Ask a Better Question, Get a Better Answer
A New Approach to Private Data Analysis

Cynthia Dwork

Microsoft Research
dwork@microsoft.com

Abstract. Cryptographic techniques for reasoning about information leakage have recently been brought to bear on the classical problem of *statistical disclosure control* – revealing accurate statistics about a population while preserving the privacy of individuals. This new perspective has been invaluable in guiding the development of a powerful approach to private data analysis, founded on precise mathematical definitions, and yielding algorithms with provable, meaningful, privacy guarantees.

1 Introduction

The problem of *statistical disclosure control* – revealing accurate statistics about a population while preserving the privacy of individuals – has a venerable history. An extensive literature spans multiple disciplines: statistics, theoretical computer science, security, and databases. In recent years the problem has been revisited, bringing to the discussion techniques from the cryptographic community for defining and reasoning about information leakage. This new perspective has been invaluable in guiding the development of a powerful approach to private data analysis, founded on precise mathematical definitions, and yielding algorithms with provable, meaningful, privacy guarantees and, frequently, excellent accuracy.

Statistical databases may be of two types: non-interactive (the traditional model) and interactive. In the former, a *sanitization* of the data is published. All statistical analysis is carried out on the published, sanitized, data. Sanitization is a broad concept, and can include summaries, histograms, and even synthetic databases generated from a model learned from the actual data. The principal aspect here is the "one-shot" nature of the non-interactive approach: once the sanitization has been published the original data have no further use; they could even be destroyed. In contrast, in the interactive model a privacy mechanism sits between the data and the user. The user interacts with the privacy mechanism, which may modify the actual query or the query outcome, in order to preserve privacy.

The division between the models is somewhat artificial; nevertheless, separation results exist, and it is now clear that the interactive setting is much more powerful; for example, to obtain statistically meaningful information in the non-interactive case can provably require a huge database (exponential in the number

T. Schwentick and D. Suciu (Eds.): ICDT 2007, LNCS 4353, pp. 18–27, 2007.

of attributes) [12], which is simply not the case for interactive mechanisms. We may use the term *privacy mechanism* for either type of mechanism.

Dinur and Nissim [7] initiated a rigorous study of the interactive model; in particular, they focused on a class of techniques that Adam and Wortmann, in their encyclopedic 1989 survey of statistical disclosure control methods, call *output perturbation* [1]. Roughly speaking, this means that noise is added to the output of the query, so a true answer of, say, 4,286, may be reported as 4,266 or 4,300. The degree of distortion, that is, the expected magnitude of the noise, is an important measure of the utility of the statistical database. Dinur and Nissim investigated the question of how large the magnitude of the noise must be when the number of queries is large.

They began with a very simplistic and abstract setting, in which the database consists of a single Boolean attribute. That is, each row of the database is either zero or one. A *query* is a subset of the rows, and the defined true answer to the query is the sum of the rows in the subset (equivalently, the number of ones in the specified set of rows). It is helpful to think of the query as a vector $x \in \{0,1\}^n$, where n is the number of rows in the database, henceforth denoted DB. The true answer to the query is $x \cdot DB$. An output perturbation mechanism adds noise to the true answer, and returns this sum as the response to the query. We use the terms *true answer* to denote the real number of ones in the rows specified by the query, and *response* to denote the output of the privacy mechanism.

Dinur and Nissim did not initially explicitly define privacy. Instead they defined what we will call *blatant non-privacy*: the ability to reconstruct, say, 99.99%, or, more precisely, $n - o(n)$, entries of a database of n rows (the adversary will not necessarily know which of the reconstructed entries are the correct ones). They showed that to prevent blatant non-privacy, the magnitude of the noise added in each response cannot always be small:

1. The magnitude of the noise cannot always be $o(n)$ if the adversary can make 2^n queries to the database (in fact, if the error is always within a bound E then the database can be approximated by a candidate of Hamming distance at most $O(E)$ from the real database);
2. If the adversary is polynomial time bounded and makes only $O(n \log^2 n)$ randomly chosen queries, the magnitude of the noise cannot always be $o(\sqrt{n})$.

These results are independent of the distribution of the noise.

