Precisely Answering Multi-dimensional Range Queries without Privacy Breaches[*]

Lingyu Wang, Yingjiu Li[**], Duminda Wijesekera, and Sushil Jajodia

Center for Secure Information Systems, George Mason University
Fairfax, VA 22030-4444, USA
{lwang3,yli2,dwijesek,jajodia}@gmu.edu

Abstract. This paper studies the privacy breaches caused by multi-dimensional range (MDR) sum queries in online analytical processing (OLAP) systems. We show that existing inference control methods are generally infeasible for controlling MDR queries. We then consider restricting users to even MDR queries (that is, the MDR queries involving even numbers of data values). We show that the collection of such even MDR queries is safe if and only if a special set of sum-two queries (that is, queries involving exactly two values) is safe. On the basis of this result, we give an efficient method to decide the safety of even MDR queries. Besides safe even MDR queries we show that any odd MDR query is unsafe. Moreover, any such odd MDR query is different from the union of some even MDR queries by only one tuple. We also extend those results to the safe subsets of unsafe even MDR queries.

1 Introduction

Multi-dimensional range (MDR) query is an important class of decision support query in online analytical processing (OLAP) systems [19]. One of the most popular data models of OLAP systems, data cube [18], can be viewed as a special collection of MDR queries. MDR queries are intended to assist users in exploring trends and patterns in large amounts of data stored in data warehouses. Contrary to this initial objective, MDR queries can be used to obtain protected sensitive data, which results in the breach of an individual's privacy. Access control alone is insufficient in controlling information disclosure, because information not released directly may be inferred indirectly from the answers to legitimate queries, which is known as the *inference problem* in databases. Providing precise answers to MDR queries without privacy breaches is the subject of this paper.

The inference problem has been investigated since the 1970's with many inference control methods proposed, especially for statistical databases. Those methods usually have run times proportional to the size of the queries or the data sets, and they are invoked only after queries have arrived. On the other hand, OLAP applications demand instant

[*] This work was partially supported by the National Science Foundation under grant CCR-0113515.

[**] The author's current address is School of Information Systems, Singapore Management University, Singapore 259756

E. Snekkenes and D. Gollmann (Eds.): ESORICS 2003, LNCS 2808, pp. 100–115, 2003.

Table 1. An Example of a Two-dimensional Data Core.

The Data Core $\boldsymbol{year\_emp\_adj}$

year / emp / adj	Alice	Bob	Mary	Jim
2002	1000.00	500.00	-2000.00	
2003		1500.00	-500.00	1000.00

responses to queries, although the queries usually aggregate a large amount of data. Consequently, the delay in query answering renders most existing methods impractical for OLAP systems. In this paper we propose efficient inference control methods by exploiting the unique structures of MDR queries.

The first contribution of this paper is that it will call more attention to the privacy issue of OLAP systems, which is unfortunately ignored in most of today's commercial products. We study several existing inference methods and the results show that they are infeasible for MDR queries. We also show that finding maximal safe subsets of unsafe MDR queries is NP-hard. Second, we reduce the inference control of MDR queries to that of sum-two queries with a necessary and sufficient condition on their compromisability. By treating sum-two queries as edges of simple undirected graphs, this reduction relates the inference control of MDR queries with existing results in inference control in statistical databases and graph theory. Finally, we give efficient methods (the complexity is bounded by $O(mn)$, where m, n are the number of queries and tuples, respectively) to determine safe MDR queries, safe arbitrary queries, and large subsets of unsafe MDR queries.

The rest of the paper is organized as follows. Section 1.1 gives a motivating example, and Section 1.2 describes our assumptions. Section 2 reviews existing inference control methods proposed in traditional statistical databases and modern decision support systems. Section 3 presents basic definitions and formalizes MDR queries and the compromisability. Section 4 gives negative results of applying existing inference control methods to MDR queries. Section 5 investigates the problem of determining safe MDR queries. Section 6 extends the results to subsets of unsafe MDR queries. Section 7 discusses the implementation. Section 8 concludes the paper. Due to space limitations, we have omitted the proofs of all theorems, lemmata, corollaries, and propositions, which can be found in [31].

1.1 Motivating Example

Suppose that part of a data set owned by a fictitious organization, *Company A*, is shown in Table 1. It contains salary adjustments for four employees in years 2002 and 2003. Let the three attributes be $year$, emp (employee), and adj (adjustment), respectively. The empty cells in Table 1 indicate that the employee did not work for *Company A* in that year.

Company A invites an analyst *Mallory* to analyze the data set. For this purpose, *Mallory* is allowed to ask sum queries about the attribute adj in Table 1. In addition, she has access to the non-sensitive attributes $year$, emp as well as the locations of empty cells in Table 1. On the other hand, *Company A* worries that *Mallory* may inappropriately use the information she learns about each employee. Hence, *Mallory* is prohibited from directly

Table 2. An Example of Even MDR Queries.

Ranges	Answer
$[(Alice, 2002), (Jim, 2003)]$	1500
$[(Alice, 2002), (Bob, 2002)]$	1500
$[(Bob, 2002), (Mary, 2002)]$	-1500
$[(Bob, 2002), (Bob, 2003)]$	2000
$[(Mary, 2003), (Jim, 2003)]$	500

asking the individual values (of attribute adj) in Table 1. Now we ask the following Questions: *Can Mallory learn any of the individual values through sum queries?* If so, *how can we safeguard these values?* Suppose *Mallory* asks the following query:

```
SELECT emp, SUM(adj)
FROM year_emp_adj
GROUP BY emp;
```

The answer to the above query contains four records $(Alice, 1000)$, $(Bob, 2000)$, $(Mary, -2500)$ and $(Jim, 1000)$. Each record corresponds to a one-dimensional MDR sum query, such as $(Alice, 1000)$, which sums the values in the first column of the table. Intuitively, by viewing each MDR query as a *box*, we can represent it using its longest *diagonal*. For example, use $[(Alice, 2002), (Alice, 2003)]$ for the first column of the table and $[(Alice, 2002), (Bob, 2003)]$ for the first two columns. For simplicity purpose, we shall use this notation instead of SQL for MDR query henceforth.

Mallory is able to learn from the MDR query $[(Alice, 2002), (Alice, 2003)]$ that the adjustment for Alice in 2002 is 1000.00, because the query sums a single value. This threat can be thwarted by answering only the MDR queries that sum two or more values. However, *Mallory* can easily get around this restriction by subtracting (the answers to) $[(Bob, 2002), (Mary, 2002)]$ from $[(Alice, 2002), (Mary, 2002)]$.

