Journal of Computer Security 15 (2007) 417–445
IOS Press

Parity-based inference control for multi-dimensional range sum queries

Lingyu Wang [a,*], Yingjiu Li [b], Sushil Jajodia [c] and Duminda Wijesekera [c]
[a] *Concordia Institute for Information Systems Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, QC H3G 1M8, Canada*
E-mail: wang@ciise.concordia.ca
[b] *School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902, Republic of Singapore*
E-mail: yjli@smu.edu.sg
[c] *Center for Secure Information Systems, George Mason University, MSN 5B5, 4400 University Drive, Fairfax, VA 22030-4444, USA*
E-mail: {dwijesek,jajodia}@gmu.edu

This paper studies the inference control of multi-dimensional range (MDR) sum queries. We show that existing inference control methods are usually inefficient for MDR queries. We then consider *parity-based* inference control that restricts users to queries involving an even number of sensitive values. Such a restriction renders inferences significantly more difficult, because an even number is closed under addition and subtraction, whereas inferences target at one value. However, more sophisticated inferences are still possible with only even MDR queries. We show that the collection of all even MDR queries causes inferences if and only if a special collection of *sum-two* queries (that is, the summation of exactly two values) does so. The result leads to an inference control method with an improved computational complexity $O(mn)$ (over the previous result of $O(m^2n)$) for m MDR queries over n values. We show that no *odd* MDR queries can be answered without causing inferences. We show how to check non-MDR queries for inferences in linear time. We also show how to find large inference-free subsets of even MDR queries when they do cause inferences.

Keywords: Database security, data privacy, OLAP, range query, inference control, data cube

1. Introduction

The multi-dimensional range (MDR) sum query is an important class of decision support queries in OLAP (On-Line Analytical Processing) systems [23]. A popular data model of OLAP systems, the data cube [22], can be regarded as a special collection of MDR queries. MDR queries are intended for analysts to generalize large amounts of data stored in data warehouses and to discover statistical trends and patterns. Contrary to this initial objective, MDR queries may be used to infer protected sensitive values, leading to the breach of an individual's privacy.

*Corresponding author. Tel.: +1-514-848-2424-5662, Fax: +1-514-848-3171, E-mail: wang@ciise.concordia.ca.

Access control can prevent unauthorized access to sensitive data, but it is unaware of indirect inferences caused by seemingly innocent queries. Inference control of ad hoc queries has been investigated since the 1970's in statistical databases and census data. However, most of the proposed methods suffer from high computational complexity. For example, the *audit expert* can effectively control inferences for SUM-only queries with a complexity of $O(m^2n)$ for m queries over n sensitive values [12], and auditing both SUM and MAX queries is NP-complete [9]. Chin pointed out that *one obvious approach to bring the complexity to a practical level is to include restrictions on user's queries, such that only statistically meaningful queries can be specified*. Our study represents such an effort in finding efficient inference control methods for statistically meaningful MDR queries.

The contributions of this paper are as follows. First, we propose the concept of *parity-based* inference control. Intuitively, an even number is closed under addition and subtraction, whereas inferences target at exactly one value (in this paper we will only consider inferences of an exact value) and one is an odd number. Hence, restricting users to *even* MDR queries (that is, MDR queries that sum an even number of sensitive values) can make inferences significantly more difficult. However, more sophisticated inferences are still possible with only even MDR queries. Second, we show that the collection of all even MDR queries is free of inferences, if and only if a special collection of sum-two queries (that is, the summation of exactly two values) is so. Finding such a collection of sum-two queries takes time $O(mn)$, and determining whether it causes inferences takes time $O(m+n)$ for m MDR queries over n values. This result thus leads to an inference control method with computational complexity $O(mn)$, which is an improvement to the best known result of $O(m^2n)$ [12].

Third, we show that in addition to answering even MDR queries, no MDR query involving an odd number of values can be answered without causing inferences. However, for any such *odd* MDR queries, we can always find a small number of even MDR queries whose union differs from the odd MDR query by exactly one value. The odd MDR query can thus be approximately answered. We study how to detect inferences for non-MDR queries in linear time in the number of values involved by the queries. We also study the case where the collection of all even MDR queries does cause inferences. We show how to find large inference-free subsets of the collection. Finally, we show that the proposed methods can be integrated on the basis of a three-tier inference control model previously proposed. The preliminary results that appeared in [38] are further elaborated in the current paper.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 discusses examples to motivate our study. Section 4 formalizes concepts needed for further discussions. Section 5 shows that directly applying existing inference control methods to MDR queries is inefficient. Section 6 studies the parity-based inference control method. of even MDR queries causes inferences. Section 7 discusses how to integrate the results in a three-tiered inference control model. Section 8 concludes the paper.

2. Related work

Inference control has been extensively studied in statistical databases [1,15,17] 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 values involved in a query) [21], restricting the size of overlaps between query sets [18], detecting inferences through auditing queries [6,10,12,24], suppressing tabular data to prevent inferences [14], partitioning values, and restricting queries to complete blocks in the partition [11,30]. Perturbation-based techniques add random noises to source data, outputs, or the structure of databases [5,34,35]. Other aspects of the inference problem include the inferences in multi-level databases [7] and the inferences targeting approximated values [26–29]. Directly applying the inference control methods in statistical databases is not desired, because these methods are intended for arbitrary queries and they typically ignore the unique structures of MDR queries. We will discuss some of those methods in more detail in Section 5.

Controlling inferences of a special class of MDR queries, namely, *data cube* queries, have been studied in [25,37,39,40]. First, the study in [39,40] shows that a SUM-only data cube is free of inferences if the number of previously known values is below a tight upper bound (the bound is tight in the sense that no better bound exists). However, the converse is not necessarily true, and an inference-free data cube may be mistakenly taken as causing inferences by the method proposed in [39,40]. Second, the study in [37] first restricts queries such that the adversary cannot combine multiple queries for an inference. This approach greatly eases inference control and applies to any aggregation functions as long as certain algebraic properties are satisfied. Third, the study in [25] addresses the approximate inferences in terms of lower and upper bounds of the actual values. However, these methods do not directly apply to our case, because data cube queries are only a subset of all MDR queries based on explicit dimension hierarchies.

The inference problem of one-dimensional range queries has been studied before, and the author considers the multi-dimensional case as difficult [10]. The *usability* (i.e., the highest possible ratio of the number of inference-free queries to that of all queries) of MDR queries in the absence of previously known values has been studied [6]. The restriction of even MDR queries is mentioned but not fully explored, and the more general case with arbitrary known values is regarded as challenging. Chin et al. give necessary and sufficient condition for the sum-two queries to be inference-free, and they show that finding the maximal inference-free subsets of sum-two queries is NP-hard [8,12]. However, in practice queries are rarely limited to sum-two queries. In this paper, we generalize the results on sum-two queries to even MDR queries.

Perturbation-based methods have been proposed for preserving privacy in data mining applications [2]. Random noises are added to destroy the sensitive information while the statistical distribution is approximately reconstructed from the per-

turbed data to facilitate data mining tasks. The methods proposed in [3] can approximately reconstruct COUNTs from perturbed data with statistically bound errors, so OLAP tasks like classification can be fulfilled. However, in general protecting sensitive data in OLAP is different from that in data mining, because OLAP tasks may demand small details, such as outliers, that cannot be obtained from distribution models alone. Potential errors in individual values may be significant, preventing OLAP users from gaining trustful insights. The methods we study are based on restrictions and hence do not introduce any noises. *Secure multi-party data mining* allows multiple distrusted parties to cooperatively compute data mining results with minimal disclosures of their own data [19,36]. This problem is different from inference control, because the threat of inferences comes from what users know, not from the way they know it.

The k-*anonymity* model releases sensitive values but renders them anonymous such that they do not threaten privacy [13,32,33,41]. In the released table, each record is indistinguishable from at least $k-1$ others due to the same combinations of identifying attribute values. An adversary can thus link an individual in the physical world to at best k records, which is considered a tolerable privacy threat. Inference control and the k-anonymity model can be considered as dual approaches, and they are suitable for different applications. The information theoretic approach in [31] formally characterizes insecure queries as those that give a user more confidence in guessing possible database instances [31]. However, such a *perfect-secrecy* metric will not tolerate any partial disclosure, including those caused by aggregated values.

The preliminary results of the current paper have appeared in [38]. The current paper elaborates on and provides full proofs to these results. Moreover, Section 7 discusses in detail how the results can be applied to OLAP systems based on the previously proposed three-tiered inference control model, and it also describes several limitations and discusses possible extensions to the proposed approach.

3. Motivating example

We discuss a running example to motivate further study. Table 1 depicts a fictitious *data set* of salary adjustments for four employees in two consecutive years. Assume the salary adjustments are sensitive and should be kept secret. In the example, the empty cells denote values that are already known to users through outbound channels. Using empty cells indicates the fact that these values can no longer be protected and will be ignored in our discussions. However, such a value is different from a zero adjustment, because the latter may not be known to users and needs to be protected. Suppose a third-party analyst Mallory is invited to analyze the above data set. Considering that Mallory may later misuse the information about individuals and causes privacy issues, she is not supposed to know each employee's salary adjustment. Access control mechanisms will thus deny any queries about an employee's salary adjustment in a year.

Table 1
An example of sensitive data set and inferences

	Alice	Bob	Mary	Jim
2002	1000	500	–2000	
2003		1500	–500	1000

However, Mallory's analyzing tasks may require her to know the summation of a range of values with similar characteristics, such as each employee's total salary adjustments in the two years (each column in Table 1) or the total salary adjustments in a year (each row). Intuitively, those values are inside a *box*, which can be represented by any of its longest *diagonals*. For example, [(*Alice*, 2002), (*Alice*, 2003)] stands for the first column of the table and [(*Alice*, 2002), (*Bob*, 2003)] for the first two columns. We will only consider a query asking for the summation of values in a continuous range, namely, a *multi-dimensional range* SUM query (or simply MDR query). For example, a request for the summation of values in the first and the fourth columns is not an MDR query since the values are not inside any continuous range.

Although access control will prohibit queries asking for a salary adjustment, Mallory can get around the restriction by asking MDR queries. For example, the MDR query [(*Alice*, 2002), (*Alice*, 2003)] gives Alice's adjustment in 2002, because the query sums a single value. As another example, the difference between the answers to [(*Bob*, 2002), (*Mary*, 2002)] and [(*Alice*, 2002), (*Mary*, 2002)] yields the same result. Mallory can potentially combine any answered MDR queries for an inference, whereas inference control must prevent all such possibilities.