The first result uses brute force to rule out databases that are too far from the actual database. The second uses linear programming to accomplish the same task; the result holds with all but negligible probability over the choice of queries.

The Dinur-Nissim setting, while at first blush simplistic, is in fact sufficiently rich to capture many natural questions. For example, the rows of the database may be quite complex, but the adversary-user may know enough information about an individual in the database to uniquely identify his row. In this case the goal is to prevent the learning of any *additional* bit of information about the individual. Of course, even knowing enough to identify a single individual does not give the adversary the power to identify everyone in the database. However,

careful use of hash functions can handle the "row-naming problem." Thus, we may have a scenario in which an adversary reconstructs a close approximation to the database, in which each row is identified with a set of hash values, and a "secret bit" is learned for many rows. If the adversary knows (or later learns) enough about an individual to identify, directly or through elimination, his row in the database, then the adversary can learn the individual's secret bit.

"Just Give Me a Noisy Table". Research statisticians like to "look at the data." Indeed, conversations with experts in this field frequently involve pleas for a "noisy table" that will permit significantly accurate answers to be derived for computations that are not specified at the outset. The Dinur-Nissim results say that no "noisy table" can provide very accurate answers to all questions; otherwise the table could be used to simulate the interactive mechanism, and a Dinur-Nissim style attack could be mounted against the table. But what about a table that yields reasonably accurate answers to "most" questions, permitting some questions to have wildly inaccurate answers? We will see in Section 2 that this relaxation is of little help in protecting privacy. We therefore advocate switching to an interactive strategy using the techniques of Section 3.

1.1 When n Is Very Large

Dinur and Nissim obtained their negative results while we were thinking about privacy for enormous databases, in particular, the Hotmail user database of over $n = 100,000,000$ users. In such a setting, asking $n \log^2 n$ queries is simply unreasonable. This suggests the following natural question: suppose the number of queries is limited, so the attacks above cannot be carried out. For example, suppose the number of queries is sub-linear in n. Can privacy be preserved by noise that is, say, always of magnitude $o(\sqrt{n})$? Since the sampling error for a property that occurs in a constant fraction of the population is on the order of $\Theta(\sqrt{n})$, this would mean that the noise added for protecting privacy is smaller than the sampling error.

More generally, let T be an upper bound on the number of queries to be tolerated. What magnitude noise is sufficient to ensure privacy against T queries? As we will see, the answer to this question is very satisfactory. In particular, the magnitude of the noise will depend only on T, and not on n.

To answer our question we must pose it precisely, which means that we must define privacy, preferably in a way that makes sense for arbitrary databases, and not just n-bit vector databases. Of course, when the databases are arbitrary the queries may be more complex than a simple inner product – which may not even make sense, depending on the data type.

Organization of This Paper. The rest of this paper is organized as follows. Section 2 summarizes some recent extensions of the Dinur-Nissim results. Section 3.1 describes a natural definition of a privacy-preserving statistical database, held as a desideratum for 29 years, and gives some intuition for why it cannot be achieved. However, just as the negative results of [7] yielded insight into how to

permit accuracy while ensuring privacy by focusing our attention on "reasonable" numbers of queries, the counter-example to the natural definition exhibited flaws in the definition – the wrong question was being asked! The deeper understanding resulted in a new concept, *differential privacy*. This is described in Section 3.2. Finally, a concrete privacy mechanism achieving differential privacy is presented in Section 3.3, and our question about the magnitude of noise sufficient to maintain privacy against T queries is answered.

2 Strengthening the Impossibility Results

We[1] have recently extended the Dinur-Nissim results in several ways summarized in Theorem 1. The proof of Part 1 is the "right" version of Dinur-Nissim: it specifies an explicit set of exactly n queries that always yields blatant non-privacy. Parts 2-4 consider the case in which there may be some small errors but also a constant fraction of the errors may be *unbounded*. The case of unbounded errors with zero small errors is similar to the situation with error-correcting codes, when a symbol is either correct (zero error) or incorrect (no assumptions). We have one result of this type, and several with "mixed" errors.

