2014

A Comprehensive Theoretical Framework for Privacy Preserving Distributed OLAP

Alfredo Cuzzocrea
University of Calabria, Cosenza, Italy, cuzzocrea@si.deis.unical.it

Elisa Bertino
Purdue University, bertino@cs.purdue.edu

Cuzzocrea, Alfredo and Bertino, Elisa, "A Comprehensive Theoretical Framework for Privacy Preserving Distributed OLAP" (2014).
Cyber Center Publications. Paper 634.
http://dx.doi.org/10.1007/978-3-662-45550-0_16

A Comprehensive Theoretical Framework for Privacy Preserving Distributed OLAP

Alfredo Cuzzocrea[1] and Elisa Bertino[2]

[1] ICAR-CNR and University of Calabria, Italy
[2] CERIAS and Purdue University, IN, USA
cuzzocrea@si.deis.unical.it, bertino@cs.purdue.edu

Abstract. This paper complements the *privacy preserving distributed OLAP framework* proposed by us in a previous work by introducing *four major theoretical properties* that extend models and algorithms presented in the previous work, where the experimental validation of the framework has also been reported. Particularly, our framework makes use of the *CUR matrix decomposition technique* as the elementary component for *computing privacy preserving two-dimensional OLAP views effectively and efficiently*. Here, we investigate theoretical properties of the CUR decomposition method, and identify four theoretical extensions of this method, which, according to our vision, may result in benefits for a wide spectrum of aspects in the context of privacy preserving distributed OLAP, such as *privacy preserving knowledge fruition schemes and query optimization*. In addition to this, we also provide a widespread experimental analysis of the framework, which fully confirms to us the major practical achievements, in terms of both efficacy and efficiency, due to our framework.

1 Introduction

Privacy Preserving Distributed OLAP [2,31,19,24,4,23] is crucial in today's world as organizations need to collaboratively harness the potential of "big data" analytics for decision making while at the same time assuring the privacy and confidentiality of their own data sets. This is referred in literature as *privacy preserving OLAP* [10,8] *in distributed environments* [2,31,19,24,4,23]. In scenarios adhering to the privacy preserving distributed OLAP paradigm, analysts working at different companies federated to the same main organization perform OLAP on business data extracted from their own respective legacy databases. To this end, analysts build and query data cubes made available in the other companies' sites, thus exposing them to *privacy breaches*. Assume that each company within the same main organization has the policy that the company's sale data is strictly confidential and not to be shared with other company. On the other hand, the organization encourages companies to collaboratively work on enhancing common *Business Intelligence* (BI). It should be noted that the respective OLAP tasks performed by analysts against their own data cube are private, as each of them uses only data belonging to her/his company. By

R. Meersman et al. (Eds.): OTM 2014 Workshops, LNCS 8842, pp. 117–136, 2014.

contrast, when analysts *aggregate* their respective data cubes into one common *SUM-based data cube* for BI purposes, data privacy breaches arise as analysts can easily *infer* data cells of other analysts' data cubes starting from data cells stored in the common data cube and her (his, respectively) proper data cube, via simple *linear interpolation techniques* [11,12].

Our reference scenario example adheres to the *Secure Multiparty Computation* (SMC) [28] model, which is well-known in the context of *Privacy Preserving Distributed Data Mining research* [5,21], with respect to which Privacy Preserving Distributed OLAP can be considered a major, yet-independent, research area. In order to solve the challenging problem of *computing and managing privacy preserving data cubes in distributed environments*, in [9] we introduced an innovative privacy preserving distributed OLAP framework that relies on the novel concept of *secure distributed OLAP aggregation task*.

Basically, this framework is based on the idea of performing OLAP across multiple distributed *SUM-based two-dimensional OLAP views* extracted from data cubes under the SMC requirements [9]. To this end, we introduced and experimentally assessed the S*ecure* D*istributed* O*LAP aggregation protocol* (SDO) [9]. Briefly, the SDO protocol works as follows. Having fixed a certain node ordering (e.g., DNS-based), the *first* node N_0 in the distributed environment computes a *privacy preserving version* V_0^{PP} of its proper OLAP view, denoted by V_0, and sends V_0^{PP} to the second node N_1 in the distributed environment. N_1 in turn: (*i*) combines V_0^{PP} with its proper local view, denoted by V_1, in order to perform the target OLAP operation; (*ii*) sends the local OLAP result, which is again represented by a view, denoted by V_1^{PP}, to the "following" node according to the fixed node ordering, and so forth. It should be noted that, since V_0^{PP} is privacy preserving, V_1^{PP} is also privacy preserving. This step is iterated until the local OLAP result computed at node N_{n-1} in the distributed environment, denoted by V_{n-1}^{PP}, is returned to the node N_0, which derives from V_{n-1}^{PP} the *exact* global OLAP result of the target OLAP task, denoted by V^{GLOBAL}, on the basis of the local view V_0, which is hidden to the other nodes of the distributed environment, and its privacy preserving version V_0^{PP}. The final global OLAP result V^{GLOBAL} is sent to the external application that finally forwards V^{GLOBAL} to all the other nodes in the distributed environment.

In this proposal, the core method exploited to obtain privacy-preserving two-dimensional OLAP views is given by the *CUR matrix decomposition* [14], which, as argued by Drineas *et al.* [13], can be used for privacy preservation purposes. In more detail, CUR is a matrix decomposition method for computing *approximate representations* of large matrices. It can be applied to several application contexts ranging from classification problems to similarity search problems, from analysis of biological data to compression of hyper-spectral data for image processing, and so forth [14].

Formally, given a large $m \times n$ matrix **A**, a CUR matrix decomposition is a *low-rank approximation* of **A**, denoted by **A'**, that represents **A** *in terms of a small number of columns and rows of* **A**, as follows: $\mathbf{A} \approx \mathbf{C} \cdot \mathbf{U} \cdot \mathbf{R} \equiv \mathbf{A'}$, where: (*i*) **C** is an $m \times c$

matrix *that* stores $O(1)$ columns of **A**; (*ii*) **R** is an $r \times n$ matrix that stores $O(1)$ rows of **A**; (*iii*) **U** is a $c \times r$ carefully-chosen matrix. In particular, the number of columns of **C** consists of $c = \theta(1/\varepsilon^2)$ columns of **A**, and the number of rows of **R** consists of $r = \theta(1/\varepsilon^2)$ rows of **A**, respectively, with $\varepsilon > 0$ arbitrarily small. **C** and **R** are built by means of *adaptive sampling* [30], via c (r, respectively) trials by picking a column (a row, respectively) of **A** with probability p_j defined as follows: $p_j = \frac{|\mathbf{A}[\bullet][j]|^2}{\sum_j |\mathbf{A}[\bullet][j]|^2}$, whereas the probability p_i for rows is defined as follows: $p_i = \frac{|\mathbf{A}[i]|^2}{\sum_i |\mathbf{A}[i]|^2}$, respectively.

In the previous work [9], we have specifically focused on the class of *SUM-based distributed OLAP aggregation tasks*, being SUM a popular aggregate operator for OLAP applications (e.g., [19]). Despite this, the framework [9] is general enough to deal with more sophisticated distributed OLAP aggregation tasks that embed complex OLAP aggregations (e.g., [18]) rather than conventional ones (e.g., SUM, COUNT, AVG). Also, while the framework [9] is general enough to deal with OLAP views computed over any arbitrary kind of data sources, in [9] we considered the specialized case represented by data cubes computed on top of *distributed collections of XML documents*, which are increasingly relevant for BI applications. The core of the framework [9] is represented by the *CUR matrix decomposition technique* [14], which allows us to compute privacy preserving two-dimensional OLAP views effectively and efficiently, *at a provable approximation error* [9].