The key observation from the above example is that one of the queries asks for the summation of an odd number of values. Considering the fact that even number is closed under addition and subtraction, it would be more difficult to infer one (which is an odd number) value if only *even* MDR queries are to be allowed. For example, in Table 1, inferences may no longer be straightforward if only even MDR queries are to be asked. We will call the restriction a *parity-based inference control* henceforth.

Nonetheless, more sophisticated inferences are still possible with MDR queries. Table 2 depicts five even MDR queries and their answers. The first query sums all six values and the remaining four queries each sums two values. Mallory then adds the answers to the last four queries (2500) and subtracts from the result the answer to the first query (1500). Dividing the result (1000) by two gives Bob's adjustment in 2002 (500).

The rest of the paper answers following questions naturally motivated by the above example. 1. *How can we efficiently determine whether the collection of all even MDR queries causes inferences?* 2. *In addition to even MDR queries, what else can be answered without causing inferences?* 3. *How can we find large subsets of even MDR queries that are inference free?*

Table 2
An example of even MDR queries and answers

Queries	Answers
[(*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

4. The model

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 $u \leqslant v$ and $t \in [u, v]$ to mean that $u[i] \leqslant v[i]$ and $\min\{u[i], v[i]\} \leqslant t[i] \leqslant \max\{u[i], v[i]\}$ hold for all $1 \leqslant i \leqslant k$, respectively. We use t for the singleton set $\{t\}$ whenever it is clear from the context.

Definition 1 formalizes *domain*, *data set*, and *tuple*. The domain is the Cartesian product of closed integer intervals. A data set is any subset of the domain. A tuple is any vector in the domain. With respect to Table 1 in Section 3, we use a tuple to interchangebly refer to a cell and the sensitive value in that cell (notice that Table 1 is a cross-tabular instead of a flat relational table, and hence our notion of a tuple is different from a relational tuple). A tuple *missing* from the data set is any vector in the complement of the data set with respect to the domain. In our study, missing tuples represent sensitive values that adversaries have learned through outbound channels (that is, empty cells in Table 1).

Definition 1 (Data Set). 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 **domain**, any $C \subseteq F$ a **data set**, any $t \in F$ a **tuple**, and any $t \in F \setminus C$ a tuple **missing** from C.

Example 1. Table 3 rephrases the example in Table 2 using notations given in Definition 1. The six tuples in the data set correspond to the six sensitive values unknown to users, and the missing tuples represent previously known values (the subscripts are needed later in this section for the correspondence between tuples and columns of the incidence matrix).

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

Table 3
Modeling data set

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

Table 4
Modeling MDR queries

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

Definition 2 (Arbitrary Query, MDR Query, and Sum-two Query). Given any domain F and data set $C \subseteq F$,

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 subset of C an **arbitrary query**, any $q^\star(u,v) \in \mathcal{Q}_d$ an **MDR query** (or simply a query), and any $q^2(u,v) \in \mathcal{Q}_t$ a **sum-two query**.

Example 2. Table 4 rephrases the five MDR queries in Table 1 using our notations. The left side of the table specifies each query and the right side gives the set of tuples included in that query.

Definition 3 formalizes the concept of *compromiseability*. Because an arbitrary query is a set of tuples, any given collection of arbitrary queries can be characterized by the incidence matrix of the set system formed by the data set C and the collection of arbitrary queries $\mathcal{S}$ (that is, a matrix $\mathcal{M}$ satisfying that $\mathcal{M}(\mathcal{S})[i,j] = 1$ if the ith arbitrary query in $\mathcal{S}$ contains the jth tuple in C, and $\mathcal{M}(\mathcal{S})[i,j] = 0$ otherwise). Given two collections 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 means the former can be computed from the latter and hence discloses less information than the latter does. We say $\mathcal{S}_1$ *compromises* a tuple t in the data set, if the singleton set of queries $\{\{t\}\}$ (notice $\{t\}$ is an arbitrary query) is derivable from $\mathcal{S}_1$, and $\mathcal{S}_1$ is *safe* if it compromises no tuple in the data set. We say any two sets of arbitrary queries are *equivalent* if they are mutually derivable. Example 3 illustrates the concepts we just defined.

$$(1,2) \preceq_d \mathcal{S} \text{ because } [0,1,0,0,0,0] = [-\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}] \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Fig. 1. Modeling compromiseability.

Definition 3 (Compromiseability). Given any domain F, data set $C \subseteq F$, and set of arbitrary queries $\mathcal{S}$, use $\mathcal{M}(\mathcal{S})$ for the incidence matrix 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 $\mathcal{S}_2 \preceq_d \mathcal{S}_1$.

Example 3. Following Example 1 and Example 2, Fig. 1 gives an example of the compromiseability. The equation shows that the five queries in Example 2 compromise a tuple $(1, 2)$. The left side of the equation is the incidence matrix of the query $\{(1, 2)\}$, and the right side is a linear combination of the row vectors in the incidence matrix of the five MDR queries in Example 2.

The relation $\equiv_d$ of Definition 3 is an equivalence relation on the family of all sets of arbitrary queries, because it is clearly reflexive, symmetric and transitive. Hence, if any two sets of arbitrary queries are equivalent, then one is safe iff the other is. This observation is the basis for our discussions in Section 6 about reducing the compromiseability of even MDR queries to that of sum-two queries.

5. Applying existing inference control method to MDR queries

This section studies the feasibility of applying existing restriction-based inference control methods to MDR queries. First, Section 5.1 considers three methods, namely, *Query set size control*, *overlap size control* and *Audit Expert*. Second, Section 5.2 studies the problem of finding maximal safe subsets of MDR queries.

5.1. Query set size control, overlap size control and audit expert

Query set size control. This method prohibits users from asking *small* queries, which ask for the summation of less than n_t values where n_t is a pre-determined threshold. For example, the inference of exact values is trivial if users can ask for the summation of one value (that is, n_t must be at least two). However, the query

set size control is necessary but not sufficient for controlling inferences. For arbitrary queries, query set size control can be easily subverted by asking two legitimate queries whose difference yields a prohibited one, a mechanism known as *tracker* in statistical databases [16]. It is shown that finding a tracker for arbitrary queries is possible even when n_t is about half of the cardinality of the data set.

At first glance, trackers may seem to be more difficult to find when users are restricted to MDR queries, because they can now ask less queries than in the case of arbitrary queries. In another word, the query set size control is expected to be more effective when applied to MDR queries. Unfortunately, this is not true. The query set size control can still be easily subverted using trackers, even when MDR queries are the only kind of queries users may ask. Proposition 1 shows that in most cases a tracker consisted of MDR queries can be found to derive any given small MDR query (such as an MDR query that includes only one tuple).

Specifically, Proposition 1 shows that even when n_t is set to be comparable to the size of the data set (3^k is a constant compared to $|C|$), any MDR query including less than n_t tuples (and hence should not be answered) can be derived from those queries including n_t or more tuples. The proof of the proposition constructs trackers for deriving a given query. Intuitively, the targeted MDR query is surrounded by 3^k other MDR queries, and one of them must include more than n_t tuples, and hence is legitimate to answer. The targeted query can then be derived by padding it with this legitimate query. Example 4 illustrates the idea in the one-dimensional case.

Proposition 1. *Given $d \in \mathbb{R}^k$, $F = \mathcal{F}(d)$ and $C \subseteq F$, let $n_t = \lfloor \frac{|C|}{3^k} \rfloor$. For any $q^\star(u_a, v_a) \in \mathcal{Q}_d$ satisfying $|q^\star(u_a, v_a)| < n_t$, we have that $q^\star(u_a, v_a) \preceq_d \{q^\star(u, v) : |q^\star(u, v)| \geqslant n_t\}$.*

Proof. We first show that the data set can be partitioned into 3^k blocks including $q^\star(u_a, v_a)$. Let $S = \{q^\star(u, v) : \forall i \in [1, k], (u[i] = 1, v[i] = u_a[i] - 1) \vee (u[i] = u_a[i], v[i] = v_a[i]) \vee (u[i] = v_a[i] + 1, v[i] = d[i])\}$. We have that $C = \bigcup_{\forall q \in S} q$, and $q^\star(u, v) \cap q^\star(u_a, v_a) = \phi$ holds for any $q^\star(u, v) \in S \setminus q^\star(u_a, v_a)$. Because $|S| = 3^k$, there must exist $q^\star(u_b, v_b) \in S$ such that $|q^\star(u_b, v_b)| \geqslant \frac{|C|}{n_t}$.

Next we define the tracker as

1. u_c, v_c satisfying that $u_c[i] = \min\{u_a[i], u_b[i], v_b[i]\}$, and $v_c[i] = \max\{u_b[i], v_a[i], v_b[i]\}$ for all $1 \leqslant i \leqslant k$.
2. For all $1 \leqslant i \leqslant k$, u_i satisfying that $u_i[i] = u_a[i]$, $v_i[i] = v_a[i]$, and for each fixed i, $u_i[j] = u_c[i]$ and $v_i[j] = v_c[i]$ for any $j \neq i$.

Now we show how to derive $q^\star(u_a, v_a)$ using the tracker, and that all queries in the tracker are legitimate. We have that

$$q^\star(u_a, v_a) = q^\star(u_c, v_c) \setminus \left(\bigcup_{i=1}^{k} q^\star(u_i, v_i) \setminus q^\star(u_b, v_b) \right) \setminus q^\star(u_b, v_b)$$

Let $r = (1, -1, -1, \ldots, -1, k-1) \in \mathbb{R}^{k+2}$, then

$$\mathcal{M}(q^\star(u_a, v_a)) = r \cdot \big(\mathcal{M}(q^\star(u_c, v_c), \mathcal{M}(q^\star(u_1, v_1), \mathcal{M}(q^\star(u_2, v_2), \ldots, \mathcal{M}(q^\star(u_k, v_k), \mathcal{M}(q^\star(u_b, v_b)\big)^T$$

Moreover, $q^\star(u_b, v_b) \subseteq q^\star(u_c, v_c)$ and $q^\star(u_b, v_b) \subseteq q^\star(u_i, v_i)$ for all $1 \leqslant i \leqslant k$ hold. Hence, we have that $|q^\star(u_c, v_c)| \geqslant n_t$ and $|q^\star(u_i, v_i)| \geqslant n_t$ holds for all $1 \leqslant i \leqslant k$. □

Example 4. When $k = 1$ the data set contains n integers between one and n. Given any $q^\star(u, v)$ satisfying $0 < v - u < (n/3)$, we have that either $|q^\star(0, u-1)| \geqslant (n/3)$ or $|q^\star(v+1, d)| \geqslant (n/3)$ holds. Without loss of generality, if $|q^\star(0, u-1)| \geqslant (n/3)$ then we have that $q^\star(u, v) = q^\star(0, v) \setminus q^\star(0, u-1)$ holds, and $|q^\star(0, v)| \geqslant (n/3)$ and $|q^\star(0, u-1)| \geqslant (n/3)$ are both true. That is, we can infer $q^\star(u, v)$ with two legitimate queries.