Theorem 1. *In each case below a query is defined by an n-dimensional vector x, the database is an n-dimensional vector DB, and the true answer is $x \cdot DB$. The response is the true answer plus noise. All the results will hold independent of how the noise is generated, and even if the privacy mechanism knows all questions in advance.*

1. *If the noise is restricted to $o(\sqrt{n})$ in every response, then the system is blatantly non-private against a polynomial time bounded adversary asking exactly n queries $x \in \{\pm 1\}^n$. More generally, a noise bound of α translates to reconstruction of $n - 9\alpha^2$ entries. The attack uses Fourier analysis in a straightforward way.*
2. *Let ρ be any constant less than 0.239. If the noise is* unbounded *on up to a ρ fraction of the responses and restricted to $o(\sqrt{n})$ on the remaining $(1-\rho)$ fraction, then the system is blatantly non-private against a polynomial time bounded adversary asking $\Theta(n)$ queries in $\mathcal{N}(0,1)^n$, that is, each query is a vector of standard normals. More generally, a bound of α on the small noise yields reconstruction in $n - \Theta(\alpha^2)$ entries.*
3. *For any fixed $\delta > 0$, if the noise is unbounded on a $(1/2 - \delta)$ fraction of the queries and restricted to $o(\sqrt{n})$ on the remaining $(1/2 + \delta)$ fraction, then the system is blatantly non-private against*
   (a) *an exponential-time adversary asking only $O(n)$ queries*
   (b) *a polynomial time adversary* against a non-interactive solution (eg, a noisy table) *asking only $O(n)$ questions, where the break is* in the list-decoding sense*; that is, the adversary can produce a constant-sized list*

[1] These results were obtained jointly with Frank McSherry, Kunal Talwar, and Sergey Yekhanin.

of candidate databases containing at least one that agrees with the true database in at least $n - o(n)$ entries.

The queries for both parts of this result are randomly chosen vectors $x \in \{\pm 1\}^n$ and the attack works with overwhelming probability over the choice of queries.

4. *If the noise is unbounded on up to $1/2 - \delta$ of the responses, but is* zero *in the remaining $1/2 + \delta$, then the system is blatantly non-private against a polynomial time bounded adversary making $O(n)$ queries with integer coefficients in the interval $[-c, c]$, where $c = c(\delta)$ is a constant that goes to infinity as δ approaches 0. The attack uses algebraic geometry codes.*

In all but Part 4, if the database has $\Omega(n)$ ones, then $x \cdot DB$ has expected magnitude close to $\sqrt{n}$. Thus, even on the queries on which the system gives "small" error $o(\sqrt{n})$, the magnitude of the error is close to the magnitude of the answer. And still the system is blatantly non-private.

The attack in Theorem 1.2 is inspired by recent results of Donoho [8, 9] and Candes, Rudelson, Tao, and Vershynin [4], in which linear programming is used for compressed sensing and decoding in the presence of errors. Indeed, our query matrices are exactly the ones studied in [4]. Our result is stronger in two ways: we tolerate small noise everywhere, and our proof is more direct, yielding a better decoding bound and a sharp threshold even in the zero small noise case[2].

3 Differential Privacy

3.1 Motivation for the Definition

Development of the notion of differential privacy was guided by a different type of impossibility result than those discussed so far. A classical desideratum for statistical databases was articulated in [5]:

> (Dalenius, 1977) Access to a statistical database should not enable one to learn anything about an individual that could not be learned without access[3].

This goal cannot be achieved when the database has any utility [10]:

> "The obstacle is in *auxiliary information*, that is, information available to the adversary other than from access to the statistical database, and the intuition behind the proof of impossibility is captured by the following example. Suppose one's exact height were considered a highly

[2] In an alternate version of Theorem 1.2 the queries may be randomly chosen vectors in $\{\pm 1\}^n$. Unlike the case with Gaussian queries, this alternate version does not necessarily return the exact database when size of the "small" errors is set to 0 (instead of $o(\sqrt{n})$).

[3] This is analagous to Goldwasser and Micali's definition of *semantic security* against an eavesdropper, which says, roughly, that nothing can be learned about a plaintext from the ciphertext that could not be learned without seeing the ciphertext [15].