This second inference occurs because the queries $[(Bob, 2002), (Mary, 2002)]$ and $[(Alice, 2002), (Mary, 2002)]$ sum even (two) and odd (three) number of values, respectively. Is it helpful for protecting the individual values to restrict *Mallory* to only *even MDR queries* (MDR queries involving even number of values) or only *odd MDR queries* (MDR queries involving odd number of values)? The restriction to odd MDR queries is ineffective because the difference of two odd numbers yields an even number. For example, the first two and three columns of Table 1 are both odd, but their difference gives the third column, which is even. Conversely, to obtain odd MDR queries from even ones is not always straightforward. Because an individual value can also be viewed as an odd MDR query, restricting users to even MDR queries makes inferences substantially more difficult.

Nonetheless, inference is still possible with only even MDR queries. A series of five even MDR queries asked by *Mallory* and their answers are given in Table 2. The first query sums all six values and the remaining four queries each sum two values. *Mallory* then adds the answers to the last four queries (2500) and subtracts from the result the answer to the first queries (1500). Dividing the result of the subtraction (1000) by two gives Bob's adjustment in 2002 (500).

In the rest of this paper we address the following questions naturally motivated by the above example: *1. How can we efficiently determine whether even MDR queries are* *safe? 2. What is the impact on users if only even MDR queries are allowed? 3. Besides the even MDR queries, what else can be answered safely? 4. If even MDR queries are unsafe, can we find a large safe subset?*

1.2 Assumptions

We only consider *stateless* inference control methods. That is, the methods that grant or deny incoming queries independent of the queries previously asked by the user. For example, restrictions on the size or parity of queries are stateless. On the other hand, the *stateful* methods base authorization decisions on the history of queries asked by a specific user, for example, controlling the size of overlaps between queries. Stateful restrictions are usually infeasible in practice, because users can subvert them by using aliases for login or colluding.

We assume users do not possess the *external knowledge*[1] about the boundaries of protected individual values. Consequently, we consider the protected values as unbounded reals. Under that assumption, it is relevant for inference control to know which values users know and which they do not, but the specific values are irrelevant. For example, all the inferences we discuss in Section 1.1 are possible regardless of the explicit values we put in Table 1. Inference of approximate values caused by external knowledge about boundaries or data types has been studied in [22, 24]. Their inference control methods can be incorporated into our methods as post-processing, because the inferences we study require less external knowledge and should be checked first.

On the other hand, we assume users may know some of the protected values from external knowledge. For example, in Table 1 users know *Alice* does not have a valid salary adjustment in year 2003 because she has left *Company A* by the end of 2002. Regardless of the specific sources of external knowledge, we shall treat all known values as empty cells. We do not consider the known values of which the inference control mechanism is not aware (undetected external knowledge). Under this assumption, the summation of any two real unbounded values is considered safe. We address the issue of undetected external knowledge in Section 7.

2 Related Work

Inference control has been extensively studied in statistical databases [12, 1, 14] and the proposed methods are usually classified into two categories: *restriction-based* techniques and *perturbation-based* techniques. Restriction-based techniques include restricting the size of *query sets* (i.e., the tuples that satisfy a single query) [17], restricting the size of overlaps [15] between query sets, detecting inferences by auditing all queries asked by a specific user [10, 8, 20, 5], suppressing sensitive data in released statistical tables [11], and grouping tuples and treating each group as a single tuple [9, 25]. Perturbation-based techniques add noise to source data or outputs [28, 4, 27]. Other aspects of the inference problem include the inference caused by arithmetic constraints [6], inferring approximate values instead of exact values [24] and inferring intervals enclosing exact values [22, 21, 23]. The inference control methods proposed for statistical databases do

[1] The knowledge obtained from sources other than queries [12].

not consider the unique structure of MDR queries. This renders them ineffective and inefficient for MDR queries. We show some examples in Section 4.

Recently a variation of the inference control problem, namely, *privacy preserving data mining*, has drawn considerable attention as seen in [2, 26]. They all attempt to perturb sensitive values while preserving the classifications or association rules that can be learned from the data set. In doing so, they assume that a user's objective of data analysis is predictable. However, in OLAP systems this assumption may not hold, because we do not know in advance what users may want to discover. Our work does not have this limitation, because what we give users is not the results (e.g., classifications or association rules), but the means (the precise answers to their queries) to obtain the results they desire.

Controlling inferences of a special class of MDR queries, namely, *data cubes*, is studied in [29]. They give sufficient conditions for safe data cubes based on the cardinality of the data core. A data core is safe if it is full or dense (the number of known values is either zero or under the given bound). Note that this condition does not apply to those MDR queries that are not included in the data cube. This paper strengthens this result by giving necessary and sufficient conditions for all MDR queries.

The inference problem of one-dimensional range queries is studied in [8], and the MDR case is considered difficult. The *usability* (i.e., the highest possible ratio of the number of safe queries to that of all queries) of MDR queries in the full core is studied in [5]. They mention but do not fully explore the restriction of even MDR queries. However, the general case with known values (referred to as *holes* in [5]) is thought to be challenging. In [7, 10] Chin gives necessary and sufficient condition for the compromisability of sum-two queries. He also proves that finding the maximal safe subsets of unsafe sum-two queries is NP-hard. However, sum-two queries are rare in practice. In this paper we use his results by reducing the compromisability of even MDR queries to that of sum-two queries.

3 Basic Definitions

This section defines the basic concepts and notations. We use $\mathbb{I}, \mathbb{R}, \mathbb{I}^k, \mathbb{R}^k, \mathbb{R}^{m \times n}$ to denote the set of integers, reals, k-dimensional integer vectors, k-dimensional real vectors and m by n real matrices, respectively. For any $u, v, t \in \mathbb{R}^k$, we write $u \leq v$ and $t \in [u, v]$ to mean that $u[i] \leq v[i]$ and $min\{u[i], v[i]\} \leq t[i] \leq max\{u[i], v[i]\}$ for all $1 \leq i \leq k$, respectively. We use t for the singleton set $\{t\}$ whenever clear from the context.

Definition 1 (Core).
For any $d \in \mathbb{I}^k$, use $\mathcal{F}(d)$ to denote the Cartesian product $\Pi_{i=1}^{k}[1, d[i]]$. We say $F = \mathcal{F}(d)$ is the full core. Any $C \subseteq F$ is a core. Any $t \in F$ is a tuple. Any $t \in F \setminus C$ is a tuple missing from C.

Definition 1 formalizes the concepts of *full core*, *core*, and *tuple*. The full core is formed by the Cartesian product of closed integer intervals. A core is any subset of the full core. A tuple is any vector in the full core and a tuple missing from the core is any vector in the complement of the core with respect to the full core.