Overlap size control. This method prevents users from asking queries with large intersections [18]. The intuition behind the method is that trackers rely on intersections between queries to *isolate* the targeted tuple, so prohibiting large intersections between queries will make inferences using trackers harder if at all possible. Specifically, the method assumes 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 satisfying $t \in q$ and subsequently $(n-1)/r$ or more queries whose union forms the complement of t with respect to q. Hence, no inference will be possible if less than $(n-1)/r + 1$ queries are answered. However, the converse is not true. That is, answering $(n-1)/r + 1$ or more queries does not necessarily cause inferences.

One may expect that the bound $(n-1)/r + 1$ will be improved if users are restricted to MDR queries, because this restriction makes inferences more difficult and likely an inference will demand more queries than in the case of arbitrary queries. In another word, the overlap size control would be more effective when applied to MDR queries, because it now allows more queries to be answered. However, Proposition 2 shows that this is not the case. The bound $(n-1)/r + 1$ is not improved (increased) by restricting users to MDR queries. Therefore, the overlap size control can answer only a small number of MDR queries, rendering most queries unanswerable.

Specifically, Proposition 2 shows the following fact. After asking an MDR query $q^\star(u, v)$, which includes the targeted tuple t and $n-1$ other tuples, it is always possible to find $|q^\star(u, v)| - 1$ other MDR queries satisfying the follows. First, the union of these queries forms the complement of t with respect to $q^\star(u, v)$. Second, the intersection between each of these queries and $q^\star(u, v)$ includes exactly one tuple. The existence of these queries shows that the bound $(n-1)/r + 1$ remains the same when applying the overlap size control to MDR queries. Example 5 illustrates the idea using the running example.

Proposition 2. *Given any $d \in \mathbb{R}^k$, $F = \mathcal{F}(d)$ and $C \subseteq F$, for any $q^\star(u, v)$ satisfying $|\{i : u[i] \neq v[i]\}| < k$ and any $t \in q^\star(u, v)$, there exists an $S \subseteq \mathcal{Q}_d$ such that $t = q^\star(u, v) \setminus \bigcup_{q \in S} q \cap q^\star(u, v)$. Moreover, for all $q \in S$ we have that $|q \cap q^\star(u, v)| = 1$.*

Proof. Suppose tuples in $q^\star(u, v)$ are in dictionary order and use t_i for the ith tuple. Without loss of generality suppose $t = t_1$ and $u[1] = v[1]$. Now we define a set S of queries as follows. For all $1 < i \leqslant |q^\star(u, v)| - 1$ let $u_i[1] = 1$, $v_i[1] = d[1]$, and for each fixed i, $u_i[j] = v_i[j] = t_i[j]$ for all $j > 1$. Let $S = \{q^\star(u_i, v_i)\}$. Because $q^\star(u_i, v_i) \cap q^\star(u, v) = t_i$ we have $t = q^\star(u, v) \setminus (\bigcup_{q \in S} q \cap q^\star(u, v))$. That is, the union of queries in S form the complement of t with respect to the given query $q^\star(u, v)$. □

Example 5. Consider the data set given in Table 3. To compromise $(1, 1)$, one first asks $q^\star((1, 1), (1, 3))$ that contains $(1, 1)$. Then to form the complement of $(1, 1)$ with respect to $q^\star((1, 1), (1, 3))$, queries $q^\star((1, 2), (2, 2))$ and $q^\star((1, 3), (2, 3))$ are asked. Asking one more query $q^\star((2, 2), (2, 3))$ would be sufficient for the intended compromise.

Audit expert. Chin et al. give a necessary and sufficient condition for safe arbitrary queries, namely, Audit Expert [12]. By regarding tuples and queries as a set system, the queries are safe iff the incidence matrix of the set system contains no unit row vector 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 $O(m^2 n)$. Using this condition *on-line* (after queries arrive) may incur unacceptable delay in answering queries, because m and n can be very large in practice. Moreover, the method requires tracking the entire history of queries asked by each user. Another way to employ the condition is to determine the compromiseability of queries off-line [6]. Although this condition applies to MDR queries, it is not efficient because it does not take into consideration the inherent redundancy among MDR queries, as illustrated by Example 6 (Section 6 further discusses this issue).

Example 6. Consider all MDR queries that can be formed on the data set in Table 3 (including those queries not shown in Table 4). There clearly exists redundancy among the MDR queries. For example, $q^\star((1, 1), (2, 2))$ is derivable from $q^\star((1, 1), (2, 1))$ and $q^\star((1, 2), (2, 2))$. Hence, if $q^\star((1, 1), (2, 1))$ and $q^\star((1, 2), (2, 2))$ are both safe then $q^\star((1, 1), (2, 2))$ must be safe. The converse is not true, that is, $q^\star((1, 1), (2, 2))$ is safe but $q^\star((1, 1), (2, 1)) = \{(1, 1)\}$ is not.

5.2. Finding maximal safe subsets of MDR queries

In addition to determining whether a collection of queries causes inferences, finding the maximum safe subset of an unsafe collection of queries is also an interesting

problem in inference control. Asking multiple queries in a batch, the user may prefer partial answers over a complete denial to all the queries. However, finding the maximum safe subset of arbitrary queries (the MQ problem) or sum-two queries (the RMQ problem) have both been shown as computationally infeasible [12].

A natural question is whether restricting users to MDR queries makes the above problem computationally feasible. Unfortunately, Theorem 1 shows this is not the case, because finding a maximum safe subset of MDR queries, namely, the *MDQ problem* is also NP-hard. The result is based on the fact that given any set of sum-two queries, we can always find a set of MDR queries such that the maximum safe subset of the former gives the maximum safe subset of the latter in polynomial time.

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

Proof. Chin et al. show the NP hardness of the *RMQ problem* [12]. We show that every instance of the RMQ problem is polynomially reducible to an instance of the MDQ problem.

Suppose an instance of the RMQ problem is given as

1. A data set of totally n tuples $C_0 = \{t_1, t_2, \ldots, t_n\}$.
2. A set of sum-two queries $S_0 = \{q^2(t_{i_1}, t_{j_1}), q^2(t_{i_2}, t_{j_2}), \ldots, q^2(t_{i_m}, t_{j_m})\}$ defined on C_0.

We construct an instance of the MDQ problem as

1. $d = (2, 2, \ldots, 2) \in \mathbb{R}^m$.
2. A data set C_1 with totally n tuples $s_1, s_2, \ldots, s_n$ satisfying that
   (a) $s_{i_1}[1] = s_{j_1}[1] = 1$, $s_{i_2}[2] = s_{j_2}[2] = 1, \ldots, s_{i_m}[m] = s_{j_m}[m] = 1$ (the subscripts i_x's and j_x's are given in the above RMQ problem).
   (b) For each fixed $x \in [1, m]$ and for all $y \neq i_x \wedge y \neq j_x$, $s_y[x] = 2$ holds.
3. The set of MDR queries $S_1 = \{q^\star(u_1, v_1), q^\star(u_2, v_2), \ldots, q^\star(u_m, v_m)\}$, where for all $1 \leqslant i \leqslant m$, $u_i[i] = v_i[i] = 1$, and for each fixed i, $u_j[i] = 1$, $v_j[i] = 2$ for all $j \neq i$.

We have that $q^\star(u_x, v_x) = \{s_{i_x}, s_{j_x}\}$ for all $1 \leqslant x \leqslant m$. Hence, for any $I \subseteq [1, m]$ we have that $\{q^2(t_{i_x}, t_{j_x}) : x \in I\}$ is safe iff $\{q^\star(u_x, v_x) : x \in I\}$ is safe. Consequently, the maximum safe subset of S_1 gives the maximum safe subset of S_0. □

Knowing that the MDQ problem is NP-hard, we may want to reduce the complexity with further restrictions on queries. We consider restricting users to an important class of MDR queries, namely, *data cubes* [22]. Notice that this restriction will only affect the MDR queries that users are allowed to ask, the data set and other assumptions (such as considering only SUMs and inferences of exact values) remain the same. Indeed, Definition 4 shows that we can rephrase concepts of a data cube using

MDR queries. That is, despite the different terminologies used by data cubes, we only consider them as special MDR queries. We demonstrate the concepts in Example 7. Corollary 1 then shows that the MDQ problem remains NP-hard for such special MDR queries.

Definition 4 (Data Cube). Given $d \in \mathbb{R}^k$, $F = \mathcal{F}(d)$ and $C \subseteq F$,

1. A skeleton query is any $q^\star(u, v)$ satisfying the condition that $u[i] \neq v[i]$ implies $u[i] = 1$ and $v[i] = d[i]$ for all $1 \leqslant i \leqslant k$. A skeleton query $q^\star(u, v)$ is called a j-star query ($1 \leqslant j \leqslant k$) if $|\{i : i \in [1, k], u[i] \neq v[i]\}| = j$.
2. For any non-empty $J \subseteq [1, k]$, let $j = |J|$. The set Q of j-star queries satisfying that $q^\star(u, v) \in Q$ iff $\{i : i \in [1, k], u[i] \neq v[i]\} = J$ is called a (j-star) cuboid.
3. The data cube is the union of all cuboids (or equivalently all skeleton queries).

Example 7. Table 3 includes 1-star cuboids $\{q^\star((1, 1), (1, 4)), q^\star((2, 1), (2, 4))\}$ and $\{q^\star((1, 1), (2, 1)),\ q^\star((1, 2),\ (2, 2)), q^\star((1, 3), (2, 3)), q^\star((1, 4), (2, 4))\}$. There is only one 2-star cuboid, which 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. *The union of some cuboids.*
3. *A data cube.*

Proof. Because the set of MDR queries constructed in the proof of Theorem 1 are actually skeleton queries, we only need to show MDQ is NP-hard under the second and third restrictions. Suppose the instance of the RMQ problem is given same as in the proof of Theorem 1. We first construct an instance of the MDQ problem under the restriction that the set of MDR queries is the union of some cuboids. The data set C_1 and the set of MDR queries S_1 are given as follows.