> sensitive piece of information, and that revealing the exact height of an individual were a privacy breach. Assume that the database yields the average heights of women of different nationalities. An adversary who has access to the statistical database and the auxiliary information "Terry Gross is two inches shorter than the average Lithuanian woman" learns Terry Gross' height, while anyone learning only the auxiliary information, without access to the average heights, learns relatively little."

As further noted in [10], the impossibility result applies regardless of whether or not Terry Gross is in the database. This led to the following, alternative notion [10, 12]:

> *Differential Privacy:* Access to a statistical database should not enable one to learn anything about an individual *given that her data are in the database* than can be learned when her data are *not* in the database.

While differential privacy does not rule out a bad disclosure, it assures the individual that it will not be the inclusion of her data in the database that causes it, nor could the disclosure be avoided through any action or inaction on the part of the user of the database.

3.2 Formal Definition

The privacy mechanism is a randomized algorithm that takes the database as input and produces an output.

Definition 1. *A randomized function $\mathcal{K}$ gives* ϵ-differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S] \tag{1}$$

A mechanism $\mathcal{K}$ satisfying this definition ensures a participant that even if she removed her data from the data set, no outputs (and thus consequences of outputs) would become significantly more or less likely. For example, if the database were to be consulted by an insurance provider before deciding whether or not to insure Terry Gross, then the presence or absence of Terry Gross in the database will not significantly affect her chance of receiving coverage.

This definition naturally extends to group privacy as well. If the definition is satisfied as written, then the inclusion/exclusion of the data of any c participants yields a factor of $\exp(\epsilon c)$ (instead of $\exp(\epsilon)$), which may be tolerable for small c. Since the *sine qua non* of a statistical database is to teach information about the population as a whole, it is natural, indeed essential, that the privacy bounds deteriorate as group size increases.

3.3 Achieving Differential Privacy

We now describe a concrete interactive privacy mechanism achieving ϵ-differential privacy (see [12] for a full treatment). The mechanism works by adding appropriately chosen random noise to the true answer $a = f(X)$, where f is the *query*

function and X is the database. A helpful example to keep in mind is (a vector of d) queries of the form "How many rows in the database satisfy predicate P?" where the true answer is a vector of d integers (one per query). It is noteworthy that "counting" queries of this type are a very powerful privacy-preserving interface to the database. For example, it is shown in [3] that many popular datamining tasks, including principal component analysis, association rules, k-means clustering, and the ID3 decision tree creation, can be carried out with excellent accuracy while only using a small number of counting queries.

The magnitude of the noise is chosen as a function of the largest change a single participant could have on the output to the query function; we refer to this quantity as the *sensitivity* of the function.

Definition 2. *For $f : \mathcal{D} \to R^d$, the* $L1$-sensitivity *of f is*

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1 \tag{2}$$

for all D_1, D_2 differing in at most one element.

Note that sensitivity is a property of the function alone, and is independent of the database. So we may assume that sensitivity is known to the user. For many types of queries Δf will be quite small. In particular, the counting queries "How many rows have property P?" have $\Delta f = 1$. Our techniques will introduce the least noise when Δf is small.

The privacy mechanism, denoted $\mathcal{K}_f$ for a query function f, computes $f(X)$ and independently adds noise with a scaled symmetric exponential distribution with variance σ^2 (to be determined in Theorem 2) in each component. This distribution is described by the density function

$$\Pr[\mathcal{K}_f(X) = a] \propto \exp(-\|f(X) - a\|_1/\sigma) \tag{3}$$

and the mechanism simply adds, to each coordinate of $f(X)$, independently generated samples of this distribution.

Theorem 2. *[10, 12] For $f : \mathcal{D} \to R^d$, $\mathcal{K}_f$ gives $(\Delta f/\sigma)$-differential privacy.*

Proof. Starting from (3), we apply the triangle inequality within the exponent, yielding for all possible responses r

$$\Pr[\mathcal{K}_f(D_1) = r] \leq \Pr[\mathcal{K}_f(D_2) = r] \times \exp(\|f(D_1) - f(D_2)\|_1/\sigma)\,. \tag{4}$$

The second term in this product is bounded by $\exp(\Delta f/\sigma)$. Thus (1) holds for singleton sets $S = \{a\}$, and the theorem follows by a union bound.

