM.
12. Boris Köpf and Geoffrey Smith. Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium (CSF'10)*, pages 44–56. IEEE Computer Society, 2010.
13. Alfréd Rényi. On Measures of Entropy and Information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability*, pages 547–561, 1961.
14. Geoffrey Smith. On the foundations of quantitative information flow. In Luca de Alfaro, editor, *Proc. of the 12th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 5504 of *LNCS*, pages 288–302, York, UK, 2009. Springer.