1. $d = (n - 1, n - 1, \ldots, n - 1) \in \mathbb{R}^m$.
2. A data set C_1 with totally m tuples $s_1, s_2, \ldots, s_m$, where for all $1 \leqslant x \leqslant m$, $s_{i_x}[x] = s_{j_x}[x] = 1$ and $1 < s_y[i] < s_z[i]$ for any $y < z$ and $y, z \in [1, n] \setminus \{i_x, i_y\}$.
3. $S_t = \{q^\star(u_1, v_1), q^\star(u_2, v_2), \ldots, q^\star(u_m, v_m)\}$, where for all $1 \leqslant i \leqslant m$, $u_i[i] = v_i[i] = 1$, and for each fixed i, $u_j[i] = 1, v_j[i] = n - 1$ for all $j \neq i$.
4. $S_1 = \bigcup_{i=1}^m Q_i$, where each Q_i is the cuboid containing $q^\star(u_i, v_i)$.

For any $q \in \bigcup_{i=1}^m Q_i \setminus S_t$ we have that $|q| = 1$. Hence, trivially the maximal safe subset of S_1 is a subset of S_t. For any $1 \leqslant x \leqslant m$ we have that $q^\star(u_x, v_x) =$

$\{s_{i_x}, s_{j_x}\}$. Hence, for any $I \subseteq [1,m]$, $\{q^2(t_{i_x}, t_{j_x}) : x \in I\}$ is safe iff $\{q^\star(u_x, v_x) : x \in I\}$ is safe. Consequently, the maximal safe subset of S_1 gives the maximal safe subset of S_0.

Next we modify this instance of the MDQ problem to the third restriction as follows.

1. $d = (n+1, n+1, \ldots, n+1) \in \mathbb{R}^m$.
2. $C_1 = \{s_1, s_2, \ldots, s_n, s_{n+1}, s_{n+2}\}$, where $s_{n+1} = (n, n, \ldots, n)$ and $s_{n+2} = (n+1, n+1, \ldots, n+1)$.
3. $S_t = \{q^\star(u_1, v_1), q^\star(u_2, v_2), \ldots, q^\star(u_m, v_m)\}$, where for all $1 \leqslant i \leqslant m$, $u_i[i] = v_i[i] = 1$ and for each fixed i, $u_j[i] = 1, v_j[i] = n+1$ for all $j \neq i$.
4. Q_i is the cuboid containing $q^\star(u_i, v_i)$ for all $1 \leqslant i \leqslant m$.
5. S_1 is the data cube.

Suppose S_{max1} is the maximal safe subset of S_1. Then similarly S_{max1} does not contain any $q \in \bigcup_{i=1}^m Q_i \backslash S_t$. Moreover, S_{max1} does not contain any j-star query for all $j < m-1$. As we will show shortly, S_{max1} contains the m-star query $q^\star(u_\star, v_\star)$, where $u_\star = (1, 1, \ldots, 1)$ and $v_\star = (n+1, n+1, \ldots, n+1)$. Hence, we have that $S_{max1} \subseteq S_t \cup \{q^\star(u_\star, v_\star)\}$ and $q^\star(u_\star, v_\star) \in S_{max1}$. For all $1 \leqslant x \leqslant m$, we have that $q^\star(u_x, v_x) = \{s_{i_x}, s_{j_x}\}$. Hence, for any $I \subseteq [1,m]$, $\{q^2(t_{i_x}, t_{j_x}) : x \in I\}$ is safe iff $\{q^\star(u_x, v_x) : x \in I\}$ is safe. Consequently, finding S_{max1} gives the maximal safe subset of S_0.

It remains to show that $q^\star(u_\star, v_\star) \in S_{max1}$. We do so by contradiction. Suppose $q^\star(u_\star, v_\star) \notin S_{max1}$ and $S_{max1} \cup \{q^\star(u_\star, v_\star)\}$ compromises some $t \in C_1$. Then we have that $S_{max1} \subseteq S_t$. Suppose that $|S_{max1}| = l$. Then there exists $r \in \mathbb{R}^{l+1}$ such that $r \cdot \mathcal{M}(\{q^\star(u_\star, v_\star)\} \cup S_{max1})^T = \mathcal{M}(t)$ holds. Let $r' = (r[2], r[3], \ldots, r[l])$. Then

$$r[1] \cdot \mathcal{M}(q^\star(u_\star, v_\star))^T + r' \cdot \mathcal{M}(S_{max1})^T = \mathcal{M}(t)$$

We have that $s_{n+1}, s_{n+2} \notin \bigcup_{q \in S_{max1}} q$ because $S_{max1} \subseteq S_t$. Moreover $\mathcal{M}(q^\star(u_\star, v_\star)) = \mathcal{M}(s_{n+1}) + \mathcal{M}(s_{n+2}) + \sum_{i=1}^n \mathcal{M}(s_i)$. We have that

$$r[1] \cdot \mathcal{M}(s_{n+1})^T + r[1] \cdot \mathcal{M}(s_{n+2})^T + \sum_{i=1}^n x_i \cdot \mathcal{M}(s_i)^T = \mathcal{M}(t)$$

holds for some $x_i \in \mathbb{R}$, $i = 1, 2, \ldots, n$.

There are two cases. First suppose $t \in \{s_1, s_2, \ldots, s_n\}$. Then we have that $r[1] = 0$. Consequently, we have that $r' \cdot \mathcal{M}(S_{max1})^T = \mathcal{M}(t)$, which contradicts the assumption that S_{max1} is safe. Secondly, suppose $t \in \{s_{n+1}, s_{n+2}\}$. Without loss of generality, assume $t = s_{n+1}$, which leads to the contradiction that $r[1] = 1$ and $r[1] = 0$. Hence, we have proved that $q^\star(u_\star, v_\star) \in S_{max1}$. $\square$

6. Parity-based inference control

Section 6.1 shows how to determine whether even MDR queries are safe. Section 6.2 studies what other queries can be answered without causing inferences. Section 6.3 then discusses how to find large safe subsets of even MDR queries.

6.1. Even MDR queries

Following the model introduced in Section 4, we denote the collection of all even MDR queries defined on a given data set as $\mathcal{Q}_e$. In order to efficiently determine whether $\mathcal{Q}_e$ causes inferences, we show that there exists a special subset of $\mathcal{Q}_t$ (the collection of all sum-two queries), denoted as $\mathcal{Q}_{dt}$, satisfying $\mathcal{Q}_{dt} \equiv_d \mathcal{Q}_e$ ($\equiv_d$ denotes the equivalence relation formalized in Definition 3). By Definition 3, we can then determine whether $\mathcal{Q}_e$ is safe by checking if $\mathcal{Q}_{dt}$ is safe. Intuitively, the latter incurs less complexity because $\mathcal{Q}_{dt}$ contains less redundant queries than $\mathcal{Q}_e$ does.

First, two natural but untrue conjectures are $\mathcal{Q}_e \equiv_d \mathcal{Q}_t$ and $\mathcal{Q}_e \equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$. The first says that the collection of all even MDR queries is equivalent to the collection of all sum-two queries, and the second says the collection of all even MDR queries is equivalent to those even MDR queries that are at the same time sum-two queries. To see why the former is untrue, consider the counter-example with the one-dimensional data set $C = \{1, 2, 3\}$. 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 8 gives a counter-example to $\mathcal{Q}_e \equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$.

Example 8. Figure 2 shows $\mathcal{Q}_e \not\equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$ because $q^\star((1, 1), (2, 4)) \in \mathcal{Q}_e$ is not derivable from $\mathcal{Q}_e \cap \mathcal{Q}_t$.

The key observation from Example 8 is that $\mathcal{Q}_e \not\equiv_d \mathcal{Q}_e \cap \mathcal{Q}_t$ due to even queries like $q^\star((1, 1), (2, 4))$. 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, no matter how we begin to pair the

The Data Set C

	1	2	3	4
1	(1,1)	(1,2)	(1,3)	
2		(2,2)	(2,3)	(2,4)

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

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

Fig. 2. An example showing $\mathcal{Q}_e$ not equivalent to $\mathcal{Q}_e \cap \mathcal{Q}_t$.

tuples in these odd queries as MDR queries (and at the same time sum-two queries), we end up with some tuples that cannot be paired as MDR queries, such as $(1, 3)$ and $(2, 4)$. This shows that the intersection between $\mathcal{Q}_e$ and $\mathcal{Q}_t$ is not enough for deriving every query in $\mathcal{Q}_e$.

On the other hand, suppose that from $\mathcal{Q}_e \cap \mathcal{Q}_t$ we can derive each odd query up to the *last tuple* (for example, $(1, 3)$ and $(2, 4)$). Then we can pair the adjacent last tuples of all the odd queries by adding additional sum-two queries to $\mathcal{Q}_e \cap \mathcal{Q}_t$ (for example, $q^2((1, 3), (2, 4))$). Hence, we can now 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 the process. We demonstrate this in Example 9 and generalize the result in Theorem 2.

Example 9. In Example 8, 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, we can derive $q^2((1, 3), (2, 4))$ 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 data set C, there exists $\mathcal{Q}_{dt} \subseteq \mathcal{Q}_t$ such that $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$ holds.*

Proof. To justify the existence of a $\mathcal{Q}_{dt}$ satisfying $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$, we first construct it using a procedure shown in Fig. 3. Roughly speaking, the procedure calls a subroutine *Sub_QDT* with each even MDR query as input. The subroutine adopts a divides-and-conquer approach in pairing all tuples included in the query. The subroutine recursively divides the input query along each dimension into sub-queries until each sub-query includes only a single tuple. The subroutine then conquers by adding pairs of tuples returned by adjacent sub-queries into the result $\mathcal{Q}_{dt}$, and returns the remaining tuple (if there is any) to an upper-level call. The final result $\mathcal{Q}_{dt}$ is a special subset of sum-two queries.

We then show that the result $\mathcal{Q}_{dt}$ does satisfy the property $\mathcal{Q}_e \equiv_d \mathcal{Q}_{dt}$. In the following discussion we assume that $d \in \mathbb{R}^k$, $F = \mathcal{F}(d)$, $C \subseteq F$, and any $S \subseteq C$ is sorted in dictionary order. For $i = 1, 2, \ldots, |S|$, we use $S[i]$ for the ith tuple in S. For any $u, v \in F$ satisfying $u \leqslant v$ and $q^\star(u, v) \in \mathcal{Q}_e$, use S_{uv} to denote the set of sum-two queries added to $\mathcal{Q}_{dt}$ by calling the subroutine *Sub_QDT* in Fig. 3.