Theorem 2 describes a relationship between Δf, σ, and the privacy differential. To achieve ϵ-differential privacy, it suffices to choose $\sigma \geq \epsilon/\Delta f$. Significantly, the theorem holds regardless of any auxiliary information that may be available to the adversary, and is independent of the computational power of the adversary. Moreover, composition is simple: to handle T adaptively chosen queries of

respective sensitivities $\Delta f_1, \ldots, \Delta f_T$ it suffices to replace Δf with $\sum_{i=1}^{T} \Delta f_i$ in the noise generation procedure[4].

We may now answer our earlier question: What magnitude noise is sufficient to ensure privacy against T queries? The sensitivity of each query in Theorems 1.1, 1.3, and the ± 1 variant of 1.2, is $\Delta f = 1$ (and the sensitivity of a query in Theorem 1.4 is c). The sensitivity of any sequence of T such queries is thus at most $T\Delta f = T$ (or $Tc = O(T)$ for the case of Theorem 1.4), so the answer in all these cases is $O(T/\epsilon)$.

The situation for Theorem 1.2 is a bit different: there is no upper bound on $|\mathcal{N}(0,1)|$, and a sanitizer that rejects Gaussian queries if they exceed any fixed constant in even one coordinate would be unreasonable. A simple-minded approach would be to take $\log^2 n$ to be an upper bound on Δ (and reject any query vector with L_∞ norm exceeding this amount), which yields $T \log^2 n$ as an upper bound on the sensitivity of any sequence of T queries. This yields noise magnitude $O(T \log^2 n/\epsilon)$. However, we can design a solution that does better. We do this for the pedagogic value of exhibiting the tightness of the tradeoff between accuracy (smallness of noise) and privacy.

A series of T queries implicitly defines a $T \times n$ matrix A, where each row of the matrix corresponds to a single inner product query, and the output is the $T \times 1$ matrix given by $A \cdot DB$. To put things in context, Theorem 1.2 discusses blatant non-privacy when $T = \Omega(n)$ and the matrix A is drawn from $\mathcal{N}(0,1)^{T \times n}$; we are now looking at smaller values of T.

The privacy mechanism will use noise calibrated to sensitivity $\Delta = 2T$. It will also impose a sensitivity *budget* of $2T$ on each row of the database, as we now explain. Let x be a query vector. For each $1 \leq i \leq n$ the budget for row i is charged $|x_i|$. More generally, the cost of A to the budget for row i of the database is the L_1 norm of the ith column of A. The privacy mechanism will answer a query unless is would break the budget of even one row in the database, in which case the mechanism will answer no further questions. Note that the budget and the charges against it are all public and are independent of the database, so this stopping condition reveals nothing about the data.

Since the noise is calibrated for sensitivity $2T$ and no sensitivity budget of $2T$ is exceeded, differential privacy is ensured. We claim that for $T \geq \text{polylog}(n)$, with overwhelming probability over choice of A, the privacy mechanism will answer all T questions before shutting down. Note that A contains nT standard normals, and so with overwhelming probability the maximum magnitude of any entry will not exceed, say, $\log^2 nT$. In the sequel we assume we are in this high probability case.

Consider random variables $X_1, \ldots, X_T$, each in $[0, \log^2 nT]$. Let $S = \sum_{i=1}^{T} X_i$. Hoeffding's inequality says that

$$\Pr[S - E[S] \geq tT] \leq \exp\left(-\frac{2T^2 t}{\sum_{i=1}^{T} \log^4 nT}\right)$$

[4] There are compelling examples in which it is possible to do much better. The interested reader is referred to [12].

We may use this as follows. Since a_{ij} is distributed according to a standard normal, its expected magnitude is $\sqrt{2/\pi}$. Consider a column j of A, and let $X_i = |a_{ij}|$ for $I = 1, \ldots, T$. By linearity of expectation, $E[S] = TE[|\mathcal{N}(0,1)|]$. So Hoeffding's bound says that

$$\Pr[S - T(\sqrt{2/\pi}) \geq tT] \leq \exp\left(-\frac{2T^2t}{\sum_{i=1}^{T} \log^4 nT}\right) = \exp\left(-\frac{2Tt}{\log^4 nT}\right)$$

In particular when $T \geq \log^6(Tn)$ this is negligible for all $t \in \Omega(1)$. By a union bound we see that, as desired, the probability that even one of the n per-row budgets is exceeded is negligible in n.