In this paper, we further extend the proposed privacy preserving distributed OLAP framework [9] by providing a number of theoretical results that nicely extend the capabilities of the framework. We believe that these theoretical results result in benefits for a wide spectrum of aspects, such as *privacy preserving knowledge fruition schemes* and *query optimization*. More specifically, we provide theoretical contributions on the following aspects of the framework [9]: (*i*) *re-construction capabilities of the CUR decomposition method*, where we prove that the privacy preserving two-dimensional OLAP views can be used for re-constructing the original OLAP views in a theoretically-sound manner; (*ii*) *independence capabilities of the CUR decomposition method*, where we prove that the final two-dimensional OLAP view obtained from the target distributed OLAP aggregation task can be obtained from the two-dimensional OLAP views of the first and the last node of the reference environment, respectively, without dependency on the OLAP views of the remaining nodes, still in a theoretically-sound manner; (*iii*) *differential privacy notions*, where we introduce significant notions of *differential privacy* [15] for the framework [9] – as widely known, differential privacy represents a useful tool for representing and managing privacy preserving databases; (*iv*) *a widespread experimental analysis of the proposed privacy preserving distributed OLAP framework*, which fully confirms to us the major practical achievements, in terms of both efficacy and efficiency, due to this framework. A preliminary study that is the theoretical basis of this work appears in the workshop paper [42].

2 Related Work

Distributed Privacy Preserving OLAP techniques solve the problem of making distributed OLAP data cubes (i.e., OLAP data cubes populating a distributed environment) able to preserve the privacy of data during common (data) management tasks (e.g., computing data cubes, querying data cubes etc) or, under an alternative interpretation, generating a privacy preserving OLAP data cube from distributed data sources. With respect to the first problem, to the best of our knowledge, no approaches exist addressing this problem, beyond the framework [9], whereas concerning the second problem, the approach by Agrawal *et al.* [2] is the state-of-the-art approach. The approach [9] belongs to the second privacy preserving distributed OLAP scientific context, as it aims at solving the problem of supporting privacy preserving OLAP over distributed two-dimensional OLAP views, but, under a broader meaning, it also encompasses some characteristics of the first distributed privacy preserving OLAP scientific context.

By looking at the recent literature, while a plethora of initiatives focusing on Privacy Preserving Distributed Data Mining [5,21] exist, to the best of our knowledge, only [2,31,19,24,4,23] deal with the problem of effectively and efficiently supporting privacy preserving OLAP over distributed data sources. Agrawal *et al.* [2] define a privacy preserving OLAP model over data partitioned across multiple clients using a *randomization approach*, which is implemented by the so-called *Retention Replacement Perturbation* algorithm, on the basis of which (*i*) clients perturb tuples which they contribute to the partition in order to achive *row-level privacy*, and (*ii*) the server is capable of evaluating OLAP queries against perturbed tables via *reconstructing* the original distributions of attributes involved by such queries. Agrawal *et al.* prove that the proposed distributed privacy preserving OLAP model is safe against privacy breaches. The approach by Tong *et al.* [31] is another distributed privacy preserving OLAP approach that is reminiscent of ours. More specifically, they [31] propose the idea of obtaining a privacy preserving OLAP data cube from *distributed data sources across multiple sites* via applying perturbation-based techniques on *aggregate data* that are retrieved from each single site as a baseline step of the main (distributed) OLAP computation task. Other approaches [19,4,24] focus on the significant issue of providing efficient data aggregation while preserving privacy over *Wireless Sensor Networks* (WSN). In more details, He *et al.* [19] propose a solution based on two privacy-preserving data aggregation schemes that make use of innovative *additive aggregation functions*. These schemes are called *Cluster-based Private Data Aggregation* (CPDA) and *Slice-Mix-AggRegaTe* (SMART), respectively. The proposed aggregation functions fully-exploit topology and dynamics of the underlying wireless sensor network, and bridge the gap between collaborative data collection over such networks and data privacy needs. Chan and Castelluccia [4] focus on a formal treatment of a *Private Data Aggregation* (PDA) security model over WSN; this contribution is general enough to cover most cases of security-demanding scenarios over WSN and from a practical perspective allow one to execute privacy preserving data aggregation operations over WSN. The work by Lin *et al.* [24] is an incremental contribution aiming at improving security and saving energy consumption of the privacy preserving data aggregation task over WSN; the main idea of such an approach consists in integrating the super-increasing sequence and perturbation techniques into compressed data aggregations in order to gain efficiency. Li *et al.* [23] propose a novel incremental method for supporting secure and privacy preserving information aggregation over smart grids; this method introduces a novel scheme according to which data aggregation is performed at all smart meters involved in routing the data from the source meter to the collector unit, and the user privacy is provided by the use of *homomorphic encryption* on the transmitted data. Finally, there are more recent efforts that clearly confirm the interest from the research community for the issues investigated in this paper. Among others, Jurezyk and Xiong [22] propose a fully-decentralized anonymization protocol over horizontally-partitioned distributed databases, which supports privacy-preserving aggregate query answering in a distributed fashion, whereas Mohammed *et al.* [26] propose *LKC-privacy*, a new privacy model for achieving anonymization over distributed high-dimensional *healthcare data*, which is perfectly compliant with the privacy preserving distributed OLAP scenario we investigate in our research, as high-dimensional data smoothly resemble data cube cells.

3 From Privacy-Preserving Views to the Original Data Sets

A critical property that is central to theoretical aspects of the CUR decomposition method is related to assessing the capabilities of the method in re-constructing the original matrix **A** from the approximating matrix **A'** that is retrieved by the method itself. In fact, beyond playing a central role in the effectiveness of the proposed privacy preserving distributed OLAP framework, the re-construction property also ensures the theoretical convergence of conceptual constructs and theory tools of the framework.

In order to prove the re-construction property ensured by the CUR decomposition method, we provide Theorem 2 (see next) whose proof is characterized by a structure inspired by the theoretical model proposed by Agrawal *et al.* [2]. In more detail, with respect to the re-construction property ensured by the proposed *Retention Replacement Perturbation* algorithm [2], Agrawal *et al.* provide rigorous probabilistic bounds over aggregates that are re-constructed from a relational table that has been perturbed by means of their algorithm. These aggregates are defined in terms of input range queries over the perturbed relational table, and their values are compared with the values of aggregates retrieved by the same queries over the original relational table. Here, we follow a similar approach, i.e. we study the re-construction property of the CUR decomposition method by considering the aggregate values of range queries over the approximating matrix **A'** in comparison with the aggregate values of the same queries over the original matrix **A**.

Before introducing Theorem 2, some definitions are necessary. First, we define a two-dimensional range query Q over the $m \times n$ matrix **A** (**A'**, respectively) as follows:

$$Q = [\langle l_1 : u_1 \rangle; \langle l_2 : u_2 \rangle] \tag{1}$$

where: (*i*) l_1 denotes a lower bound on the dimension d_1 of **A** (**A'**, respectively); (*ii*) u_1 denotes an upper bound on the dimension d_1 of **A** (**A'**, respectively); (*iii*) $l_1 < u_1$; (*iv*) l_2 denotes a lower bound on the dimension d_2 of **A** (**A'**, respectively); (*v*) u_2 denotes an upper bound on the dimension d_2 of **A** (**A'**, respectively); (*vi*) $l_2 < u_2$. On the basis of well-understood matrix algebra [17] principles, the evaluation of Q over **A** (**A'**, respectively) can be expressed as follows:

$$\mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{y} = z \tag{2}$$

where: (*i*) **x** models an m-dimensional vector whose elements $\mathbf{x}[i]$, with $0 \leq i \leq m - 1$, are defined as follows:

$$\mathbf{x}[i] = \begin{cases} 1 & \textit{if } l_1 \leq i \leq u_1 \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

(*ii*) **y** models an n-dimensional vector whose elements $\mathbf{y}[j]$, with $0 \leq j \leq n - 1$, are defined as follows:

$$\mathbf{y}[j] = \begin{cases} 1 & \textit{if } l_2 \leq j \leq u_2 \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

and (*iii*) z models the answer to Q (z' models the approximate answer to Q, respectively).