In order to prove $\mathcal{Q}_e \preceq \mathcal{Q}_{dt}$, we show that for any $u \leqslant v$ and $q^\star(u, v) \in \mathcal{Q}_e$, $q^\star(u, v) \preceq S_{uv}$ holds. Specially, we show that $q^\star(u, v) = \bigcup_{q \in S_{uv}} q$. Because $q_1 \cap q_2 = \phi$ holds for any $q_1, q_2 \in S_{uv}$, it then follows that $\mathcal{M}(q^\star(u, v)) = r \cdot \mathcal{M}(S_{uv})^T$, where $r = (1, 1, \ldots, 1) \in \mathbb{R}^{|S_{uv}|}$. We do so by mathematical induction on $|I|$, where $I = \{i : i \in [1, k], u[i] < v[i]\}$.

The Inductive Hypothesis: For $|I| = 0, 1, \ldots, k$, if $q^\star(u, v) \in \mathcal{Q}_e$, then $q^\star(u, v) = \bigcup_{q \in S_{uv}} q$. Otherwise, $q^\star(u, v) = (\bigcup_{q \in S_{uv}} q) \cup \{\textit{Sub\_QDT}(C, u, v, \mathcal{Q}_{dt})\}$.

The Base Case: For $|I| = 0$, we have that $u = v$, and $q^\star(u, v) = \{u\}$. Because $I = \phi$, the subroutine *Sub_QDT* in Fig. 3 returns u at the second step, with $S_{uv} = \phi$. Hence, $q^\star(u, v) = \phi \cup \{u\}$, validating the base case of our inductive hypothesis.

Procedure *QDT*
Input: A data set C
Output: A set of sum-two queries $\mathcal{Q}_{dt}$
Method:
1. **Let** $\mathcal{Q}_{dt} = \phi$;
2. **For** any $q^\star(u, v) \in \mathcal{Q}_e$, where $u < v$
   Let $< t_{odd}, \mathcal{Q}_{dt} >$=*Sub_QDT*$(C, u, v, \mathcal{Q}_{dt})$;
3. **Return** $\mathcal{Q}_{dt}$;

Subroutine *Sub_QDT*
Input: The data set C, tuples u and v ($u \leqslant v$), and $\mathcal{Q}_{dt}$
Output: A pair $< t_{odd}, \mathcal{Q}_{dt} >$
Method:
1. **Let** $I = \{i : i \in [1, k], u[i] < v[i]\}$ and $m = \max(I)$;
2. **If** $I = \phi$ //Stop when $u = v$
   Return $< u, \mathcal{Q}_{dt} >$;
3. **For** $i = u[m]$ to $v[m]$ //Divide
   Let $u_i[j] = u[j]$ and $v_i[j] = v[j]$ for all $j \in I \setminus \{m\}$, and $u_i[m] = v_i[m] = i$;
   If $q^\star(u_i, v_i) \neq \phi$
   Let $< t_i, \mathcal{Q}_{dt} >=$ *Sub_QDT*$(C, u_i, v_i, \mathcal{Q}_{dt})$; //Recursion
4. **For** $i = u[m]$ to $v[m]$ //Conquer
   If $t_i \neq null$
   Let $j = \min\{j : j > i, t_j \neq null \vee j > v[m]\}$;
   If $j > v[m]$
   Return $< t_i, \mathcal{Q}_{dt} >$;
   Else
   Let $\mathcal{Q}_{dt} = \mathcal{Q}_{dt} \cup \{q^2(t_i, t_j)\}$ and $i = j$;
   Return $< null, \mathcal{Q}_{dt} >$;
5. **Return** $< null, \mathcal{Q}_{dt} >$;

Fig. 3. Procedure QDT.

The Inductive Case: Suppose the inductive hypothesis holds for $|I| = 0, 1, \ldots, j < k$, we show that it holds for $|I| = j+1$. Let u and v satisfy that $u < v$ and $|I| = j+1$, where $I = \{i : i \in [1, k], u[i] < v[i]\}$.

For all $u[m] \leqslant i \leqslant v[m]$, where $m = \max(I)$, the pair(u_i, v_i) defined in the subroutine *Sub_QDT* satisfy $|\{i : i \in [1, k], u_i[i] < v_i[i]\}| = j$. Hence, when the subroutine *Sub_QDT* recursively calls itself with the input $(C, u_i, v_i, \mathcal{Q}_{dt})$, the inductive hypothesis holds inside the recursion. Let $J = \{i : i \in [u[m], v[m]], q^\star(u_i, v_i) \notin \mathcal{Q}_e\}$ and $J' = [u[m], v[m]] \setminus J$. Because of the inductive hypothesis, $q^\star(u_i, v_i) = \bigcup_{q \in S_{u_i v_i}} q$ holds for all $i \in J'$, and conversely $q^\star(u_i, v_i) = (\bigcup_{q \in S_{u_i v_i}} q) \cup \{t_i\}$ for all $i \in J$, where $t_i =$ *Sub_QDT*$(C, u_i, v_i, \mathcal{Q}_{dt})$.

If $q^\star(u, v) \in \mathcal{Q}_e$, we have that $|J|$ is even. For $i = 1, 2, \ldots, \frac{|J|}{2}$, $q^2(t_{2i-1}, t_{2i}) \in S_{uv}$ holds because of Step 4 of *Sub_QDT*. Hence, we have that

$$q^\star(u,v) = \bigcup_{i=u[m]}^{v[m]} q^\star(u_i,v_i)$$

$$= \left(\bigcup_{i=u[m]}^{v[m]}\left(\bigcup_{q\in S_{u_iv_i}} q\right)\right) \cup \left(\bigcup_{i=1}^{\frac{|J|}{2}}\{q^2(t_{2i-1},t_{2i})\right) = \bigcup_{q\in S_{uv}} q$$

Conversely, if $q^\star(u,v) \in \mathcal{Q}_d \backslash \mathcal{Q}_e$, we have that $|J|$ is odd. For $i = 1, 2, \ldots, \frac{|J|-1}{2}$, we have that $q^2(t_{2i-1}, t_{2i}) \in S_{uv}$. Furthermore, we have that *Sub_QDT*$(C, u, v, \mathcal{Q}_{dt}) = t_{|J|} \notin S_{uv}$. Hence, the following holds:

$$q^\star(u,v) = \bigcup_{i=u[m]}^{v[m]} q^\star(u_i,v_i) = S_{uv} \cup \{Sub\_QDT(C,u,v,\mathcal{Q}_{dt})\}$$

This proves the inductive case of our inductive hypothesis.

In order to prove $\mathcal{Q}_{dt} \preceq \mathcal{Q}_e$, we show that for any $q \in \mathcal{Q}_{dt}$, $q \preceq \mathcal{Q}_e$ holds. Suppose in the subroutine *Sub_QDT* in Fig. 3 a sum-two query $q^2(t_i, t_j)$ is added to $\mathcal{Q}_{dt}$, where $u[m] \leqslant i < j \leqslant v[m]$.

We only need to show that $q^\star(u_i, v_i) \setminus \{t_i\} = \bigcup_{q\in S_i} q$ and similarly $q^\star(u_j, v_j) \setminus \{t_j\} = \bigcup_{q\in S_j} q$, where $S_i, S_j \subseteq \mathcal{Q}_e$ and u_i, v_i, u_j, v_j are defined in Fig. 3. Because then we have

$$q^2(t_i,t_j) = q^\star(u_i,v_j) \setminus \left(\left(\bigcup_{l=i+1}^{j-1} q^\star(u_l,v_l)\right) \cup \left(\bigcup_{q\in S_i\cup S_j} q\right)\right)$$

This implies that $q^2(t_i, t_j) \preceq \mathcal{Q}_e$, because $q^\star(u_i, v_j) \in \mathcal{Q}_e$ and $q^\star(u_l, v_l) \in \mathcal{Q}_e$ for any $i < l < j$. We do so by induction on $|I|$.

The Inductive Hypothesis: For any $i \in [u[m], v[m]]$, if $t_i \neq null$ then $q^\star(u_i, v_i) \setminus \{t_i\} = \bigcup_{\forall q\in S_i} s(q)$, for some $S_i \subseteq \mathcal{Q}_e$, where $u[m]$, $v[m]$, t_i are defined in Fig. 3.

The Base Case: For $|I| = 0$, we have that $u = v$, $i = u[m]$, and $t_i = u$. Hence, $q^\star(u, u) \setminus \{u\} = \phi$. The base case of the inductive hypothesis trivially holds with $S_i = \phi$.

The Inductive Case: Suppose the inductive hypothesis holds for all $|I| = 0, 1, \ldots, j$ for some $0 \leqslant j < k$, we show that it holds for $j + 1$. Because the subroutine *Sub_QDT* recursively calls itself, inside the recursion we have that $|I| = j$. Suppose the inputs to the recursive call are C, u, v and $q^\star(u, v) \notin \mathcal{Q}_e$. We have that $q^\star(u,v) = q^\star(u, v_{l-1}) \cup q^\star(u_{l+1}, v) \cup q^\star(u_l, v_l)$ if $l < v[m]$, or $q^\star(u,v) = q^\star(u, v_{l-1}) \cup q^\star(u_l, v_l)$ if $l = v[m]$. Moreover, because of the inductive hypothesis we have that $q^\star(u_l, v_l) \setminus \{t_l\} = q^\star(u_l, v_l) \setminus \{t_l\} = \bigcup_{\forall q\in S_l} s(q)$ holds for some $S_l \subseteq \mathcal{Q}_e$. Hence, we have $q^\star(u,v) \setminus \{t_l\} = \bigcup_{\forall q\in S} s(q)$, where $S = S_l \cup \{q^\star(u, v_{l-1}), q^\star(u_{l+1}, v)\}$ if $l < v[m]$,

or $Q = Q_l \cup \{q^\star(u, v_{l-1})\}$ if $l = v[m]$. Because $q^\star(u, v) \notin \mathcal{Q}_e$, we have that $|\{i : i \in [u[m], v[m]], t_i \neq null\}|$ is odd. Hence, we have $q^\star(u, v_{l-1}), q^\star(u_{l+1}, v) \in \mathcal{Q}_e$. Consequently, $S \subseteq \mathcal{Q}_e$ holds. Because t_l =*Sub_QDT*$(C, u, v, \mathcal{Q}_{dt})$, this validates the inductive case of our inductive hypothesis. □

Example 10. To demonstrate how the procedure *QDT* constructs $\mathcal{Q}_{dt}$, we consider Example 9 again. Without loss of generality, assume the procedure calls the subroutine *Sub_QDT* in the following order (the order does not affect the final result). First, each query in $\mathcal{Q}_e \cup \mathcal{Q}_t$ is the input. The subroutine *Sub_QDT* divides each such input into its two tuples (step 3). Upon conquering, the subroutine adds the query itself into $\mathcal{Q}_{dt}$ (step 4). Second, the procedure calls the subroutine *Sub_QDT* with $q^\star((1, 2), (2, 3))$ as the input and adds $q^2((1, 2), (1, 3))$ and $q^2((2, 2), (2, 3))$ to $\mathcal{Q}_{dt}$ (which are already there). Finally, when $q^\star((1, 1), (2, 4))$ is the input, the subroutine divides the query into $q^\star((1, 1), (1, 3))$ and $q^\star((2, 2), (2, 4))$, and upon conquering the subroutine adds $q^2((1, 1), (1, 2))$, $q^2((2, 2), (2, 3))$, and $q^2((1, 3), (2, 4))$ to $\mathcal{Q}_{dt}$ (the first two are already in $\mathcal{Q}_{dt}$). The final result of $\mathcal{Q}_{dt}$ will be exactly as described in Example 9.

