How Auditors May Inadvertently Compromise Your Privacy

Position Paper

Nina Mishra* Kobbi Nissim†

Introduction Let D be a database containing sensitive information about n individuals. For example, D may consist of medical or census data. We consider the setting where statistical or aggregate queries are posed against the database D. The privacy challenge, which we refer to as the *statistical database privacy problem*, is to ensure that answering such queries does not leak information about individuals.

Solutions to the statistical database privacy problem can be broadly characterized into perturbation and query restriction methods. In the perturbation family of methods, either noise is added to the input data D or noise is added to the output query responses. In the query restriction family of methods, the trail of queries is monitored to ensure that it is not possible to combine answers to queries so as to deduce information about any individual. We are concerned with the latter query restriction style of statistical database solutions, in particular with a subclass of restriction methods called auditing (defined below). A 1989 survey of statistical databases can be found in [1].

To formally define the auditing problem, assume for the sake of simplicity that the database D is a one-dimensional dataset, i.e., $D = \{d_1, \ldots, d_n\}$ where each $d_i \in R$ (or $d_i \in \{0, 1\}$) A statistical query $q = (Q, f)$ specifies a subset of entries $Q \subseteq [n]$ and a function f like sum, max, min, or median. The answer to the query is the value $f(\{d_j | j \in Q\})$ or "denied" if answering the query leaks information about an individual. The *query auditing problem* is: Given a sequence of queries, $q_1, \ldots, q_i$, corresponding answers $a_1, \ldots, a_i$, and given a new query q_{i+1}, provide an answer to the query a_{i+1} if and only if for all d_j it is not possible to uniquely determine d_j from the queries $q_1, \ldots, q_i, q_{i+1}$ and the answers $a_1, \ldots, a_i, a_{i+1}$. In the event that some d_j can be compromised, the auditor denies the answer to the query, i.e., a_{i+1} is "denied"[1].

The query auditing problem is particularly important to solve in the context of medical databases where, on the one hand, medical researchers require the ability to pose queries to datasets in order to gain medical insight and, on the other hand, should not be able to stitch together the answers to queries so as to deduce information about a single patient. The query auditing problem is also important for other types of data, including census data.

Research on the query restriction problem has considered allowing queries that involve a lower bounded number of elements (query-set size control), limiting the overlap between queries (query-set-overlap control), and upper bounding the number of queries posed to a database [5, 9]. We do not consider such types of restriction methods since auditing may allow more and/or different kinds of queries since the auditor is tuned to the specific query/answer trail for privacy breaches. Other research on query restriction has

*HP Labs and Stanford University, Research supported in part by NSF Grant EIA-0137761, Palo Alto, CA.

†Microsoft Corporation, Mountain View, CA.

[1]Note that in this definition, at least one value of the dataset must be exactly compromised in order for a privacy breach to occur. Other definitions of privacy also exist, for example, a privacy breach occurs if there are a small number of datasets consistent with the queries and answers, but these definitions are not considered here.

focused on providing efficient (polynomial-time) algorithms for auditing SUM, MAX, MIN, and Interval-based queries [2, 3, 8]. More recently, it has been shown that there is unlikely to be an efficient SUM query auditor in the case that the input data is Boolean, i.e., $d_i \in \{0, 1\}$ [7]. Additional results in this direction can be found in [6]. In contrast to the work presented here, research on the query auditing problem has largely ignored denials in the answers $a_1, \ldots, a_i$ when deciding whether to answer q_{i+1}[2].

Spectrum of Query Restriction Methods We define a spectrum of query restriction methods that vary depending on the type of information used by the query restrictor in order to answer a query. On the lower end of the spectrum, the restrictor uses combinatorial properties of the queries, such as the size of each query, the overlap between pairs of queries and the number of queries in order to determine whether to answer the new query. On the highest end of the spectrum, the query restrictor (auditor) uses the queries $q_1, \ldots, q_{i+1}$ and the answers $a_1, \ldots, a_{i+1}$, and the database in order to determine how to answer q_{i+1}. We characterize prior research on query restriction based on where it falls in the spectrum.

Inadvertent Privacy Compromise We uncover a fundamental problem in auditing algorithms at the highest end of the spectrum, namely that query denials leak information. Consequently, the denials themselves can be adversarially used to leak information. We illustrate the problem through simple, revealing examples of auditing SUM, interval-based SUM, and MAX queries.

Simulation – When Denials Do Not Leak Information On the positive side, we consider a point in the spectrum put forward in [4] where query denials provably do not leak information. We argue that this place in the spectrum is an interesting launching point for future work on query auditing. Informally, we draw upon the simulation paradigm, used in cryptography, and consider a model where a user may simulate the denial responses given by the auditor. In particular, we show that since the user can simulate the responses given by the auditor, the denials themselves do not leak any information.

References

[1] N. Adam and J. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[2] F. Chin. Security problems on inference control for sum, max, and min queries. *J. ACM*, 33(3):451–464, 1986.

[3] F. Chin and G. Ozsoyoglu. Auditing for secure statistical databases. In *Proceedings of the ACM '81 conference*, pages 53–59. ACM Press, 1981.

[4] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.

[5] D. Dobkin, A. Jones, and R. Lipton. Secure databases: protection against user influence. *ACM Trans. Database Syst.*, 4(1):97–106, 1979.

[6] P. Jonsson and A. Krokhin. Computational complexity of auditing discrete attributes in statistical databases. In *Manuscript*, 2003.

[7] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *PODS*, pages 86–91, 2000.

[8] Y. Li, L. Wang, X. Wang, and S. Jajodia. Auditing interval-based inference. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pages 553–567, 2002.

[9] S. Reiss. Security in databases: A combinatorial study. *J. ACM*, 26(1):45–57, 1979.

[2] Previous work has largely considered the 'offline' version of the problem, where the auditor is given an offline set of queries and either answers or denies all of them. Note however that with such 'offline' auditors the database has to be closed down after the first 'denial' (even if it occured in the first query) as a further run of the auditor would make it 'online'.