For the sake of clarity, Theorem 2 proves that the approximate answer to Q, z', is *probabilistically-close* to the exact answer to Q, z, or, in other words, the re-construction property of the CUR decomposition method.

Second, we introduce the concept of *re-constructible function*, also inspired by [2], whose formal definition is provided in Definition 1. Intuitively, a numeric function γ is said to be re-constructible iff it allows us to "invert" the transformation of the original matrix **A** (cell partitions of **A**) in the perturbed matrix **A'** (cell partitions of **A'**) due to the CUR decomposition method, in our case. In our theoretical analysis, we interpret numeric functions γ as the data distributions associated with elements of the original matrix **A** (the approximating matrix **A'**, respectively). A relevant property of a re-constructible function γ is that of verifying whether it is *⟨n,ε,δ⟩-re-constructible* by means of the so-called *re-constructing function* γ', such that n is the number of items in γ, and ε and δ are positive integer arbitrarily small. In other words, this corresponds to verifying whether an *unbiased estimator* [27] γ' for γ exists. If this is the case, γ' gives us theoretically-proofed probabilistic bounds on the error we commit in reconstructing the function γ (by means of γ').

Definition 1. *Let* $\alpha : \mathbb{R}^m \rightarrow \mathbb{R}^n$ *be a perturbation function converting a matrix* **A** *into the approximating matrix* **A'**; *a numeric function* γ *on* **A** *is said to be* $\langle n,\varepsilon,\delta\rangle$*-re-constructible by means of a re-constructing function* γ'*, such that* n *is the number of items in* γ*, and* ε *and* δ *are positive integers arbitrarily small, iff* γ' *can be evaluated on* **A'** *and the following condition holds:* $|\gamma - \gamma'| = max\{\varepsilon, \varepsilon \cdot \gamma\}$*, such that* $max\{\mathcal{I}\}$ *denotes the operator max over a given item set* $\mathcal{I}$.

Based on these theoretical constructs and concepts, we now focus on re-constructing the answer z to a given range query $Q = [\langle l_1:u_1\rangle; \langle l_2:u_2\rangle]$ over **A** from the approximating matrix **A’** (or, equally, retrieving the approximate answer to Q, z') and the probabilities p_i and p_j (see Section 1) exploited by the CUR decomposition method to obtain **A’** from **A**. For this theoretical setting, the re-constructing function γ' we adopt is defined as follows:

$$\gamma'(Q=[\langle l_1:u_1\rangle;\langle l_2:u_2\rangle])=\sum_{i=l_1}^{u_1}\sum_{j=l_2}^{u_1}\left[\mathbf{A'}[i][j]-\frac{(1-p_i)\cdot p_j}{p_i\cdot(1-p_j)}\cdot b\right] \tag{5}$$

where: (*i*) **A’**[*i*][*j*] denotes an element of **A’**; (*ii*) p_i (see Section 1) denotes the probability of picking the *i*-th row of **A** during the CUR decomposition method; (*iii*) p_j (see Section 1) denotes the probability of picking the *j*-th column of **A** during the CUR decomposition method; (*iv*) b is defined as follows:

$$b=\frac{max\{\mathbf{A'}\}-min\{\mathbf{A'}\}}{max\{\mathbf{A}\}-min\{\mathbf{A}\}} \tag{6}$$

such that $max\{\mathbf{B}\}$ denotes the operator *max* over the elements of **B**, with **B** in {**A**, **A’**}, and $min\{\mathbf{B}\}$ denotes the operator *min* over the elements of **B**, with **B** in {**A**, **A’**}, respectively. Theorem 2 states that the re-constructing function γ' (5) is an unbiased estimator for the function γ determined by the CUR decomposition method, under the following condition:

$$n\geq 4\cdot\log(\frac{2}{\delta})\cdot(p_i\cdot p_j\cdot\varepsilon)^{-2} \tag{7}$$

where: (*i*) n denotes the number of elements of **A** involved in the evaluation of Q; (*ii*) ε and δ are positive integers arbitrarily small; (*iii*) p_i and p_j are the probabilities, exploited by the CUR decomposition method, respectively (see Section 1).

Theorem 2. *Let the value* **A**[*i*][*j*] *in* [*min*{**A'**}, *max*{**A'**}] *be estimated by the re-constructing function γ’; then γ’ is a* ⟨*n*,*ε*,*δ*⟩*-unbiased-estimator for γ if the following condition holds:* $n\geq 4\cdot\log(\frac{2}{\delta})\cdot(p_i\cdot p_j\cdot\varepsilon)^{-2}\cdot$

Proof. Let $\mathcal{X}_{ij}$ denote a *random variable* [27] for the event that element **A**[*i*][*j*] of **A** is perturbed, and the perturbed element **A’**[*i*][*j*] is contained by the interval [*min*{**A'**}, *max*{**A'**}]. It should be noted that the collection of random variables $\mathcal{X}_{ij}$ are i.i.d. [27], and that the probability that element **A**[*i*][*j*] of **A** is perturbed is given by the following expression:

$$\mathrm{P}(\mathcal{X}_{ij}=1)=(1-p_i)\cdot(1-p_j)\cdot b \tag{8}$$

As a consequence, the following equality holds:

$$\mathrm{P}(\mathcal{X}_{ij}=0)=1-\mathrm{P}(\mathcal{X}_{ij}=1)=1-(1-p_i)\cdot(1-p_j)\cdot b \tag{9}$$

Likewise, let $\mathcal{Y}_{ij}$ denote a random variable for the event that element **A**[*i*][*j*] of **A** is *not* perturbed, and it is contained by the interval [*min*{**A'**}, *max*{**A'**}]. Similarly to the case of random variables $\mathcal{Y}_{ij}$, it should be clear that the collection of random variables

$\mathcal{Y}_{ij}$ are i.i.d. and that the probability that element **A**[i][j] of **A** is not perturbed is given by the following expression:

$$\mathrm{P}(\mathcal{Y}_{ij} = 1) = p_i \cdot p_j \tag{10}$$

In turn, the following expression holds:

$$\mathrm{P}(\mathcal{Y}_{ij} = 0) = 1 - \mathrm{P}(\mathcal{Y}_{ij} = 1) = 1 - p_i \cdot p_j \tag{11}$$

Now, let $\mathcal{Z}_{ij}$ denote a random variable for the event that, during the CUR decomposition method, element **A**[i][j] of **A** falls within the interval [min{**A'**}, max{**A'**}]. It follows that $\mathcal{Z}_{ij}$ can be defined in terms of the previous random variable $\mathcal{X}_{ij}$ and $\mathcal{Y}_{ij}$, as follows:

$$\mathcal{Z}_{ij} = \mathcal{X}_{ij} + \mathcal{Y}_{ij} \tag{12}$$

due to the fact that, during the CUR decomposition method, an arbitrary element **A**[i][j] of **A** may be contained (i.e., $\mathcal{X}_{ij}$ = 1 and $\mathcal{Y}_{ij}$ = 0) or not (i.e., $\mathcal{X}_{ij}$ = 0 and $\mathcal{Y}_{ij}$ = 1) by the interval [min{**A'**}, max{**A'**}]. From (12), it follows that the collection of random variables $\mathcal{Z}_{ij}$ are i.i.d. and that the probability that element **A**[i][j] of **A** falls within the interval [min{**A'**}, max{**A'**}] is given by the following expression:

$$\mathrm{P}(\mathcal{Z}_{ij} = 1) = \mathrm{P}((\mathcal{X}_{ij} + \mathcal{Y}_{ij}) = 1) = \mathrm{P}(\mathcal{X}_{ij} = 1) + \mathrm{P}(\mathcal{Y}_{ij} = 1) \tag{13}$$