Definition 2 (MDR Query, Sum-two Query and Arbitrary Query). *Given any full core F and core $C \subseteq F$,*

1. *Define functions*
   (a) $q^\star(.) : F \times F \to 2^C$ as $q^\star(u,v) = \{t : t \in C, t \in [u,v]\}$.
   (b) $q^2(.) : C \times C \to 2^C$ as $q^2(u,v) = \{u,v\}$ if $u \neq v$, and ϕ otherwise.
2. *Use $\mathcal{Q}_d(C)$ and $\mathcal{Q}_t(C)$ (or simply $\mathcal{Q}_d$ and $\mathcal{Q}_t$ when C is clear from context) for $\{q^\star(u,v) : q^\star(u,v) \neq \phi\}$ and $\{q^2(u,v) : q^2(u,v) \neq \phi\}$, respectively.*
3. *We call any non-empty $q \subseteq C$ an arbitrary query, any $q^\star(u,v) \in \mathcal{Q}_d$ an MDR query (or simply query), and any $q^2(u,v) \in \mathcal{Q}_t$ a sum-two query.*

In Definition 2 we formalize the concepts of *arbitrary query*, *MDR query*, and *sum-two query*. An arbitrary query is any non-empty subset of the given core. An MDR query $q^\star(u,v)$ is a non-empty subset of the core that includes all and only those tuples *bounded* by two given tuples. Intuitively, an MDR query can be viewed as a multi-dimensional axis-parallel box. A sum-two query is any set of exactly two tuples. We use $\mathcal{Q}_d$ and $\mathcal{Q}_t$ for the set of all MDR queries and all sum-two queries, respectively.

Definition 3 (Compromisability). *Given any full core F, core $C \subseteq F$, and any set of arbitrary queries $\mathcal{S}$, use $\mathcal{M}(\mathcal{S})$ to denote the incidence matrix[2] of the set system formed by C and $\mathcal{S}$, we say that*

1. *$\mathcal{S}_1$ is derivable from $\mathcal{S}_2$, denoted as $\mathcal{S}_1 \preceq_d \mathcal{S}_2$, if there exists $M \in \mathbb{R}^{|\mathcal{S}_1| \times |\mathcal{S}_2|}$ such that $\mathcal{M}(\mathcal{S}_1) = M \cdot \mathcal{M}(\mathcal{S}_2)$ holds, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are sets of arbitrary queries.*
2. *$\mathcal{S}_1$ compromises $t \in C$ if $t \preceq_d \mathcal{S}_1$ (we write t for $\{\{t\}\}$), and $\mathcal{S}_1$ is safe if it compromises no $t \in C$.*
3. *$\mathcal{S}_1$ is equivalent to $\mathcal{S}_2$, denoted as $\mathcal{S}_1 \equiv_d \mathcal{S}_2$, if $\mathcal{S}_1 \preceq_d \mathcal{S}_2$ and $\mathcal{S}_2 \preceq_d \mathcal{S}_1$.*

Definition 3 formalizes the concept of compromisability and related concepts. Because an arbitrary query is a set of tuples, any given set of arbitrary queries can be characterized by the incidence matrix of the set system formed by the core and the set of arbitrary queries. Given two sets of arbitrary queries $\mathcal{S}_1$, $\mathcal{S}_2$, and the incidence matrices $\mathcal{M}(\mathcal{S}_1)$, $\mathcal{M}(\mathcal{S}_2)$, we say $\mathcal{S}_1$ is derivable from $\mathcal{S}_2$ if the row vectors of $\mathcal{M}(\mathcal{S}_1)$ can be represented as the linear combination of those of $\mathcal{M}(\mathcal{S}_2)$. Intuitively, this implies that the information disclosed through $\mathcal{S}_1$ can be computed from that through $\mathcal{S}_2$. We say $\mathcal{S}_1$ compromises a tuple t in the core if the set of queries $\{\{t\}\}$ (notice $\{t\}$ is an MDR query) is derivable from $\mathcal{S}_1$, and $\mathcal{S}_1$ is safe if it compromises no tuple in the core. We say any two sets of arbitrary queries are equivalent if they are mutually derivable.

Example 1. Table 3 gives an example of the core, MDR queries, and compromisability. As shown in the left upper table in Table 3, the core C contains six tuples. The subscripts of the tuples give their order. The right upper table shows a set of five MDR queries S. The lower equation shows that S compromises $(1,2)$. The left side of the equation is

[2] $\mathcal{M}(\mathcal{S})[i,j] = 1$ if the i^{th} arbitrary query in $\mathcal{S}$ contains the j^{th} tuple in C, and $\mathcal{M}(\mathcal{S})[i,j] = 0$ otherwise.

Table 3. An Example of Core, MDR Queries, and Compromisability.

The Core C

	1	2	3	4
1	$(1,1)_1$	$(1,2)_2$	$(1,3)_3$	
2		$(2,2)_4$	$(2,3)_5$	$(2,4)_6$

A Set of MDR Queries:$\mathcal{S}$

$q^\star((1,1),(2,4))$	$\{(1,1),(1,2),(1,3),$ $(2,2),(2,3),(2,4)\}$
$q^\star((1,1),(1,2))$	$\{(1,1),(1,2)\}$
$q^\star((1,2),(1,3))$	$\{(1,2),(1,3)\}$
$q^\star((1,2),(2,2))$	$\{(1,2),(2,2)\}$
$q^\star((2,3),(2,4))$	$\{(2,3),(2,4)\}$

$\mathbf{(1,2)} \preceq_d \boldsymbol{\mathcal{S}}$ **because**

$$\mathbf{[0,1,0,0,0,0] = [-\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2}]}\cdot \begin{pmatrix} 1\,1\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0 \\ 0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix}$$

the incidence matrix of $(1,2)$, and the right side gives a linear combination of the row vectors in the incidence matrix of S. Table 3 characterizes exactly the same example given in Tables 1 and 2, except that it uses the concepts and notations we defined in this section.

The relation $\equiv_d$ of Definition 3 is an equivalence relation on the family of all sets of arbitrary queries, because it is reflexive, symmetric, and transitive. Hence if any two sets of arbitrary queries are equivalent, then one is safe iff the other is. In Section 5 we shall reduce the compromisability of even MDR queries to that of a special set of sum-two queries based on this fact.

4 Ineffective or Infeasible Restrictions

In this section we apply several existing restriction-based inference control methods to MDR queries. Our results show that they are ineffective or infeasible for MDR queries. We first investigate three methods, *Query set size control*, *overlap size control*, and *Audit Expert* in Section 4.1. Then we consider the problem of finding maximal safe subsets of unsafe MDR queries in Section 4.2.

4.1 Query Set Size Control, Overlap Size Control, and Audit Expert

Query Set Size Control. This method prohibits users from asking *small* queries whose cardinalities are smaller than some pre-determined threshold n_t [17]. For arbitrary queries, query set size control can be easily subverted by asking two legitimate queries whose intersection yields a prohibited one, a mechanism known as the *tracker* in statistical databases [13]. It is shown that finding a tracker for arbitrary queries is possible even when n_t is about half of the cardinality of the core. At first glance, trackers may seem to be more difficult to find when users are restricted to MDR queries. However, in [30] we show that when n_t is not big enough ($n_t \leq \frac{n}{3^k}$) a tracker can always be found to derive *any* given small MDR query, and the tracker consists of only MDR queries.