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

Definition 5. For any $S \subseteq \mathcal{Q}_{dt}$, 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*.

Figure 4 illustrates the QDT graph for our running example. It has been shown 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 odd number of edges) [8]. In Fig. 4 it is easy to observe that an odd cycle exists, so the graph is not a bipartite graph. Whether a graph is bipartite can be decided with a breadth-first search (BFS) on $G(C, \mathcal{Q}_{dt})$, taking time $O(n + |\mathcal{Q}_{dt}|) = O(m + n)$. Hence, the complexity of determining the compromiseability of $\mathcal{Q}_e$ is dominated by the construction of $\mathcal{Q}_{dt}$, which is $O(mn)$. Notice that from Section 5 we know that directly applying the condition of Audit Expert has the complexity of $O(m^2 n)$ [12]. Therefore, our solution is more efficient than Audit Expert with respect to MDR queries.

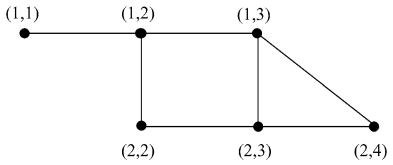

Fig. 4. An example of QDT graph.

6.2. Beyond even MDR queries

Characterizing the QDT Graph. Lemma 1 gives some properties of the QDT graph that are useful for the rest of this section. The first property is straightforward. The second property is based on the intuition that if any two tuples t_1, t_2 in the data set are not *close enough* (that is, $q^\star(t_1, t_2) \notin \mathcal{Q}_{dt}$), then we can find another tuple $t_3 \in q^\star(t_1, t_2)$, such that $q^\star(t_1, t_3) \in \mathcal{Q}_{dt}$ and t_3 is closer to t_1 than t_2 is. 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* $|q^\star(t_1, t_2)| > 2$, *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.*

Proof. The first claim holds as *Sub_QDT*$(C, u_0, v_0, \mathcal{Q}_{dt})$ will be called for all u_0, v_0 satisfying $q^\star(u_0, v_0) = \{u_0, v_0\}$.

For the second claim, suppose $t_3 \neq t_1$, $t_3 \neq t_2$ and $|q^\star(t_1, t_3)| > 2$. Then $q_2 \notin q^\star(t_1, t_3)$ holds. For otherwise, for any $i \in [1, k]$ we have $\min\{t_1[i], t_2[i]\} \leqslant t_3[i] \leqslant \max\{t_1[i], t_2[i]\}$ and $\min\{t_1[i], t_3[i]\} \leqslant t_2[i] \leqslant \max\{t_1[i], t_3[i]\}$, and hence $t_2 = t_3$ contradicting our assumption. Consequently, we have that $|q^\star(t_1, t_3)| < |q^\star(t_1, t_2)|$. Let $t_4 \in q^\star(t_1, t_3)$ satisfying $t_4 \neq t_1$ and $t_4 \neq t_3$. We can repeat the same argument by replacing t_3 with t_4 and so on, until $|q^\star(t_1, t)| = 2$ for some $t \in q^\star(t_1, t_2)$. This together with the first claim of Lemma 1 justifies the second claim.

We prove the third claim by contradiction. Suppose G_1 and G_2 are any two connected components of any $G(C, \mathcal{Q}_{dt})$, and let $t_1 \in V(G_1)$ (the vertex set of G_1), $t_2 \in V(G_2)$. By the first claim of Lemma 1 we have that $|q^\star(t_1, t_2)| > 2$. By the second claim there exists $t_3 \in q^\star(t_1, t_2)$ such that $q^\star(t_1, t_3) \in \mathcal{Q}_{dt}$ and hence $t_3 \in V(G_1)$. Similarly as stated above, $t_1 \notin q^\star(t_3, t_2)$ and hence $|q^\star(t_1, t_2)| > |q^\star(t_3, t_2)|$. Repeat above reasoning with t_1 replaced by t_3 and so on, until that for some t we have $|q^\star(t, t_2)| = 2$, and hence $q^\star(t, t_2) \in \mathcal{Q}_{dt}$ by the first claim. But then G_1 and G_2 are connected because $t \in V(G_1)$, contradicting our assumption. □

Properties of $\mathcal{Q}_{dt}$. Although we have shown that $\mathcal{Q}_{dt} \equiv_d \mathcal{Q}_e$, $\mathcal{Q}_{dt}$ may be neither the smallest nor the largest subset of $\mathcal{Q}_t$ equivalent to $\mathcal{Q}_e$. The smallest subset can be obtained by removing all 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

$\mathcal{Q}_e$, because this reveals the extent of knowledge users may ever learn from query results. According to Lemma 2, a safe $\mathcal{Q}_e$ essentially allows users to sum any two tuples from difference 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 as that of $G(C, \mathcal{Q}_{dt})$. These results are formally stated in Lemma 2.

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 $\mathcal{M}(t_1) - \mathcal{M}(t_2) = r \cdot \mathcal{M}(\mathcal{Q}_{dt})$.*

Proof. $\mathcal{Q}_{dt} \preceq \mathcal{Q}^\star_{dt}$ is trivial because $\mathcal{Q}_{dt} \subseteq \mathcal{Q}^\star_{dt}$. We only need to show $\mathcal{Q}^\star_{dt} \preceq \mathcal{Q}_{dt}$. By Lemma 1, $G(C, \mathcal{Q}_{dt})$ is a connected bipartite. Hence, there exists a path containing odd number of edges between any $t_1 \in C_1$ and $t_0 \in C_2$. Let it be $S = \{q^2(t_1, t_2), q^2(t_2, t_3), \ldots, q^2(t_{2n}, t_{2n+1}), q^2(t_{2n+1}, t_0)\}$, where $n \geqslant 0$. We have that $\mathcal{M}(q^2(t_1, t_0)) = ((-1)^0, (-1)^1, (-1)^2, \ldots, (-1)^{2n}) \cdot \mathcal{M}(S)^T$. Hence, $q^2(t_1, t_0) \preceq \mathcal{Q}_{dt}$.

Because $\mathcal{Q}^\star_{dt}$ corresponds to the complete bipartite graph (that is, a bipartite graph whose edge set includes all the edges that incident two vertices from different color classes) with bipartition (C_1, C_2), any proper superset S of $\mathcal{Q}^\star_{dt}$ is not a bipartite. Hence, S cannot be safe, and consequently $S \not\preceq \mathcal{Q}_{dt}$.

For any $t_1, t_{11} \in S_1$, because $G(C, \mathcal{Q}_{dt})$ is connected there must exists $t_2 \in S_2$ such that $q^2(t_1, t_2) \in \mathcal{Q}_{dt}$. Taken together with $q^2(t_2, t_{11}) \preceq \mathcal{Q}_{dt}$, we have that the third claim holds. □

Odd MDR Queries. Now that we can determine the compromiseability 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 in the returned pair. 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 disjoint even MDR queries together with a singleton set $\{t\}$. This is formally stated in Corollary 2 and illustrated in Example 11.

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 $|\{i : u[i] \neq v[i]\}| = j$, there exists $q^\star(u_i, v_i) \in \mathcal{Q}_e$ for all $1 \leqslant i \leqslant 2j-1$, such that $|q^\star(u, v) \setminus \bigcup_{i=1}^{2j-1} q^\star(u_i, v_i)| = 1$ and $q^\star(u_i, v_i) \cap q^\star(u_l, v_l) = \phi$ for all $1 \leqslant i < l \leqslant 2j-1$.*

Proof. Suppose we call subroutine *Sub_QDT* in Fig. 3 with input $(q^\star(u,v), u, v, \mathcal{Q}_{dt})$ and let the output be t_{odd}. For $i = 1, 2, \ldots, k$ and $l = 1, 2, 3, 4$, define tuples u_{il} as:

1. $u_{il}[j] = t_{odd}[j]$ for all $j > i$ and $l = 1, 2, 3, 4$.
2. $u_{i1}[i] = u[i]$, $u_{i2}[i] = u_{odd}[i] - 1$, $u_{i3}[i] = t_{odd}[i] + 1$ and $u_{i4}[i] = v[i]$.
3. $u_{i1}[j] = u_{i3}[j] = u[j]$ and $u_{i2}[j] = u_{i4}[j] = v[j]$ for all $j < i$.

We then have that $q^\star(u,v) = \bigcup_{i=1}^{k}(q^\star(u_{i1}, u_{i2}) \cup q^\star(u_{i3}, u_{i4})) \cup \{t_{odd}\}$ and all the $q^\star(u_{il}, u_{il})$'s are disjointed. Because $q^\star(u_{13}, u_{14}) = \phi$, we have totally 2k−1 disjointed even MDR queries. □

Example 11. In Table 2, use $q^\star((1,1),(2,3))$ as the input of *Sub_QDT* gives $(1,3)$. $q^\star((1,1),(2,3))$ 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 even MDR queries by only one tuple. This difference is usually tolerable because most users of MDR queries are interested in patterns and trends instead of individual values. The odd query can thus be approximately answered using the even ones.

Arbitrary Queries. We know the implication of $\mathcal{Q}_e$ in terms of sum-two queries from Lemma 2. Hence, we can decide which arbitrary queries can be answered in addition to $\mathcal{Q}_e$. 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 compromiseability 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 $|q \cap C_1| = |q \cap C_2|$, where (C_1, C_2) is the bipartition of $G(C, \mathcal{Q}_{dt})$.*

Proof. If $|c \cap C_1| = |c \cap C_2|$, then $c = \bigcup_{q \in S} q$ for some $S \subseteq \mathcal{Q}^\star_{dt}$. Hence, $c \preceq \mathcal{Q}^\star_{dt}$ and consequently $c \preceq \mathcal{Q}_e$.