From (8) and (10), (13) we finally obtain the following expression:

$$\mathrm{P}(\mathcal{Z}_{ij} = 1) = (1 - p_i) \cdot (1 - p_j) \cdot b + p_i \cdot p_j \tag{14}$$

As a consequence, the following formula holds:

$$\mathrm{P}(\mathcal{Z}_{ij} = 0) = 1 - \mathrm{P}(\mathcal{Z}_{ij} = 1) = 1 - (1 - p_i) \cdot (1 - p_j) \cdot b + p_i \cdot p_j \tag{15}$$

Furthermore, let Δ_1 denote the range of Q on the dimension d_1 of **A** (**A'**, respectively). From (1), it clearly follows that the cardinality of Δ_1, $||\Delta_1||$, is given by the following expression:

$$|\Delta_1|| = u_1 - l_1 \tag{16}$$

Similarly, let Δ_2 denote the range of Q on the dimension d_2 of **A** (**A'**, respectively). From (1), it clearly follows again that the cardinality of Δ_2, $||\Delta_2||$, is given by the following expression:

$$||\Delta_2|| = u_2 - l_2 \tag{17}$$

Also, let $||Q||$ denote the *volume* (or *selectivity* [6,7]) of Q. Based on (1), (16) and (17), $||Q||$ is given by the following expression:

$$||Q|| = ||\Delta_1|| \cdot ||\Delta_2|| \tag{18}$$

such that $||\Delta_1||$ denotes the cardinality of Δ_1, and $||\Delta_2||$ denotes the cardinality of Δ_2, respectively.

Now, let $\mathcal{U}_{ij}$ denote a random variable defined as the *summation of random variables* $\mathcal{Z}_{ij}$ [P] over the two-dimensional domain of **A** (**A'**, respectively) modeling the range of Q, i.e. [$\langle l_1{:}u_1\rangle$; $\langle l_2{:}u_2\rangle$] that is defined as follows:

$$\mathcal{U}_{ij}(\Delta_1,\Delta_2)=\sum_{h=i}^{i+\Delta_1-1}\sum_{k=j}^{j+\Delta_2-1}\mathcal{Z}_{hk}\cdot\mathbf{A'}[h][k] \tag{19}$$

It should be noted that random variables $\mathcal{U}_{ij}$ are those associated with the evaluation of the approximate answer to Q, z', and they underlie the definition of the re-constructing function γ' (5). The number of elements of **A'** involved in the Q's evaluation process, n (or, equally, the number of items of $\gamma' - \gamma$, respectively), is given by the following expression:

$$n = \|Q\| = \|\Delta_1\| \cdot \|\Delta_2\| \tag{20}$$

How to model the approximate evaluation of Q over **A'** *in a probabilistic manner?* In order to answer this critical question, first note that each one among the n elements **A'**[i][j] of **A'** may contribute (i.e., $\mathcal{U}_{ij}$ = 1) or not (i.e., $\mathcal{U}_{ij}$ = 0) to the approximate answer to Q, z'. Our final aim is to find probabilistic bounds for the probability P($\mathcal{U}_{ij}$ = 1). Since the random variables $\mathcal{Z}_{ij}$ are i.i.d. and the random variables $\mathcal{U}_{ij}$ are defined as the summation of $\mathcal{Z}_{ij}$, *then* $\mathcal{U}_{ij}$ *are independent Bernoulli random variables* [27]. Under the condition (7), by applying the well-known *Chernoff bound* [27], the following inequality holds:

$$P\left[\left|\mathcal{U}_{ij}(\Delta_1,\Delta_2)-n\cdot t\cdot\mathsf{AVG}(\Delta_1,\Delta_2)\right|>n\cdot\theta\right]<2e^{\frac{-n\cdot\theta^2}{4\cdot t}}\le\delta \tag{21}$$

such that (i) t = P($\mathcal{Z}_{ij}$ = 1) (14); (ii) AVG(Δ_1,Δ_2) denotes the average value of elements **A'**[i][j] of **A'** contained by the two-dimensional range of Q, [$\langle l_1$:$u_1\rangle$; $\langle l_2$:$u_2\rangle$]; (iii) θ is defined as follows:

$$\theta=\prod_{i=l_1}^{u_1}\prod_{j=l_2}^{u_2}p_i\cdot p_j\cdot\varepsilon \tag{22}$$

where p_i and p_j are the probabilities exploited by the CUR decomposition method (see Section 1), respectively, and ε is a positive integer arbitrarily small; (iv) δ is a positive integer arbitrarily small. From (21), it follows that, with probability greater than 1 – δ, the following inequality holds:

$$\gamma-\varepsilon<\sum_{i=l_1}^{u_1}\sum_{j=l_2}^{u_1}\left[\mathbf{A'}[i][j]-\frac{(1-p_i)\cdot p_j}{p_i\cdot(1-p_j)}\cdot b\right]<\gamma+\varepsilon \tag{23}$$

from which it follows that |γ – γ'| < ε with probability 1 – δ, and that re-constructing function γ' (5) is an unbiased estimator for the function γ determined by the CUR decomposition method.

4 Breaking the Dependency from Local Views: Towards Theoretically-Sound Privacy-Preserving Distributed OLAP

Based on the results in [9] and the fact that SUM-based OLAP aggregation is a *non-holistic* operator [16], it is easy to demonstrate that the final global result of the target distributed OLAP aggregation task, i.e. the view V^{GLOBAL}, can be reconstructed as follows (as formally stated by Theorem 3):

$$V^{GLOBAL} = V_0 + \left(V_{n-1}^{PP} - V_0^{PP}\right) \tag{24}$$

Theorem 3. *The final global* OLAP *view* V^{GLOBAL} *obtained from any arbitrary* SUM*-based secure* OLAP *aggregation task over a distributed environment populated by n nodes can be retrieved from combining the local* OLAP *view* V_0 *at node* N_0*, the privacy preserving* OLAP *view* V_0^{PP} *at node* N_0 *and the privacy preserving* OLAP *view* V_{n-1}^{PP} *at node* N_{n-1} *without dependency on the* OLAP *views located at other nodes* N_i*, with* $1 \leq i \leq n - 2$*, of the reference distributed environment, i.e.* $V^{GLOBAL} = V_0 + \left(V_{n-1}^{PP} - V_0^{PP}\right)$.

Proof. Take as reference a distributed environment populated by n nodes. First, note that, given two consecutive nodes N_{i-1} and N_i in the fixed node ordering, such that $1 \leq i \leq n - 2$, since we focus on SUM-based OLAP aggregations, the privacy preserving view V_i^{PP} at node N_i is obtained by combining the local view V_i at node N_i with the privacy preserving view V_{i-1}^{PP} returned to node N_i from node N_{i-1}, as follows (see Section 1):

$$V_i^{PP} = V_i + V_{i-1}^{PP} \tag{25}$$

By contrast, for the *sole* instance represented by the first node N_0, the privacy preserving view V_0^{PP} is directly obtained from the local view V_0 via the CUR-based approximation method (see Section 1). Hence, with respect to privacy preserving views located at nodes of the reference distributed environment, the following equalities hold:

$$\begin{aligned} V_0^{PP} &= \mathsf{CUR}(V_0) \\ V_1^{PP} &= V_1 + V_0^{PP} \\ V_2^{PP} &= V_2 + V_1^{PP} \\ &\ldots \\ V_{n-1}^{PP} &= V_{n-1} + V_{n-2}^{PP} \end{aligned} \tag{26}$$

Based on (25), by applying simple mathematical substitutions, (26) can be re-written as follows:

$$\begin{aligned} V_0^{PP} &= \mathsf{CUR}(V_0) \\ V_1^{PP} &= V_1 + V_0^{PP} \\ V_2^{PP} &= V_2 + V_1^{PP} = V_2 + V_1 + V_0^{PP} \\ &\ldots \\ V_{n-1}^{PP} &= V_{n-1} + V_{n-2} + \ldots + V_1 + V_0^{PP} \end{aligned} \tag{27}$$

Based on (27), (24) can be expanded as follows:

$$\begin{aligned} V^{GLOBAL} &= V_0 + \left(V_{n-1}^{PP} - V_0^{PP}\right) \\ &= V_0 + \left(\left(V_{n-1} + V_{n-2} + \ldots + V_1 + V_0^{PP}\right) - V_0^{PP}\right) \end{aligned} \tag{28}$$

i.e.:

$$V^{GLOBAL} = V_0 + V_1 + V_2 + \ldots + V_{n-1} \tag{29}$$

which, from Section 1, represents the (exact) final result of the target distributed OLAP aggregation task.