Overlap Size Control. This method prevents users from asking queries with large intersections [15]. Any answerable query must have a cardinality of at least n, and the intersection of any two queries is required to be no larger than r. In order to compromise any tuple t, one must first ask one query $q \ni t$ and subsequently $(n-1)/r$ or more queries to form the complement of t with respect to q. Consequently, no inference is possible if less than $(n-1)/r+1$ queries are answered. This bound is not improved (increased) by restricting users to MDR queries, because for almost any MDR query the complement of a tuple can always be formed. Overlap size control is infeasible because it is a stateful method. Moreover, it depends on the restriction of small queries, which is ineffective as described above.

Audit Expert. Chin gives a necessary and sufficient condition for determining safe arbitrary queries in Audit Expert [10]. By treating tuples and queries as a set system, the queries are safe iff the incidence matrix of the set system contains one or more unit row vectors in its reduced row echelon form (RREF). The elementary row transformation used to obtain the RREF of an m by n matrix has the complexity $O(m^2n)$. Using this condition *online* (after queries arrive) may incur unacceptable delay in answering queries because m and n can be very large in OLAP systems. Moreover, it is a stateful method because it requires the entire history of queries. A better way to use the condition is to determine the compromisability of queries off-line [5]. However, although this condition certainly applies to MDR queries, it is not efficient because it does not take into consideration the inherent redundancy among MDR queries. In Section 5 we further investigate this issue in detail.

4.2 Finding Maximal Safe Subsets of Unsafe MDR Queries

When a set of queries is not safe, it is desired to find its maximal safe subset. In [10] it has been shown that finding the maximal safe subset of unsafe arbitrary queries (the MQ problem) or sum-two queries (the RMQ problem) is NP-hard. A natural question is whether restricting users to MDR queries makes the problem easier. Unfortunately, this is not the case. We show that this problem remains NP-hard even when restricted to MDR queries (the MDQ problem). The result is based on the intuition that given any core C_0 and any set of sum-two queries $S_0 \subseteq \mathcal{Q}_t(C_0)$, we can find another core C_1 and a set of MDR queries $S_1 \subseteq \mathcal{Q}_d(C_1)$, such that the maximal safe subset of S_1 gives the maximal safe subset of S_0 in polynomial time. Consequently, MDQ problem is also NP-hard.

Theorem 1. *The MDQ problem is NP-hard.*

Restricted MDQ Problem. Knowing that the MDQ problem is NP-hard, is it possible to reduce the complexity with further restrictions? We consider *data cubes*, a special class of MDR queries originally defined in [18]. We illustrate some of the concepts of data cubes in Example 2. The following Corollary 1 shows that the MDQ problem remains NP-hard even when it is restricted to those special MDR queries.

Example 2. In Table 3, the two *1-star cuboids* are $\{q^\star((1,1),(1,4)), q^\star((2,1),(2,4))\}$ and $\{q^\star((1,1),(2,1)), q^\star((1,2),(2,2)), q^\star((1,3),(2,3)), q^\star((1,4),(2,4))\}$. The only

Table 4. An Example Showing $\mathcal{Q}_e$ Not Equivalent to $\mathcal{Q}_e \cap \mathcal{Q}_t$ using the same C in Table 3.

$\mathcal{Q}_e$	$q^\star((1,1),(1,2)), q^\star((1,2),(1,3)), q^\star((2,2),(2,3)), q^\star((2,3),(2,4))$ $q^\star((1,2),(2,2)), q^\star((1,3),(2,3)), q^\star((1,2),(2,3)), q^\star((1,1),(2,4))$
$\mathcal{Q}_e \cap \mathcal{Q}_t$	$\mathcal{Q}_e \setminus \{q^\star((1,2),(2,3))\} \cup \{q^\star((1,1),(2,4))\}$

$$\boldsymbol{q^\star((1,1),(2,4)) \not\preceq_d \mathcal{Q}_e \cap \mathcal{Q}_t}$$

2-star cuboid is a singleton set $\{q^\star((1,1),(2,4))\}$. The *data cube* is the union of the three cuboids, which also includes all *skeleton queries*.

Corollary 1. *The problem MDQ remains NP-hard under the restriction that the given set of MDR queries must be: 1. a set of skeleton queries; 2. a union of some cuboids; or 3. a data cube.*

5 Compromisability of Even MDR Queries

This section investigates the compromisability of *even MDR queries* (that is, MDR queries involving even number of tuples). First, in Section 5.1 we show that the set of even MDR queries is equivalent to a subset of sum-two queries (that is, sets of two tuples). Based on this equivalence, the compromisability of even MDR queries can be efficiently determined. In Section 5.2 we show that answering any odd MDR query in addition to even MDR queries leads to compromises, and any odd MDR query is different from the union of a few even MDR queries by only one tuple. We also show that the compromisability of arbitrary queries can be efficiently determined given that the even MDR queries are safe.

5.1 Equivalence between MDR Queries and Sum-Two Queries

Denote the set of all even MDR queries as $\mathcal{Q}_e$. To efficiently determine the compromisability of the even MDR queries $\mathcal{Q}_e$, we show that there exists a subset $\mathcal{Q}_{dt}$ of sum-two queries $\mathcal{Q}_t$, such that $\mathcal{Q}_{dt} \equiv_d \mathcal{Q}_e$. Then we can determine whether $\mathcal{Q}_e$ is safe by checking if $\mathcal{Q}_{dt}$ is safe. Intuitively, determining the compromisability of $\mathcal{Q}_{dt}$ is easier because by reducing $\mathcal{Q}_e$ to $\mathcal{Q}_{dt}$ we have removed most redundant queries.

Two natural but untrue conjectures are $\mathcal{Q}_e \equiv_d \mathcal{Q}_t$ and $\mathcal{Q}_e \equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$. To see why $\mathcal{Q}_e \equiv_d \mathcal{Q}_t$ is untrue, consider the counter-example with the one-dimensional core $C = \{1, 2, 3\}$. We have that $q^2(1,3) \in \mathcal{Q}_t$ is not derivable from $\mathcal{Q}_e = \{q^\star(1,2), q^\star(2,3)\}$. Example 3 gives a counter-example to $\mathcal{Q}_e \equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$.

Example 3. Table 4 shows $\mathcal{Q}_e \not\preceq_d \mathcal{Q}_e \cap \mathcal{Q}_t$ because $q^\star((1,1),(2,4)) \in \mathcal{Q}_e$ is not derivable from $\mathcal{Q}_e \cap \mathcal{Q}_t$.