We prove the only if part by contradiction. Without loss of generality suppose $|c \cap C_1| > |c \cap C_2|$ and $c \preceq \mathcal{Q}_e$. Then $c = c_0 \cup c_1$, where c_0, c_1 satisfy that $c_0 \cap c_1 = \phi$, $|c_0 \cap C_1| = |c_0 \cap C_2|$ and $c_1 \subseteq C_1$. Then we have that $c_0 \preceq \mathcal{Q}_e$ and hence $V(c_1) \preceq \mathcal{Q}_e$ follows. Suppose $c_1 = \{t_0, t_1, \ldots, t_n\}$ where $n \geqslant 1$. Then by the third claim of Lemma 2 we have that $\mathcal{M}(t_0) - \mathcal{M}(t_i) = r_i \cdot \mathcal{M}(\mathcal{Q}_{dt})^T$ holds for all $1 \leqslant i \leqslant n$, where each $r_i \in \mathbb{R}^{|\mathcal{Q}_{dt}|}$. By adding the two sides of all the n equation we have that $n \cdot \mathcal{M}(t_0) = \sum_{i=1}^{n} \mathcal{M}(t_i) + \sum_{i=1}^{n} r_i \cdot \mathcal{M}(\mathcal{Q}_{dt})^T$. Let $\mathcal{M}(c_1) = r \cdot \mathcal{M}(\mathcal{Q}_{dt})^T$, where $r \in \mathbb{R}^{|\mathcal{Q}_{dt}|}$. Because $\sum_{i=1}^{n} \mathcal{M}(t_i) = \mathcal{M}(c_1) - \mathcal{M}(t_0) = r \cdot \mathcal{M}(\mathcal{Q}_{dt})^T - \mathcal{M}(t_0)$ we have that $(n+1)\mathcal{M}(t_0) = \sum_{i=1}^{n} r_i \cdot \mathcal{M}(\mathcal{Q}_{dt})^T + r \cdot \mathcal{M}(\mathcal{Q}_{dt})^T$. Hence, t_0 is compromised by $\mathcal{Q}_{dt}$ contradicting our assumption that $c \preceq \mathcal{Q}_e$. □

6.3. Unsafe even MDR queries

When the collection of even MDR queries $\mathcal{Q}_e$ is not safe, we may want to find its safe subsets. Section 5.2 shows that finding maximum safe subsets of queries is infeasible even for queries of restricted forms, such as sum-two queries and data cubes. Hence, we turn to large but not necessarily maximum safe subsets. Recall that Section 6.1 determines the compromiseability of $\mathcal{Q}_e$ by finding an equivalent collection of sum-two queries $\mathcal{Q}_{dt}$. If we could establish the equivalence between their subsets, we would be able to extend the results in Section 6.1 to those subsets. However, for arbitrary subsets of $\mathcal{Q}_e$ or $\mathcal{Q}_{dt}$, such equivalence may not exist, as illustrated by Example 12.

Example 12. Consider $\mathcal{Q}_{dt}$ in Example 9, 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 $q^2((1,3),(2,4)) \preceq_d S_e$, S_e must contain $q^\star((1,1),(1,2))$, but then $q^\star((1,1),(1,2)) \not\preceq_d S_{dt}$ leads to a contradiction. Hence, S_{dt} cannot be equivalent to any subset of $\mathcal{Q}_e$. Similarly, we have that $\mathcal{Q}_e \setminus \{q^\star((1,1),(1,2))\}$ is not equivalent to any subset of $\mathcal{Q}_{dt}$.

If we regard an MDR query as a smaller data set C', then the equivalence given in Theorem 2 must also hold in this new data set as follows. First, we say an even MDR query is *defined on* C' if it is a subset of C'. The collection of all even MDR queries defined on C' is then equivalent to the set of sum-two queries produced by the procedure *Sub_QDT* with those even MDR queries as the inputs. This result can be extended to any subset of the data set, because we can always regard the subset as the union of multiple disjoint MDR queries. Given any $S \subseteq \mathcal{Q}_e$, we first find a subset C' of the data set such that all the queries defined on C' are included in S. This allows us to regard C' as a new data set and to apply the above discussion to establish the equivalence. Similar result holds for any $S \subseteq \mathcal{Q}_{dt}$. Those are formally stated in Proposition 3 and illustrated in Example 13.

Proposition 3.

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* $S_{dt} \equiv_d S_e$.

Proof. We only need to justify the first claim. For any $q^2(u_0,v_0) \in S_{dt}$, suppose $q^2(u_0,v_0) \in \mathcal{Q}_{dt}$ because we know that $q^\star(u_1,v_1) \in S_e$. Then $\{q^\star(u,v) : q^\star(u,v) \in \mathcal{Q}_e \wedge q^\star(u,v) \subseteq q^\star(u_1,v_1)\} \subseteq S_e$ holds. Hence, $q^2(u_0,v_0) \preceq S_e$. Conversely, for any $q^\star(u_0,v_0) \in S_e$, we have $\{q^2(u,v) : q^2(u,v) \in \mathcal{Q}_{dt}$ because of $q^\star(u_0,v_0)\} \subseteq S_{dt}$. Therefore, $q^\star(u_0,v_0) \preceq S_{dt}$. □

Example 13. Following Example 12, from $S = \mathcal{Q}_{dt} \setminus \{q^2((1,1),(1,2))\}$ we obtain S_e as $\{q^\star((1,3),(2,3)), q^\star((2,2),(2,3)), q^\star((2,3),(2,4))\}$ and S_{dt} as $\{q^2((1,3),(2,3)), q^2((2,2),(2,3)), q^2((2,3),(2,4))\}$. Notice that S_e includes all and only the queries defined on the new data set $C' = \{(1,3),(2,2),(2,3),(2,4)\}$.

Proposition 3 guarantees the equivalence at the cost of smaller subsets. In some situations we may be satisfied with weaker results, such as $S_{dt} \succeq S_e$, because then we know that if S_{dt} is safe then S_e must also be safe even though the converse is not necessarily true. The result in Proposition 4 is similar to Corollary 3 but gives only the sufficient condition. In Proposition 4, 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 $m = |\mathcal{Q}_e|$ and $n = |C|$.

Proposition 4. *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 $q^\star(u,v) \in S_e$, $|q^\star(u,v) \cap C_1| = |q^\star(u,v) \cap C_2| = |q^\star(u,v)|/2$ holds.*

Proof. Let $S \subseteq \mathcal{Q}_t$ satisfy that $G(C, S)$ is the complete bipartite graph with bipartition (C_1, C_2). Clearly $S_e \preceq S \equiv S_{dt}$. □

By Proposition 3 and Proposition 4, we can 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 $|S_e|$. This is equivalent to computing the *combinatorial discrepancy* of the set system formed by C and $\mathcal{Q}_e$ [4]. The alternative approach is to maximize $|S_{dt}|$, which is equivalent to finding the maximum bipartite subgraph of $G(C, \mathcal{Q}_{dt})$. Unfortunately, both solutions incur high computational complexity.

We can instead apply a simple procedure given in [20], as illustrated in Fig. 5. 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 $|\mathcal{Q}_{ds}| \geqslant |\mathcal{Q}_{dt}|/2$.

7. Implementation issues

We first show how the proposed techniques can be implemented based on a three-tiered inference control model [39], we then discuss various issues in such an implementation. The parity-based inference control method introduced in the current paper can be applied to OLAP systems based upon this three-tiered inference control model. The objective of the three-tiered inference control model is to minimize

Procedure *Bipartize_QDT*
Input: The data set C, $\mathcal{Q}_{dt}$
Output: the safe subset $\mathcal{Q}_{ds}$
Method:

1. **Let** $\mathcal{Q}_{ds} = \phi$, $S_{old} = \phi$;
2. **For** each $t_{new} \in C \setminus S_{old}$
   Let $C_{red} = \{t : t \in S_{old}$, t is red, and $q^2(t, t_{new}) \in \mathcal{Q}_{dt}\}$
   and $C_{blue} = \{t : t \in S_{old}$, t is blue, and $q^2(t, t_{new}) \in \mathcal{Q}_{dt}\}$;
   If $\mid C_{red} \mid > \mid C_{blue} \mid$
   Color t_{new} blue;
   For each $t_{old} \in C_{red}$
   Let $\mathcal{Q}_{ds} = \mathcal{Q}_{ds} \cup \{q^2(t_{old}, t_{new})\}$;
   Else
   Color t_{new} red;
   For each $t_{old} \in C_{blue}$
   Let $\mathcal{Q}_{ds} = \mathcal{Q}_{ds} \cup \{q^2(t_{old}, t_{new})\}$;
3. **Return** $\mathcal{Q}_{ds}$;

Fig. 5. A procedure for finding large safe subsets of $\mathcal{Q}_{dt}$.

the performance penalty of inference control methods. This is achieved through introducing a new tier, *aggregation tier* A, to the traditional two tiered view (i.e., *data tier* D and *query tier* Q) of inference control. The three tiers are related by three relations $R_{AD} \subseteq A \times D$, $R_{QA} \subseteq Q \times A$, and $R_{QD} = R_{AD} \circ R_{QA}$. The aggregation tier A satisfies three conditions. Firstly, $|A|$ is comparable to $|D|$. Second, there exists partition $\mathcal{P}$ on A such that the composition of R_{AD} and the equivalence relation given by $\mathcal{P}$ gives a partition on D. Finally, inferences are eliminated in the aggregation tier A.

The three-tiered model owes its advantages to the three conditions mentioned above. First, because $|A|$ is relatively small (in most cases $|Q| \gg |D|$ is true), controlling inferences caused by A is easier than controlling inferences caused by Q because of the smaller input to inference control methods. Second, because A and D can both be partitioned, inference control can be *localized* to the R_{AD}-related blocks of A and D, which further reduces the complexity. Moreover, the consequences of any undetected external knowledge are confined to blocks, making inference control more *robust*. Finally, as the most expensive task of three-tiered inference control, the construction of A can be processed off-line (i.e., before any query arrives). Decomposing queries into pre-computed aggregations is a built-in capability in most OLAP systems, and hence the online performance overhead of the three-tiered inference control is usually acceptable in these systems.

To apply the parity-based techniques, we first partition a given collection of data based on its inherent dimension hierarchies. Each block in the partition is re-

garded as a separate data set. The safe $\mathcal{Q}_{dt}$ (or its safe subsets S_{dt} if $\mathcal{Q}_{dt}$ 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 6 we have shown that $|\mathcal{Q}_{dt}| = O(n^2)$, where $n = |C|$, satisfying the first condition of the three-tiered model (that is, $|A|$ is comparable to $|C|$). Because $\mathcal{Q}_{dt}$ is separately defined on each data set, the aggregation tier has a natural partition corresponding to the partition of the data tier, satisfying the second condition (that is, A and D can be partitioned). The last condition (that is, A is free of inferences) is satisfied because we use the safe subsets of $\mathcal{Q}_{dt}$ when it is unsafe.