Theorem 3 is another relevant theoretical result of our research. It allows us to obtain the final global result of the target secure distributed OLAP aggregation task, V^{GLOBAL} from the OLAP views stored at the first node N_0 of the reference distributed environment, V_0 and V_0^{PP}, respectively, one exact (i.e., V_0) and one privacy preserving (i.e., V_0^{PP}), and from the privacy preserving OLAP view V_{n-1}^{PP} returned to node N_0 from node N_{n-1}, *without dependency on the OLAP views* (local and privacy preserving) *of other nodes* N_i, with $1 \le i \le n - 2$, of the reference distributed environment. Intuitively, this phenomenon opens interesting theoretical as well as query-optimization opportunities for enhancing our privacy preserving distributed OLAP framework.

As a useful corollary deriving from Theorem 3 (Corollary 1), it follows that our proposed framework is *orthogonal* to the specific method used for obtaining the privacy preserving view V_i^{PP} at node N_i (CUR, in our case). Hence our framework maintains its validity and generality with any arbitrary privacy preserving method (e.g., [11,20,29,32,33,34,12,25,3]). This gives further merits to our research.

Corollary 1. *The proposed privacy preserving distributed OLAP framework is orthogonal to the method used to compute privacy preserving two-dimensional OLAP views.*
Proof. A direct consequence from Theorem 3.

5 Innovative Differential Privacy Notions for Privacy Preserving Distributed OLAP

We now focus on other theoretical properties of our privacy preserving distributed OLAP framework that are related to the well-understood notion of differential privacy [15]. Based on the non-holistic nature of SUM-based OLAP aggregations [16], given a local view V_i at node N_i, we introduce two different differential privacy notions for the privacy preserving process over V_i (i.e., $V_i^{PP} = V_i + V_{i-1}^{PP}$). The first one, formally introduced by Definition 2, is referred to as *full-differential privacy* and denoted by ΔP_i^F. It is modeled as a (two-dimensional) view that makes V_i^{PP} different from V_i (or, equally, that makes V_i^{PP} privacy preserving with respect to V_i).

Definition 2. *Given a two-dimensional OLAP view V and its privacy preserving version V^{PP}, the full-differential privacy between V^{PP} and V, denoted by ΔP^F, is a two-dimensional view defined as follows:* $\Delta P^F = V^{PP} - V$.

The second differential privacy notion, formally introduced by Definition 3, is referred to as *marginal-differential privacy* and denoted by ΔP_i^M.

Definition 3. *Given a two-dimensional OLAP view V and its privacy preserving version V^{PP}, the marginal-differential privacy between V^{PP} and V, denoted by ΔP^M, is a partition of two-dimensional cells whose elements are defined as follows:*

$$\Delta P_i^M[m][c] = \begin{cases} V_i[m][c] & \text{if } V_i^{PP}[m][c] = V_i[m][c],\, 0 \leq m \leq \| d_{i,0} \| \wedge 0 \leq c \leq \| d_{i,1} \| \\ V_i^{PP}[m][c] & \text{otherwise} \end{cases} \textit{where } d_{i,0} \textit{ and } d_{i,1}$$

denote the dimensions of V.

Let $d_{i,0}^{PP}$ and $d_{i,1}^{PP}$ denote the dimensions of V_i^{PP}. It follows that $\| d_{i,0} \| = \| d_{i,0}^{PP} \|$ and $\| d_{i,1} \| = \| d_{i,1}^{PP} \|$. It should be noted that, while full-differential privacy ΔP_i^F defined according to Definition 2 conforms to the classical notion of differential privacy [15], the marginal-differential privacy ΔP_i^M defined according to Definition 3 plays a relevant role with respect to the critical problem of maintaining privacy under the occurrence of *updates* on the target data sources which affects Privacy Preserving Data Mining [1].

Starting from the basic differential privacy notions, their *global versions* are derived in our framework as follows. First, from Section 1, the following two critical aspects of our privacy preserving distributed OLAP framework should be clear enough: (*i*) the full-differential privacy ΔP_i^F that propagates across pairs of consecutive nodes N_{i-1} and N_i (in the fixed node ordering) of the reference distributed environment is originated by the "first" full-differential privacy ΔP_0^F at node N_0; (*ii*) ΔP_0^F represents the differential privacy that, by propagating node by node, makes each local view V_i at node N_i, with $1 \leq i \leq n - 1$, privacy preserving (i.e., V_i^{PP}), in consequence of the secure OLAP aggregation routine performed at node N_i as a baseline step of the whole secure distributed OLAP aggregation task. Hence, at the last node N_{n-1} of the reference distributed environment, *the full-differential privacy ΔP_{n-1}^F at node N_{n-1} globally embeds all the contributions of the "previous" full-differential privacies ΔP_0^F, ΔP_1^F, …, ΔP_{n-2}^F*. This leads to the concept of *global full-differential privacy*, formally introduced by Definition 4, that is associated with the target secure distributed OLAP aggregation task $\mathcal{T}$, denoted by $\Delta P_{\mathcal{T}}^F$.

Definition 4. *Given an arbitrary SUM-based secure OLAP aggregation task $\mathcal{T}$ over a distributed environment populated by n nodes, the global full-differential privacy associated to $\mathcal{T}$, denoted by $\Delta P_{\mathcal{T}}^F$, is defined as follows:* $\Delta P_{\mathcal{T}}^F = \sum_{i=0}^{n-1} \Delta P_i^F = \Delta P_0^F + \Delta P_1^F + \ldots + \Delta P_{n-1}^F$, *where ΔP_i^F denotes the full-differential privacy at node N_i.*

Intuitively like Theorem 3, Definition 4 can be exploited for further theoretical analysis as well as for query-optimization opportunities. At the same time, we derive, with similar insights, the concept of *global marginal-differential privacy*, formally introduced by Definition 5, that is associated with the target secure distributed OLAP aggregation task $\mathcal{T}$, denoted by $\Delta P_{\mathcal{T}}^{M}$.

Definition 5. *Given an arbitrary SUM-based secure OLAP aggregation task* $\mathcal{T}$ *over a distributed environment populated by n nodes, the global marginal-differential privacy associated to* $\mathcal{T}$, *denoted by* $\Delta P_{\mathcal{T}}^{M}$, *is defined as follows:* $\Delta P_{\mathcal{T}}^{M} = \sum_{i=0}^{n-1} \Delta P_{i}^{M} = \Delta P_{0}^{M} + \Delta P_{1}^{M} + \ldots + \Delta P_{n-1}^{M}$, *such that* ΔP_{i}^{M} *denotes the marginal-differential privacy at node* N_i.

Like the other differential privacy notions introduced in this paper, both the global full- and marginal-differential privacy can be used as baseline operators to devise more sophisticated privacy-preserving protocols with SMC features (like SDO [9]) over distributed OLAP settings more complex than the one assumed in this paper. In these settings, participant nodes may belong to *different classes* (e.g., classes of nodes with different behavior), or may hold *disjoint partial knowledge* on the overall environment. By combine such "pieces" of knowledge nodes can enhance their ability to infer additional knowledge which leads to increased data privacy risks.