From Example 3 we see that $\mathcal{Q}_e \not\preceq_d \mathcal{Q}_e \cap \mathcal{Q}_t$ because of even queries such as $q^\star((1,1),(2,4))$. Such an even query is the union of *odd queries* like $q^\star((1,1),(1,3))$ and $q^\star((2,2),(2,4))$. Intuitively, suppose that from $\mathcal{Q}_e \cap \mathcal{Q}_t$ we can derive each odd query up to the *last tuple*. Then we *pair* the adjacent last tuples of all the odd queries by adding other sum-two queries to $\mathcal{Q}_e \cap \mathcal{Q}_t$. Hence, we can derive the even query with these additional sum-two queries. Conversely, these additional sum-two queries can be derived from $\mathcal{Q}_e$ by reversing this process. We demonstrate this in Example 4 and generalize the result in Theorem 2.

Example 4. In Example 3, we can let $\mathcal{Q}_{dt} = \mathcal{Q}_e \cap \mathcal{Q}_t \cup \{q^2((1,3),(2,4))\}$. Consequently, we derive $q^\star((1,1),(2,4))$ as the union of $q^2((1,1),(1,2))$, $q^2((2,2),(2,3))$ and $q^2((1,3),(2,4))$. Conversely, $q^2((1,3),(2,4)$ can be derived as $q^\star((1,1),(2,4)) \setminus (q^2((1,1),(1,2)) \cup q^2((2,2),(2,3)))$. Hence, now we have $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$.

Theorem 2. *For any core C, there exists $\mathcal{Q}_{dt} \subseteq \mathcal{Q}_t$ such that $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$ holds.*

The proof of Theorem 2 [30] includes a procedure that constructs $\mathcal{Q}_{dt}$ by calling a subroutine *Sub_QDT* for each even MDR query $q^\star(u_0, v_0)$. *Sub_QDT* adopts a divide-and-conquer approach in pairing the tuples in $q^\star(u_0, v_0)$. Intuitively, we view each MDR query as an axis-parallel box. At the first stage, *Sub_QDT* recursively divides the current j-dimensional box into $(j-1)$-dimensional boxes, until single tuples are returned as zero-dimensional boxes. Then at the second stage, suppose the current box $q^\star(u,v)$ is j-dimensional; *Sub_QDT* pairs every two tuples returned by the $(j-1)$-dimensional boxes (that $q^\star(u,v)$ has been divided into). If $q^\star(u,v)$ contains even number of tuples, then all of them can be properly paired and *null* is returned to the $(j+1)$-dimensional box. Otherwise, the returned tuple t from the last $(j-1)$-dimensional box cannot be paired and is returned by $q^\star(u,v)$.

Graph Representation and Complexity Analysis. The time complexity of building $\mathcal{Q}_{dt}$ using *Sub_QDT* is $O(mn)$, where $m = \mid \mathcal{Q}_e \mid$ and $n = \mid C \mid$. Because $\mid \mathcal{Q}_{dt} \mid \leq \mid \mathcal{Q}_t \mid \leq \binom{|C|}{2}$ and $m = O(\binom{|C|}{2})$, we have $\mid \mathcal{Q}_{dt} \mid = O(m)$. Hence, no more storage is required by $\mathcal{Q}_{dt}$ than by $\mathcal{Q}_e$.

For any $S \subseteq \mathcal{Q}_{dt}$, we use $G(C,S)$ for the undirected simple graph having C as the vertex set, S as the edge set, and each edge $q^2(t_1,t_2)$ incident the vertices t_1 and t_2. We call $G(C,\mathcal{Q}_{dt})$ the *QDT Graph*. It has been shown in [7] that a set of sum-two queries is safe iff the corresponding graph is a bipartite graph (that is, a graph with no cycle containing an odd number of edges). This can easily be decided with a breadth-first search (BFS) on $G(C,\mathcal{Q}_{dt})$, taking time $O(n+ \mid \mathcal{Q}_{dt} \mid) = O(m+n)$. Hence, the complexity of determining the compromisability of $\mathcal{Q}_e$ is dominated by the construction of $\mathcal{Q}_{dt}$, which is $O(mn)$. Notice that from Section 4 we know that directly applying the condition of Audit Expert [10] has the complexity of $O(m^2 n)$. Therefore, our solution is more efficient than Audit Expert with respect to MDR queries.

Example 5. Example 3 has the cycle composed of $q^2((1,3),(2,3))$, $q^2((2,3),(2,4))$, and $q^2((1,3),(2,4))$ in G_{dt}. Hence, G_{dt} is not a bipartite graph and $\mathcal{Q}_{dt}$ (and hence $\mathcal{Q}_e$) is not safe.

5.2 Beyond Even MDR Queries

Characterizing the QDT Graph. We give some properties of the QDT graph in Lemma 1 that are useful for the rest of this section. The first property shown in Lemma 1 is straightforward. The second property is based on the intuition that if any two tuples t_1,

t_2 in the core are not *close enough* (i.e., $q^\star(t_1,t_2) \notin \mathcal{Q}_{dt}$), then we can find another tuple $t_3 \in q^\star(t_1,t_2)$, such that $q^\star(t_1,t_2) \in \mathcal{Q}_{dt}$ and t_3 is closer to t_1 than t_2. If $q^\star(t_1,t_3) \notin \mathcal{Q}_{dt}$, we repeat this process. This process can be repeated less than $\mid q^\star(t_1,t_2) \mid$ times, and upon termination we have a tuple that is close enough to t_1. The third claim is a natural extension of the first two.

Lemma 1. *1.* $\mathcal{Q}_e \cap \mathcal{Q}_t \subseteq \mathcal{Q}_{dt}$.
2. For any $t_1, t_2 \in C$ *satisfying that* $\mid q^\star(t_1,t_2) \mid > 2$, *there exists* $t_3 \in q^\star(t_1,t_2)$ *such that* $q^\star(t_1,t_3) \in \mathcal{Q}_{dt}$.
3. $G(C,\mathcal{Q}_{dt})$ *is connected.*

Properties of $\boldsymbol{\mathcal{Q}_{dt}}$. Although we have shown that $\mathcal{Q}_{dt} \equiv_d \mathcal{Q}_e$, $\mathcal{Q}_{dt}$ may not be the smallest or the largest subset of $\mathcal{Q}_t$ that is equivalent to $\mathcal{Q}_e$. The smallest subset can be obtained by removing all of the cycles containing even number of edges from $G(C,\mathcal{Q}_{dt})$. If $\mathcal{Q}_e$ is safe, we then have a spanning tree of $G(C,\mathcal{Q}_{dt})$, which corresponds to a set of linearly independent row vectors in the incidence matrix. On the other hand, we are more interested in the maximal subset of $\mathcal{Q}_t$ that is equivalent to $\mathcal{Q}_e$. According to Lemma 2, a safe $\mathcal{Q}_e$ essentially allows users to sum any two tuples from different color classes of $G(C,\mathcal{Q}_{dt})$, and to subtract any two tuples of the same color. The maximal subset of $\mathcal{Q}_t$ equivalent to $\mathcal{Q}_e$ is hence the complete bipartite graph with the same bipartition of $G(C,\mathcal{Q}_{dt})$.