By integrating our results on the basis of the three-tiered model, we inherit all the advantages of the model, such as the capability of shifting computational overhead to off-line processing. At the same time, we also inherit limitations of the model, such as the impact on availability of queries. For example, even MDR queries that span more than one block in the partition may be rejected if any of their intersections with these blocks are not answerable. One important reason to apply only the parity-based method to blocks in a partition of the data set but not to the entire data set is as follows. By Lemma 1, every QDT graph is connected, and hence compromising one tuple can lead to the inference of all tuples in the data set. This situation should be avoided considering that there might be undetected external knowledge. Applying the parity-based method to each block in a partition of the data set can confine the damage caused by undetected external knowledge to a single block. However, as mentioned above, doing so may also render some queries spanning multiple blocks unanswerable. This reflects a fundamental tradeoff between the availability of queries and the security of data. We could always improve the security by making a more conservative assumption about external knowledge and having smaller blocks in the partition of the data set, but at the same time we sacrifice the availability, and vice versa.

As in statistical databases, the damage caused by undetected external knowledge can be alleviated by not allowing inferences of k ($k > 1$) tuples. For this purpose, one may suspect a *generalized* parity-based approach that prohibits any MDR query whose cardinality is not divisible by k, where k can be any number greater than one. By removing inferences of $k - 1$ or less values, such an approach could then tolerate undetected knowledge about no more than $k - 2$ values. It can be shown that the Procedure *QDT* in Fig. 3 can be easily extended to construct a set of k tuples (similar to the Q_{dt} in the case of $k = 2$). An extension to the proof of Theorem 2 will then show that this set of k tuples and the collection of all MDR queries whose cardinalities are divisible by k are mutually derivable. However, unlike the case of $k = 2$, a set of k ($k > 2$) tuples forms a hyper graph, and it remains an open problem to determine efficiently whether such a set causes inferences to $k - 1$ or less values (here, *efficient* refers to having a lower complexity than that of Audit Expert).

As discussed earlier, the proposed parity-based method has a complexity of $O(mn)$ for m queries over n tuples. The performance overhead implied by this complexity is certainly not negligible. However, the construction is done off-line before any query arrives at the system. The complexity $O(mn)$ thus has little impact on the online performance. This is a reasonable approach in OLAP systems since they typically rely on extensive off-line processing to reduce online delays in interacting with users. Partitioning the data set and applying the parity-based method to each block in the partition will also reduce the overall complexity (because the number of queries m is smaller in each block), although this is achieved at the price of reduced availability of queries. Data in OLAP systems are typically updated less frequently than they are in transactional databases, so it is usually acceptable to re-compute periodically the result of inference control on blocks of data that have been updated.

8. Conclusions

In this paper we have shown that directly applying existing inference control methods to MDR queries is usually inefficient because these methods ignore the inherent redundancy in MDR queries. We then proved the equivalence between the collection of all even MDR queries and a special collection of sum-two queries. On the basis of this equivalence, we showed how to determine the compromisability of even MDR queries with improved performance. We showed that odd MDR queries must be restricted but can be closely approximated by the even ones. We showed that safe arbitrary queries can be efficiently determined. We have also established the equivalence between subsets of even MDR queries and sum-two queries and given sufficient conditions for finding safe subsets of MDR queries.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable comments. This material is based upon work supported by the National Science Foundation under grants IIS-0242237 and IIS-0430402. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] N.R. Adam and J.C. Wortmann, Security-control methods for statistical databases: a comparative study, *ACM Computing Surveys* **21**(4) (1989), 515–556.

[2] R. Agrawal and R. Srikant, Privacy-preserving data mining, in: *Proceedings of the Nineteenth ACM SIGMOD Conference on Management of Data (SIGMOD'00)*, 2000, pp. 439–450.

[3] R. Agrawal, R. Srikant and D. Thomas, Privacy-preserving olap, in: *Proceedings of the Twenty-fourth ACM SIGMOD Conference on Management of Data (SIGMOD'05)*, 2005, pp. 251–262.

[4] J. Beck and V.T. Sós, Discrepancy theory, in: *Handbook of Combinatorics*, R.L. Graham, M. Grötschel and L. Lovász, eds, Elsevier Science, 1995, pp. 1405–1446.

[5] L.L. Beck, A security mechanism for statistical databases, *ACM Trans. on Database Systems* **5**(3) (1980), 316–338.

[6] L. Brankovic, M. Miller, P. Horak and G. Wrightson, Usability of compromise-free statistical databases, in: *Proceedings of the Ninth International Conference on Scientific and Statistical Database Management (SSDBM'97)*, 1997, pp. 144–154.

[7] A. Brodsky, C. Farkas, D. Wijesekera and X.S. Wang, Constraints, inference channels and secure databases, in: *Proceedings of the Sixth International Conference on Principles and Practice of Constraint Programming*, 2000, pp. 98–113.

[8] F.Y. Chin, Security in statistical databases for queries with small counts, *ACM Transaction on Database Systems* **3**(1) (1978), 92–104.

[9] F.Y. Chin, Security problems on inference control for sum, max, and min queries, *Journal of the Association for Computing Machinery* **33**(3) (1986), 451–464.

[10] F.Y. Chin, P. Kossowski and S.C. Loh, Efficient inference control for range sum queries, *Theoretical Computer Science* **32** (1984), 77–86.

[11] F.Y. Chin and G. Özsoyoglu, Security in partitioned dynamic statistical databases, in: *Proceedings of the Third IEEE International Computer Software and Applications Conference (COMPSAC'79)*, 1979, pp. 594–601.

[12] F.Y. Chin and G. Özsoyoglu, Auditing and inference control in statistical databases, *IEEE Trans. on Software Engineering* **8**(6) (1982), 574–582.

[13] V. Ciriani, S. De Capitani di Vimercati, S. Foresti and P. Samarati, K-anonymity, in: *Security in Decentralized Data Management*, T. Yu and S. Jajodia, eds, Springer, 2006.

[14] L.H. Cox, Suppression methodology and statistical disclosure control, *Journal of American Statistical Association* **75**(370) (1980), 377–385.

[15] D.E. Denning and P.J. Denning, Data security, *ACM Computing Surveys* **11**(3) (1979), 227–249.

[16] 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) (1979), 76–96.

[17] D.E. Denning and J. Schlörer, Inference controls for statistical databases, *IEEE Computer* **16**(7) (1983), 69–82.

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

[19] W. Du and Z. Zhan, Building decision tree classifier on private data, in: *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, 2002.

[20] P. Erdös, On some extremal problems in graph theory, *Israel Journal of Math.* **3** (1965), 113–116.

[21] L.P. Fellegi, On the question of statistical confidentiality, *Journal of American Statistic Association* **67**(337) (1972), 7–18.

[22] 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 Twelfth International Conference on Data Engineering*, 1996, pp. 152–159.

[23] D.T. Ho, R. Agrawal, N. Megiddo and R. Srikant, Range queries in olap data cubes, in: *Proceedings Sixteenth ACM SIGMOD International Conference on Management of Data (SIGMOD'97)*, 1997, pp. 73–88.

[24] J. Kleinberg, C. Papadimitriou and P. Raghavan, Auditing boolean attributes, in: *Proceedings of the Ninth ACM SIGMOD-SIG ACT-SIGART Symposium on Principles of Database System*, 2000, pp. 86–91.

[25] Y. Li, H. Lu and R.H. Deng, Practical inference control for data cubes (extended abstract), in: *Proceedings of the IEEE Symposium on Security and Privacy*, 2006.

[26] Y. Li, L. Wang and S. Jajodia, Preventing interval based inference by random data perturbation, in: *Proceedings of The Second Workshop on Privacy Enhancing Technologies (PET'02)*, 2002.

[27] Y. Li, L. Wang, X.S. Wang and S. Jajodia, Auditing interval-based inference, in: *Proceedings of the Fourteenth Conference on Advanced Information Systems Engineering (CAiSE'02)*, 2002, pp. 553–568.

[28] 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)*, 2002, pp. 148–159.

[29] F.M. Malvestuto and M. Mezzini, Auditing sum queries, in: *Proceedings of the Ninth International Conference on Database Theory (ICDT'03)*, 2003, pp. 126–146.

[30] 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*, 1998, pp. 89–99.

[31] G. Miklau and D. Suciu, A formal analysis of information disclosure in data exchange, in: *Proceedings of the 23th ACM SIGMOD Conference on Management of Data (SIGMOD'04)*, 2004.

[32] P. Samarati, Protecting respondents' identities in microdata release, *IEEE Transactions on Knowledge and Data Engineering* **13**(6) (2001), 1010–1027.

[33] P. Samarati and L. Sweeney, Generalizing data to provide anonymity when disclosing information (abstract), in: *PODS*, 1998.

[34] J. Schlörer, Security of statistical databases: multidimensional transformation, *ACM Trans. on Database Systems* **6**(1) (1981), 95–112.

[35] J.F. Traub, Y. Yemini and H. Woźniakowski, The statistical security of a statistical database, *ACM Trans. on Database Systems* **9**(4) (1984), 672–679.

[36] J. Vaidya and C. Clifton, Privacy preserving association rule mining in vertically partitioned data, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002, pp. 639–644.

[37] L. Wang, S. Jajodia and D. Wijesekera, Securing OLAP data cubes against privacy breaches, in: *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P'04)*, 2004, pp. 161–175.

[38] L. Wang, Y.J. Li, D. Wijesekera and S. Jajodia, Precisely answering multi-dimensional range queries without privacy breaches, in: *Proceedings of the Eighth European Symposium on Research in Computer Security (ESORICS'03)*, 2003, pp. 100–115.

[39] L. Wang, D. Wijesekera and S. Jajodia, Cardinality-based inference control in sum-only data cubes, in: *Proceedings of the Seventh European Symposium on Research in Computer Security (ESORICS'02)*, 2002, pp. 55–71.

[40] L. Wang, D. Wijesekera and S. Jajodia, Cardinality-based inference control in data cubes, *Journal of Computer Security* **12**(5) (2004), 655–692.

[41] C. Yao, X. Wang and S. Jajodia, Checking for k-anonymity violation by views, in: *Proceedings of the Thirty-first Conference on Very Large Data Base (VLDB'05)*, 2005.