6 Experimental Evaluation and Analysis

In order to assess the effectiveness and the efficiency of our privacy preserving distributed OLAP framework [9], we conducted a comprehensive experimental campaign on distributed collections of synthetic, benchmark and real-life XML documents stored in (synthetic, benchmark and real-life) XML data sets, against which we tested the performance of the proposed secure distributed OLAP aggregation protocol SDO [9] under the ranging of several experimental parameters. This finally allowed us to achieve a wide and reliable experimental evaluation and analysis. We selected XML data sets as the reference ones for our experimental assessment as XML is widely known as the *de facto* standard for OLAP over distributed environments.

First, we provide a description about the XML data sets adopted in our experimental campaign. For what regards synthetic XML data sets, element values of the synthetic XML documents have been generated according to three distinct data distributions, namely *Uniform* [6], *Gauss* [27], and *Zipf* [35]. In more detail, Uniform data sets have been generated by means of a Uniform distribution on the interval [75, 125]; Gauss data sets have been generated by means of a *normal* Gauss distribution; Zipf data sets have been generated by means of a Zipf distribution whose parameter z ranges on the interval [0.5, 0.9]. As regards benchmark XML data sets, we considered

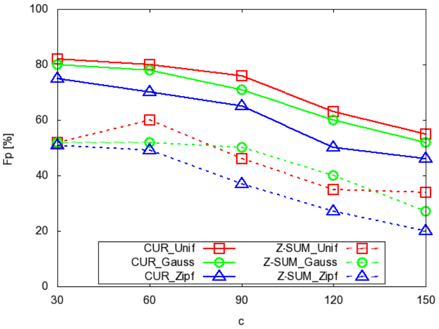

Fig. 1. Variation of F_P w.r.t. the number of columns c (number of blocks, respectively) on 2,000 × 2,000 OLAP views extracted from synthetic XML data sets for the CUR decomposition method and *Zero-Sum*

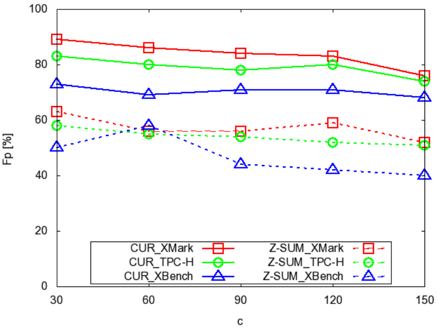

Fig. 2. Variation of F_P w.r.t. the number of columns c (number of blocks, respectively) on 2,000 × 2,000 OLAP views extracted from benchmark XML data sets for the CUR decomposition method and *Zero-Sum*

the popular data sets *XMark* [36], the *XML-version of TPC-H* [37] and *XBench* [38]. In particular, for the data set *TCP-H*, we considered the XML document extracted from the table *Lineitem*, which is the biggest one among those composing the whole data set. Finally, as regards real-life XML data sets, we considered the well-known data sets *Treebank* [39], *SwissProt* [40] and *NASA* [41]. In turn, from each one of these XML data sets, we extracted a 2,000 × 2,000 two-dimensional OLAP view, whose (two-dimensional) data cells follow the same distribution of the underlying (XML) data set.

We defined four kinds of experiment. In the first kind of experiment, we analyze the privacy preserving capabilities of the CUR decomposition method on a *singleton* two-dimensional OLAP view with respect to the ranging of the number of columns c exploited by the method itself to compute privacy preserving OLAP views (see Section 1). In the second kind of experiment, we conducted a similar experience but focused to study the CUR-decomposition's privacy preserving capabilities with respect to the ranging of the probability p_j (see Section 1) of picking the j-th column of the target OLAP view during the decomposition process (see Section 1). In the third kind of experiment, we analyze the privacy preserving capabilities of our distributed OLAP framework, by studying how the "privacy degree" of distributed OLAP views, stored at nodes of the target experimental setting, varies across the nodes under the execution of a SUM-based distributed OLAP aggregation task (see Section 1). In all the first three kinds of experiment, we considered *Zero-Sum* [29] as the comparison method. This because of three main reasons: (*i*) *Zero-Sum* makes use of a *matrix-like* formalism to face-off and solve the privacy preserving of OLAP data cubes, like ours; (*ii*) *Zero-Sum* can be reasonably considered as one of the state-of-the-art perturbation-based approach for centralized privacy preserving OLAP; (*iii*) due to its simplicity, *Zero-Sum* can be easily extended as to deal with the more probing case of distributed privacy preserving OLAP, like the one addressed and solved by our proposed framework (in this extended implementation, the "original" method *Zero-Sum* plays the same role of the one played by the CUR decomposition method in our framework, i.e. dealing with the privacy preservation of a singleton two-dimensional OLAP view). Finally, in the fourth kind of experiment, we stressed the *sensitivity* of the CUR decomposition method by studying the variation of the probability $P(e^E)$ of the event of $e^E = \mathbf{A} \equiv \mathbf{A'}$, which models the case of obtaining the approximate matrix **A'** as *equal* to the input matrix **A** (see Section 1), with respect to the ranging of the probability p_j (see Section 1), like in the second kind of experiment.

As regards the metrics of evaluating our privacy preserving distributed OLAP framework, we considered the *privacy factor* F_P introduced by Sung *et al.* in [29], which gives a reliable measure of how much a privacy preserving OLAP data cube (OLAP view, respectively) $\mathcal{D}'$ preserves the privacy of the original OLAP data cube (OLAP view, respectively) $\mathcal{D}$ by inspecting the privacy of cells of $\mathcal{D}'$ with respect to cells of $\mathcal{D}$. In more detail, let (*i*) $\mathcal{D}$ be the input data cube, (*ii*) $\mathcal{D}'$ be the privacy preserving data cube computed by means of a given (privacy preserving) method, (*iii*) $X\{\mathbf{k}\}$ be a data cube cell having **k** as multidimensional index, with X in $\{\mathcal{D}, \mathcal{D}'\}$, the privacy factor F_P is defined as follows [29]:

$$F_P(\mathcal{D},\mathcal{D}') = \frac{1}{\|\mathcal{D}\|}\cdot\sum_{\mathbf{k}=0}^{\|\mathcal{D}\|-1}\frac{|\mathcal{D}'\{\mathbf{k}\}-\mathcal{D}\{\mathbf{k}\}|}{|\mathcal{D}\{\mathbf{k}\}|} \tag{30}$$

Figure 1 shows the results obtained from the first kind of experiment, i.e. the percentage variation of the privacy factor F_P with respect to the number of columns c on 2,000 × 2,000 two-dimensional OLAP views extracted from the synthetic XML data sets. Similarly to this, Figure 2 and Figure 3 show the same experimental pattern on benchmark and real-life XML data sets, respectively. With respect to the comparison approach *Zero-Sum*, the parameter c models the *number of blocks* of the partition used to compute the final privacy preserving OLAP view [29]. As shown by Figure 1, Figure 2 and Figure 3, privacy factor values ensured by the CUR decomposition method are high so that obtained (perturbed) OLAP views are privacy preserving accordingly. Also, it turns that the CUR decomposition method outperforms *Zero-Sum* . Note that, with respect to the synthetic XML data sets, the CUR decomposition method works well on Uniform data sets rather than on Gauss and Zipf data sets (for which the performance is still high), because of the sampling phase introduced by the method (as widely-known, sampling works well on Uniform data sets [7] rather than other kinds of data sets).