Lemma 2. *Given that* $\mathcal{Q}_e$ *is safe, let* (C_1,C_2) *be the bipartition of* $G(C,\mathcal{Q}_{dt})$ *and* $\mathcal{Q}^\star_{dt} = \{q^2(u,v) : u \in C_1, v \in C_2\}$. *We have that*

1. $\mathcal{Q}^\star_{dt} \equiv_d \mathcal{Q}_{dt}$.
2. For any $S \subseteq \mathcal{Q}_t$, *if* $S \equiv_d \mathcal{Q}_{dt}$ *then* $S \subseteq \mathcal{Q}^\star_{dt}$.
3. For any $t_1,t_2 \in C_1$ *(or* $t_1,t_2 \in C_2$ *), there exists* $r \in \mathbb{R}^{|\mathcal{Q}_{dt}|}$ *such that* $\mathcal{M}(t_1) - \mathcal{M}(t_2) = r \cdot \mathcal{M}(\mathcal{Q}_{dt})$.

Odd MDR Queries. Now that we can determine the compromisability of $\mathcal{Q}_e$, we would like to know if anything else can be answered safely. First, we consider odd MDR queries that form the complement of $\mathcal{Q}_e$ with respect to all MDR queries $\mathcal{Q}_d$. Intuitively, feeding any odd MDR query $q^\star(u_0,v_0)$ into *Sub_QDT* as the input gives us a single tuple t. Suppose $q^\star(u_0,v_0)$ is a j-dimensional box. It can be divided into two j-dimensional boxes excluding t, together with a $(j-1)$-dimensional box containing t. We can recursively divide the $(j-1)$-dimensional box in the same way. Hence, $q^\star(u_0,v_0)$ is the union of a few disjointed even MDR queries together with a singleton set $\{t\}$. This is formally stated in Corollary 2.

Corollary 2. *Given* $d \in \mathbb{R}^k$, $F = \mathcal{F}(d)$, $C \subseteq F$, *and any* $q^\star(u,v) \in \mathcal{Q}_d \setminus \mathcal{Q}_e$ *satisfying* $\mid \{i : u[i] \neq v[i]\} \mid = j$, *there exists* $q^\star(u_i,v_i) \in \mathcal{Q}_e$ *for all* $1 \leq i \leq 2j-1$, *such that* $\mid q^\star(u,v) \setminus \bigcup_{i=1}^{2j-1} q^\star(u_i,v_i) \mid = 1$ *and* $q^\star(u_i,v_i) \cap q^\star(u_l,v_l) = \phi$ *for all* $1 \leq i < l \leq 2j-1$.

Example 6. In Table 4, using $q^\star((1,1),(2,3))$ as the input of *Sub_QDT* gives the output $(1,3)$. $q^\star((1,1),(2,3))$ can be divided into $q^\star((1,1),(1,3))$ and $q^\star((2,2),(2,3))$. $q^\star((1,1),(1,3))$ can be further divided into $q^\star((1,1),(1,2))$ and $\{(1,3)\}$. Hence, we have $q^\star((1,1),(2,3)) = q^\star((1,1),(1,2)) \cup q^\star((2,2),(2,3)) \cup \{(1,3)\}$

Corollary 2 has two immediate consequences. First, no odd MDR query is safe in addition to $\mathcal{Q}_e$. Equivalently, any subset of $\mathcal{Q}_d$ with $\mathcal{Q}_e$ as its proper subset is unsafe. Second, any odd MDR query is different from the union of a few number of even MDR queries by only one tuple. This difference is negligible because most users of MDR queries are interested in patterns and trends instead of individual values.

Arbitrary Queries. We know the implication of $\mathcal{Q}_e$ in terms of sum-two queries from Lemma 2. Hence, we can easily decide which arbitrary queries can be answered in addition to a safe $\mathcal{Q}_e$. Corollary 3 shows that any arbitrary query can be answered iff it contains the same number of tuples from the two color classes of $G(C, \mathcal{Q}_{dt})$. This can be decided in linear time in the size of the query by counting the tuples it contains. The compromisability of odd MDR queries hence becomes a special case of Corollary 3, because no odd MDR query can satisfy this condition.

Corollary 3. *Given that $\mathcal{Q}_e$ is safe, for any $q \subseteq C$, $q \preceq_d \mathcal{Q}_e$ iff $\mid q \cap C_1 \mid = \mid q \cap C_2 \mid$, where (C_1, C_2) is the bipartition of $G(C, \mathcal{Q}_{dt})$.*

6 Unsafe Even MDR Queries

In this section we consider the situations where even MDR queries are unsafe. We show the equivalence between subsets of even MDR queries and sum-two queries, and give a sufficient condition for the safe subsets.

We have seen in Section 4.2 that finding maximal safe subsets of queries is infeasible even for queries of restricted form, such as sum-two queries and data cubes. Hence, we turn to large but not necessarily maximal safe subsets that can be found efficiently. Recall that in Section 5 we were able to efficiently determine the compromisability of $\mathcal{Q}_e$ because of $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$. If we could establish the equivalence between their subsets, we would be able to extend the results in Section 5 to those subsets. However, equivalence does not hold for arbitrary subsets of $\mathcal{Q}_e$ or $\mathcal{Q}_{dt}$, as shown in Example 7.

Example 7. Consider $\mathcal{Q}_{dt}$ of Example 4. Let $S_{dt} = \mathcal{Q}_{dt} \setminus \{q^2((1,1),(1,2))\}$. Suppose $S_{dt} \equiv_d S_e$ for some $S_e \subseteq \mathcal{Q}_e$. Because $q^\star((1,3),(2,4)) \preceq_d S_e$, S_e must contain $q^\star((1,1),(1,2))$, but then $q^\star((1,1),(1,2)) \not\preceq_d S_{dt}$, a contradiction. Hence, S_{dt} is not equivalent to any subset of $\mathcal{Q}_e$. Similarly, $\mathcal{Q}_e \setminus \{q^\star((1,1),(1,2))\}$ is not equivalent to any subset of $\mathcal{Q}_{dt}$.

Intuitively, any MDR query can be viewed as a *sub-core*. The equivalence given in Theorem 2 must also hold for this sub-core as the following. The even MDR queries defined in the sub-core are equivalent to the sum-two queries added to $\mathcal{Q}_{dt}$ by *Sub_QDT* with those even MDR queries as its inputs. This result can be extended to any subset of the core, as long as the subset can be represented as the union of some sub-cores. Given any $S \subseteq \mathcal{Q}_e$, if we delete each $q^\star(u,v) \in \mathcal{Q}_e \setminus S$ from the core then the result must be the union of some sub-cores. Similarly, given any $S \subseteq \mathcal{Q}_{dt}$, for each $q^2(u,v) \in \mathcal{Q}_{dt} \setminus S$, if we delete $q^\star(u,v)$ from the core then the result is the union of some sub-cores. In this way, the equivalence between subsets of $\mathcal{Q}_e$ and subsets of $\mathcal{Q}_{dt}$ can always be established. This is formalized in Proposition 1.