The bottom line is that, even in the setting of Theorem 1.2, noise of magnitude $O(T/\epsilon)$ is sufficient to ensure privacy against T queries.

We remark that a "better" answer appears in the literature [7, 13, 3]. This is obtained using a slightly weaker, but also reasonable, definition of privacy, in which, roughly speaking, the mechanism is permitted to fail to deliver full ϵ-differential privacy with some small probability δ. Under this relaxed definition one may employ Gaussian noise rather than symmetric exponential noise. This leads to noise of magnitude $\Omega((\sqrt{\log 1/\delta})\sqrt{T}/\epsilon)$. We prefer the exponential noise because it "behaves better" under composition and because the guarantee is absolute ($\delta = 0$).

4 Final Remarks

A Trusted Center. Throughout this paper we have assumed that the data collector and the privacy mechanism are trustworthy. Thus, we are making the problem as easy as possible, yielding stronger lower bounds and impossibility results. The literature also studies the setting in which the data contributors do not trust the data collector to maintain privacy and so first randomize their own data [14, 2]. Of course, since randomized response is a non-interactive mechanism it is subject to the negative conciseness result of [12] mentioned in the Introduction.

"Our Data, Ourselves". A different tack was taken in [11], where, using cryptographic techniques for *secure function evaluation*, the data collector/protector is replaced by a distributed privacy mechanism.

When Noise Makes no Sense. McSherry and Talwar have initiated an exciting investigation of differential privacy in cases in which adding noise may not make sense; for example, the output of a "query," or in general of any operation on a set of private inputs, may not be a number. Given an input vector x (playing the role of a database) and a possible outcome y, assume there is a real-valued *utility* function $u(x, y)$ that evaluates the outcome y for the input set x. As an example, x could be bids for a digital good, y could be a price, and $u(x, y)$ could be the resulting revenue. This has resulted in the design of approximately-truthful and collusion-resistant mechanisms with near-optimal revenue. More generally, y can be a classifer, an expert, or a heuristic.

References

[1] N. R. Adam and J. C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys* 21(4): 515-556 (1989).

[2] R. Agrawal, R. Srikant, D. Thomas. Privacy Preserving OLAP. *Proceedings of SIGMOD 2005.*

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138, 2005.

[4] E. J. Candes, M. Rudelson, T. Tao, and R. Vershynin, Error Correction via Linear Programming. In *Proceedings of the 46th IEEE Annual Symposium on Foundations of Computer Science,* 2005.

[5] T. Dalenius, Towards a methodology for statistical disclosure control. *Statistik Tidskrift 15*, pp. 429–222, 1977.

[6] D. E. Denning, *Secure statistical databases with random sample queries*, ACM Transactions on Database Systems, 5(3):291–315, September 1980.

[7] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[8] D. Donoho. For Most Large Underdetermined Systems of Linear Equations, the minimal l1-norm solution is also the sparsest solution. *Manuscript*, 2004. Available at `http://stat.stanford.edu/~donoho/reports.html`

[9] D. Donoho. For Most Large Underdetermined Systems of Linear Equations, the minimal l1-norm near-solution approximates the sparsest near-solution. *Manuscript*, 2004. Available at `http://stat.stanford.edu/~donoho/reports.html`

[10] C. Dwork. Differential Privacy. *Invited Paper; Proceedings of ICALP 2006.*

[11] C. Dwork, K. Kenthapadi, F. McSherry , I. Mironov and M. Naor. Our Data, Ourselves: Privacy via Distributed Noise Generation. *Proceedings of Eurocrypt 2006*

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[13] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology: Proceedings of Crypto*, pages 528–544, 2004.

[14] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, June 2003.

[15] S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences 28*, pp. 270–299, 1984; prelminary version appeared in *Proceedings 14th Annual ACM Symposium on Theory of Computing*, 1982.