Figure 4 shows the results obtained from the second kind of experiment, i.e. the percentage variation of the privacy factor F_P with respect to the probability p_j (see Section 1) on the target OLAP views over synthetic XML data sets considered in our experimental assessment. Similarly to this, Figure 5 and Figure 6 show the same experimental pattern on benchmark and real-life XML data sets, respectively. With respect to the comparison approach *Zero-Sum*, the parameter p_j models the *probability of perturbing data cube cells belonging to the j-th block* of the partition used to

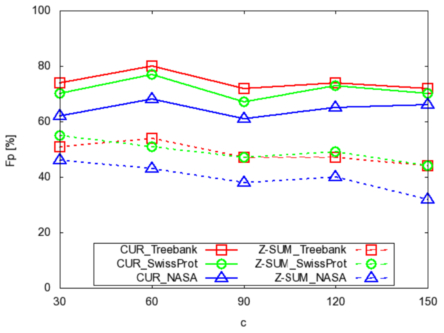

Fig. 3. Variation of F_P w.r.t. the number of columns c (number of blocks, respectively) on 2,000 × 2,000 OLAP views extracted from real-life XML data sets for the CUR decomposition method and *Zero-Sum*

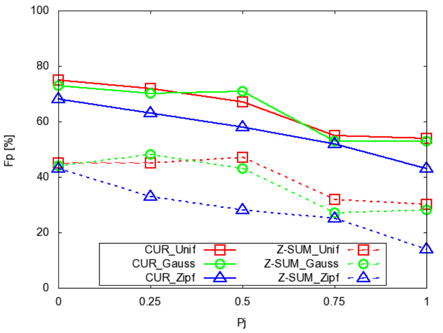

Fig. 4. Variation of F_P w.r.t. the number of columns c (number of blocks, respectively) on 2,000 × 2,000 OLAP views extracted from real-life XML data sets for the CUR decomposition method and *Zero-Sum*

compute the final privacy preserving OLAP view [29]. Like for the case of the first kind of experiment, as confirmed by Figure 4, Figure 5 and Figure 6, we again observed a good performance of the CUR decomposition method.

Figure 7 shows the results obtained from the third kind of experiment, i.e. the percentage variation of the privacy factor F_P with respect to the position i of nodes populating a target experimental distributed environment composed by 20 nodes, such that each node stores one singleton OLAP view over synthetic XML data sets among those considered in our experimental assessment. Similarly to this, Figure 8 and Figure 9 show the same experimental pattern on benchmark and real-life XML data sets, respectively. In more detail, in this experiment we inspected the "privacy degree" of each new local OLAP view generated in each node by a singleton aggregation step of the whole distributed OLAP aggregation task. As shown by Figure 7, Figure 8 and Figure 9, the privacy factor F_P increases with the node position, i.e. each new local OLAP view achieves a *higher* privacy degree than the degree of previous views (in the fixed node ordering). The latter is a nice property confirming that our proposed privacy preserving distributed OLAP framework fully satisfies the rigorous requirements and constraints posed by the SMC model. Also, Figure 7, Figure 8 and Figure 9 demonstrate that the proposed privacy framework, beyond ensuring a good privacy preservation effect on singleton two-dimensional OLAP views as confirmed by the previous two kinds of experiment, exposes a good performance even over the target experimental distributed environment, hence it perfectly fulfills the initial goals (i.e., effectively and efficiently supporting secure distributed OLAP aggregation tasks – see Section 1), yet outperforming the comparison approach *Zero-Sum* in a distributed setting as well as in a centralized one.

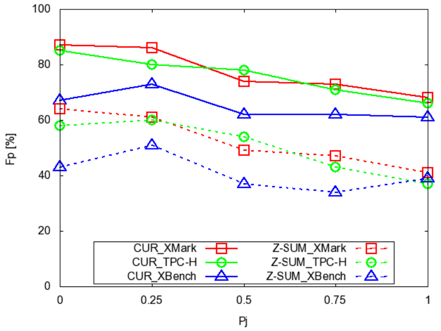

Fig. 5. Variation of F_P w.r.t. the probability p_j on 2,000 × 2,000 OLAP views extracted from benchmark XML data sets for the CUR decomposition method and *Zero-Sum*

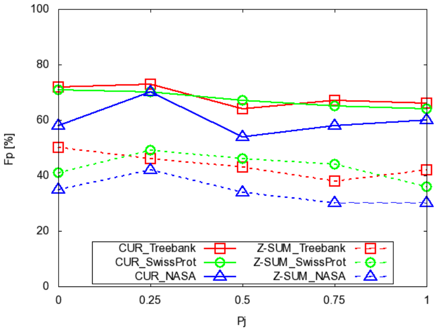

Fig. 6. Variation of F_P w.r.t. the probability p_j on 2,000 × 2,000 OLAP views extracted from real-life XML data sets for the CUR decomposition method and *Zero-Sum*

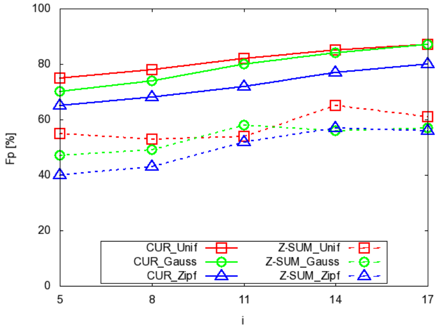

Fig. 7. Variation of F_P w.r.t. the node position i of a 20-node distributed environment on 2,000 × 2,000 OLAP views extracted from synthetic XML data sets for the CUR decomposition method and *Zero-Sum*

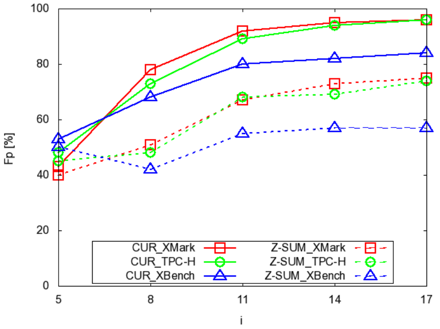

Fig. 8. Variation of F_P w.r.t. the node position i of a 20-node distributed environment on 2,000 × 2,000 OLAP views extracted from benchmark XML data sets for the CUR decomposition method and *Zero-Sum*

Finally, Figure 10 shows the results obtained from the fourth kind of experiment, i.e. the variation of the probability $P(e^E)$ with respect to the ranging of the probability p_j (see Section 1), both being critical model parameters of the CUR decomposition method, again on the target OLAP views over synthetic XML data sets considered in our experimental assessment. Similarly to this, Figure 11 and Figure 12 show the same experimental pattern on benchmark and real-life XML data sets, respectively. As shown in Figure 10, Figure 11 and Figure 12, we observe an initial increase of $P(e^E)$ as p_j increases (as expected), but then $P(e^E)$ makes stable to around under the value 0.5. This further confirms to us the benefits of the CUR decomposition method in computing effective privacy preserving OLAP views.