Proposition 1. *1. Given any $S \subseteq \mathcal{Q}_e$, let $S_e = S \setminus \{q^\star(u,v) : \exists q^\star(u_0,v_0) \in \mathcal{Q}_e \setminus S, q^\star(u,v) \cap q^\star(u_0,v_0) \neq \phi\}$ and $S_{dt} = \{q^2(u,v) : \exists q^\star(u_0,v_0) \in S_e, q^2(u,v) \in \mathcal{Q}_{dt}$ due to $q^\star(u_0,v_0)\}$. Then $S_e \equiv_d S_{dt}$.*

2. Given any $S \subseteq \mathcal{Q}_{dt}$, let $S_e = \mathcal{Q}_e \setminus \{q^\star(u,v) : \exists (u_0,v_0), q^2(u_0,v_0) \in S \wedge q^\star(u,v) \cap q^\star(u_0,v_0) \neq \phi\}$, and $S_{dt} = \{q^2(u,v) : \exists q^\star(u_0,v_0) \in S_e, q^2(u,v) \in \mathcal{Q}_{dt}$ due to $q^\star(u_0,v_0)\}$. Then $S_{dt} \equiv_d S_e$.

Proposition 1 guarantees the equivalence at the cost of smaller subsets. In some situations, we are satisfied with the weaker result, such as $S_{dt} \succeq S_e$ for some $S_e \subseteq \mathcal{Q}_e$. Because then if S_{dt} is safe, then S_e must also be safe, although the converse is not always true. The result in Proposition 2 is similar to Corollary 3 but gives only the sufficient condition. In Proposition 2, S_e can be found by examining each query in $\mathcal{Q}_e$ against the bipartition (C_1, C_2), taking time $O(mn)$, where $m =\mid \mathcal{Q}_e \mid$ and $n =\mid C \mid$.

Proposition 2. *For any $S_{dt} \subseteq \mathcal{Q}_{dt}$, let (C_1, C_2) be the bipartition of $G(C, S_{dt})$. Then $S_{dt} \succeq S_e$ holds, where $S_e \subseteq \mathcal{Q}_e$ satisfies that for any $q^\star(u,v) \in S_e$, $\mid q^\star(u,v) \cap C_1 \mid = \mid q^\star(u,v) \cap C_2 \mid = \mid q^\star(u,v) \mid /2$ holds.*

By Proposition 2 we can efficiently find a safe subset S_e of $\mathcal{Q}_e$ if a safe subset S_{dt} of $\mathcal{Q}_{dt}$ is given. The ideal choice of S_{dt} should maximize $\mid S_e \mid$. This is equivalent to computing the *combinatorial discrepancy* of the set system formed by C and $\mathcal{Q}_e$ [3]. The alternative approach is to maximize $\mid S_{dt} \mid$, which is equivalent to finding the maximal bipartite subgraph of $G(C, \mathcal{Q}_{dt})$.

Instead of those solutions that may incur high complexity, we can apply a simple procedure given in [16]. It takes the graph $G(C, \mathcal{Q}_{dt})$ as the input and outputs a bipartite subgraph. It starts from an empty vertex set and empty edge set and processes one vertex at each step. The unprocessed vertex is colored blue if at least half of the processed vertices to which it connects are red. It is colored red, otherwise. Any edge in the original graph is included in the output bipartite subgraph if it connects two vertices in different colors. The procedure terminates with a bipartite graph $G(C, \mathcal{Q}_{ds})$ satisfying that $\mid \mathcal{Q}_{ds} \mid \geq \mid \mathcal{Q}_{dt} \mid /2$. The procedure runs in $O(n^2) = O(m)$, where $n =\mid C \mid$ and $m =\mid \mathcal{Q}_e \mid$. Our ongoing work will address the effectiveness of this procedure through empirical results.

7 Discussion

A novel three-tier inference control model was proposed for OLAP systems in [29]. The results given in Sections 5 and 6 fit in this model perfectly. In this section, we briefly justify this claim but leave out more details due to space limitations.

The Three-Tier Inference Control Model of [29]. The objectives of the three-tier inference control model are to minimize the performance penalty of inference control methods and to make inference control less vulnerable to undetected external knowledge. This is achieved by introducing a new tier, *aggregation tier A*, to the traditional two-tier view (i.e., *data tier D* and *query tier Q*) of inference control. The three tiers are related by $R_{AD} \subseteq A \times D$, $R_{QA} \subseteq Q \times A$, and $R_{QD} = R_{AD} \circ R_{QA}$. The aggregation tier A satisfies three conditions. First, $\mid A \mid$ is comparable to $\mid D \mid$. Second, there exists partition $\mathcal{P}$ on A such that the composition of R_{AD} and the equivalence relation decided by $\mathcal{P}$ gives a partition on D. Finally, inferences are eliminated in the aggregation tier A.

The three-tier model gains its advantages through its three properties. Because $\mid A \mid$ is relatively small (suppose $\mid Q \mid >> \mid D \mid$), controlling inferences of A is easier than that of Q because of the smaller input to inference control methods. Because of the second property of A, inference control can be *localized* to the R_{AD}-related blocks of A and D, which further reduces the complexity. Moreover, any consequences of undetected external knowledge in some blocks are confined to these blocks, making inference control more *robust*. Finally, as the most expensive task of three-tier inference control, the construction of A can be processed off-line (i.e., before any query arrives). Because decomposing queries into pre-computed aggregations is a built-in capability in most OLAP systems, the online performance overhead of three-tier inference control is almost negligible.

Applicability of Our Results. Partitions of data sets based on the dimension hierarchies naturally compose the data tier. Each block in the partition corresponds to a core. The safe $\mathcal{Q}_{dt}$ (or its safe subsets S_{dt} if it is unsafe) composes each block of the aggregation tier. The query tier includes any arbitrary query derivable from the aggregation tier. If we characterize $\mathcal{Q}_e$ using the row vectors in $\mathcal{M}(\mathcal{Q}_e)$, then the query tier is the linear space they span. The relations R_{AD} and R_{QA} are both the derivability relation $\preceq_d$ given in Definition 3, and $R_{QD} = R_{AD} \circ R_{QA}$ is a subset of $\preceq_d$, because $\preceq_d$ is transitive.

In Section 5 we showed that $\mid \mathcal{Q}_{dt} \mid = O(n^2)$, where $n = \mid C \mid$, satisfying the first condition of the three tier model. Because $\mathcal{Q}_{dt}$ is defined separately on each core, the aggregation tier has a natural partition corresponding to the partition of the data tier, satisfying the second condition. The last condition is satisfied because we use the safe subsets of $\mathcal{Q}_{dt}$ when it is unsafe. Hence by integrating our results on the basis of the three tier model, we inherit all the advantages including negligible online performance overhead, and the robustness in the face of undetected external knowledge (that is, the damage caused by undetected external knowledge is confined to blocks of the partition of data tier and aggregation tier).

Moreover, our results provide better usability to OLAP systems than the cardinality-based approach in [29] does. Firstly, the cardinality-based conditions become invalid when MDR queries other than those contained in the data cube (i.e., skeleton queries) are answered. In this paper we allow any MDR queries if only they are safe. The MDR queries generalize data cubes and various data cube operations, such as slicing, dicing, roll up and drill down. Our answers to even MDR queries are precise, and the answered even MDR queries closely approximate the restricted odd ones. Secondly, when a data cube is unsafe, it is simply denied in [29]. However, in this paper we are able to give partial answers to an unsafe set of even MDR queries, implying better usability. Our methods for computing the partial answers are also efficient. Thirdly, we use necessary and sufficient conditions to determine safe even MDR queries, while the cardinality-based conditions are only sufficient. Therefore, we can provide more answers to users without privacy breaches than the methods of [29] do.

8 Conclusion and Future Direction

In this paper we have shown the infeasibility of applying several existing restrictions to MDR queries. We then proved the equivalence between the even MDR queries and a special set of sum-two queries. On the basis of this equivalence we are able to efficiently determine the compromisability of even MDR queries. We showed that the restricted odd MDR queries are closely approximated by the answered even ones. We showed that safe arbitrary queries can be efficiently determined. We can also maintain this equivalence when even MDR queries are unsafe. Our on-going work implements the proposed algorithms in order to explore their fine tunings. Another future direction is to investigate the aggregation operators other than SUM.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable comments.

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 and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 439–450, 2000.
3. J. Beck and V.T. Sós. Discrepancy theory. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of combinatorics*, pages 1405–1446. Elsevier Science, 1995.
4. L.L. Beck. A security mechanism for statistical databases. *ACM Trans. on Database Systems*, 5(3):316–338, 1980.
5. L. Brankovic, M. Miller, P. Horak, and G. Wrightson. Usability of compromise-free statistical databases. In *Proceedings of ninth International Conference on Scientific and Statistical Database Management (SSDBM '97)*, pages 144–154, 1997.
6. A. Brodsky, C. Farkas, D. Wijesekera, and X.S. Wang. Constraints, inference channels and secure databases. In *the 6th International Conference on Principles and Practice of Constraint Programming*, pages 98–113, 2000.
7. F.Y. Chin. Security in statistical databases for queries with small counts. *ACM Transaction on Database Systems*, 3(1):92–104, 1978.
8. F.Y. Chin, P. Kossowski, and S.C. Loh. Efficient inference control for range sum queries. *Theoretical Computer Science*, 32:77–86, 1984.
9. F.Y. Chin and G. Özsoyoglu. Security in partitioned dynamic statistical databases. In *Proc. of IEEE COMPSAC*, pages 594–601, 1979.
10. F.Y. Chin and G. Özsoyoglu. Auditing and inference control in statistical databases. *IEEE Trans. on Software Engineering*, 8(6):574–582, 1982.
11. L.H. Cox. Suppression methodology and statistical disclosure control. *Journal of American Statistical Association*, 75(370):377–385, 1980.
12. D.E. Denning and P.J. Denning. Data security. *ACM computing surveys*, 11(3):227–249, 1979.
13. D.E. Denning, P.J. Denning, and M.D. Schwartz. The tracker: A threat to statistical database security. *ACM Trans. on Database Systems*, 4(1):76–96, 1979.
14. D.E. Denning and J. Schlörer. Inference controls for statistical databases. *IEEE Computer*, 16(7):69–82, 1983.

15. D. Dobkin, A.K. Jones, and R.J. Lipton. Secure databases: protection against user influence. *ACM Trans. on Database Systems*, 4(1):97–106, 1979.
16. P. Erdös. On some extremal problems in graph theory. *Isarel Journal of Math.*, 3:113–116, 1965.
17. L.P. Fellegi. On the qestion of statistical confidentiality. *Journal of American Statistic Association*, 67(337):7–18, 1972.
18. J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational operator generalizing group-by, crosstab and sub-totals. In *Proceedings of the 12th International Conference on Data Engineering*, pages 152–159, 1996.
19. D.T. Ho, R. Agrawal, N. Megiddo, and R. Srikant. Range queries in olap data cubes. In *Proceedings 1997 ACM SIGMOD International Conference on Management of Data*, pages 73–88, 1997.
20. J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *Proc. of the 9th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 86–91, 2000.
21. Y. Li, L. Wang, and S. Jajodia. Preventing interval based inferece by random data perturbation. In *Proceedings of The Second Workshop on Privacy Enhancing Technologies (PET'02)*, 2002.
22. Y. Li, L. Wang, X.S. Wang, and S. Jajodia. Auditing interval-based inference. In *Proceedings of the 14th Conference on Advanced Information Systems Engineering (CAiSE'02)*, pages 553–568, 2002.
23. Y. Li, L. Wang, S.C. Zhu, and S. Jajodia. A privacy enhanced microaggregation method. In *Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2002)*, pages 148–159, 2002.
24. F.M. Malvestuto and M. Mezzini. Auditing sum queries. In *Proceedings of the 9th International Conference on Database Theory (ICDT'03)*, pages 126–146, 2003.
25. J.M. Mateo-Sanz and J. Domingo-Ferrer. A method for data-oriented multivariate microaggregation. In *Proceedings of the Conference on Statistical Data Protection'98*, pages 89–99, 1998.
26. S. Rizvi and J.R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th Conference on Very Large Data Base (VLDB'02)*, 2002.
27. J. Schlörer. Security of statistical databases: multidimensional transformation. *ACM Trans. on Database Systems*, 6(1):95–112, 1981.
28. J.F. Traub, Y. Yemini, and H. Woźniakowski. The statistical security of a statistical database. *ACM Trans. on Database Systems*, 9(4):672–679, 1984.
29. L. Wang, D. Wijesekera, and S. Jajodia. Cardinality-based inference control in sum-only data cubes. In *Proceedings of the 7th European Symposium on Research in Computer Security (ESORICS'02)*, pages 55–71, 2002.
30. L. Wang, D. Wijesekera, and S. Jajodia. Olap means on-line anti-privacy. Technical Report, 2003. Available at http://ise.gmu.edu/techrep/2003/.
31. L. Wang, D. Wijesekera, and S. Jajodia. Precisely answering multi-dimensional range queries without privacy breaches. Technical Report, 2003. Available at http://ise.gmu.edu/techrep/2003/.