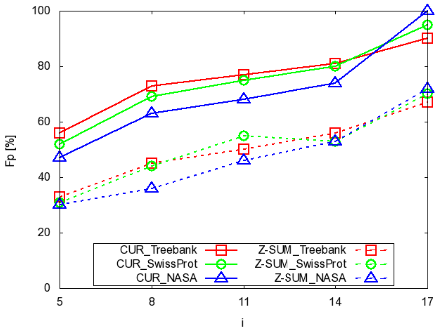

Fig. 9. Variation of F_P w.r.t. the node position i of a 20-node distributed environment on 2,000 × 2,000 OLAP views extracted from real-life XML data sets for the CUR decomposition method and *Zero-Sum*

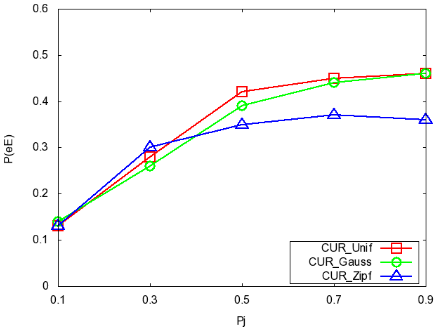

Fig. 10. Variation of $P(e^E)$ w.r.t. the probability p_j on 2,000 × 2,000 OLAP views extracted from synthetic XML data sets for the CUR decomposition method

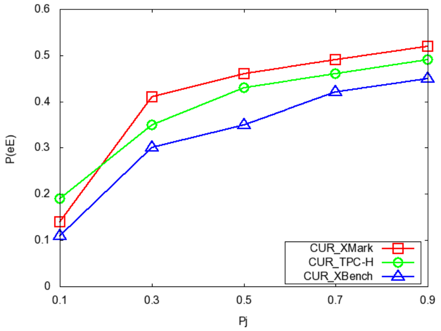

Fig. 11. Variation of $P(e^E)$ w.r.t. the probability p_j on 2,000 × 2,000 OLAP views extracted from benchmark XML data sets for the CUR decomposition method

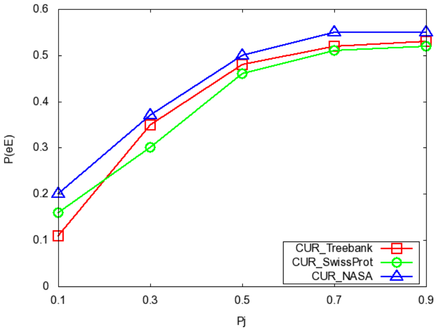

Fig. 12. Variation of $P(e^E)$ w.r.t. the probability p_j on 2,000 × 2,000 OLAP views extracted from real-life XML data sets for the CUR decomposition method

Concluding, our comprehensive experimental campaign conducted on distributed collections of synthetic, benchmark and real-life XML data sets, has clearly demonstrated, under the stressing of a wide variety of experimental parameters, the effectiveness and the efficiency of the proposed privacy preserving distributed OLAP framework, even in comparison with the performance of the state-of-the-art perturbation-based method *Zero-Sum*.

7 Conclusions and Future Work

Starting from the results of our previous research [9], which defined a privacy preserving distributed OLAP framework, in this paper we have introduced a number of theoretical results that nicely extend the capabilities and the potentialities of this framework. These theoretical results are mainly related to some relevant capabilities of the CUR matrix decomposition method, which is the core tool for computing privacy preserving two-dimensional OLAP views in the framework [9]. In order to further support our framework [9], in this paper we have also provided a widespread experimental analysis that fully confirms to us the major practical achievements, in terms of both efficacy and efficiency, due to this framework. Future work is mainly oriented to extend the theoretical results presented here as to make them more robust in order to cover two "difficult" privacy preserving distributed OLAP scenarios of the main framework [9], i.e. (*i*) the need for *multi-resolution OLAP analysis across suitable dimensional hierarchies*, and (*ii*) the presence of *coalition of attackers* that may share *partial knowledge* in order to magnify the capabilities of sensitive data cell inference tasks.

References

[1] Agrawal, S., Haritsa, J.R., Prakash, B.A.: FRAPP: A Framework for High-Accuracy Privacy-Preserving Mining. Data Mining and Knowledge Discovery 18(1), 101–139 (2009)

[2] Agrawal, R., Srikant, R., Thomas, D.: Privacy-Preserving OLAP. Proc. of SIGMOD, 251–262 (2005)

[3] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release. In: Proc. of PODS, pp. 273–282 (2007)

[4] Chan, A.C.-F., Castelluccia, C.: A Security Framework for Privacy-Preserving Data Aggregation in Wireless Sensor Networks. ACM Transactions on Sensor Networks 7(4), art. 29 (2011)

[5] Clifton, C., Kantarcioglu, M., Lin, X., Vaidya, J., Zhu, M.: Tools for Privacy Preserving Distributed Data Mining. SIGKDD Explorations 4(2), 28–34 (2002)

[6] Colliat, G.: OLAP, Relational, and Multidimensional Database Systems. SIGMOD Record 25(3), 64–69 (1996)

[7] Cuzzocrea, A.: Accuracy Control in Compressed Multidimensional Data Cubes for Quality of Answer-based OLAP Tools. In: Proc. of SSDBM, pp. 301–310 (2006)

[8] Cuzzocrea, A.: Privacy Preserving OLAP: Models, Issues, Algorithms. In: Proc. of MIPRO, pp. 1538–1543 (2011)

[9] Cuzzocrea, A., Bertino, E.: A Secure Multiparty Computation Privacy Preserving OLAP Framework over Distributed XML Data. In: Proc. of SAC, pp. 1666–1673 (2010)

[10] Cuzzocrea, A., Russo, V.: Privacy Preserving OLAP and OLAP Security. In: Wang, J. (ed.) Encyclopedia of Data Warehousing and Mining, 2nd edn., pp. 1575–1581. IGI Global (2009)

[11] Cuzzocrea, A., Russo, V., Saccà, D.: A robust sampling-based framework for privacy preserving OLAP. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 97–114. Springer, Heidelberg (2008)

[12] Cuzzocrea, A., Saccà, D.: Balancing Accuracy and Privacy of OLAP Aggregations on Data Cubes. In: Proc. of DOLAP, pp. 93–98 (2010)

[13] Drineas, P., Kannan, R., Mahoney, M.W.: Computing Sketches of Matrices Efficiently and Privacy Preserving Data Mining. In: Proc. of DIMACS PPDM (2004), `http://dimacs.rutgers.edu/Workshops/Privacy/`

[14] Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition. SIAM Journal on Computing 36(1), 184–206 (2006)

[15] Dwork, C.: Differential privacy: A survey of results. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008)

[16] Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery 1(1), 29–53 (1997)

[17] Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press (1989)

[18] Han, J., Pei, J., Dong, G., Wang, K.: Efficient Computation of Iceberg Cubes with Complex Measures. Proc. of SIGMOD, 1–12 (2001)

[19] He, W., Liu, X., Nguyen, H., Nahrstedt, K., Abdelzaher, T.: PDA: Privacy-Preserving Data Aggregation for Information Collection. ACM Transactions on Sensor Networks 8(1), art. 6 (2011)

[20] Hua, M., Zhang, S., Wang, W., Zhou, H., Shi, B.-L.: FMC: An approach for privacy preserving OLAP. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 408–417. Springer, Heidelberg (2005)

[21] Jiang, W., Clifton, C.: A Secure Distributed Framework for Achieving k-Anonymity. Very Large Data Bases Journal 15(4), 316–333 (2006)

[22] Jurczyk, P., Xiong, L.: Distributed anonymization: Achieving privacy for both data subjects and data providers. In: Gudes, E., Vaidya, J. (eds.) Data and Applications Security XXIII. LNCS, vol. 5645, pp. 191–207. Springer, Heidelberg (2009)

[23] Li, F., Luo, B., Liu, P.: Secure and Privacy-Preserving Information Aggregation for Smart Grids. International Journal of Security and Networks 6(1), 28–39 (2011)

[24] Lin, X., Lu, R., Shen, X.: MDPA: Multidimensional Privacy-Preserving Aggregation Scheme for Wireless Sensor Networks. Wireless Communications and Mobile Computing 10(6), 843–856 (2010)

[25] Liu, Y., Sung, S.Y., Xiong, H.: A Cubic-Wise Balance Approach for Privacy Preservation in Data Cubes. Information Sciences 176(9), 1215–1240 (2006)

[26] Mohammed, N., Fung, B.C.M., Hung, P.C.K., Lee, C.-K.: Centralized and Distributed Anonymization for High-Dimensional Healthcare Data. ACM Transactions on Knowledge Discovery from Data 4(4), art. 18 (2010)

[27] Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill (1984)

[28] Pinkas, B.: Cryptographic Techniques for Privacy-Preserving Data Mining. SIGKDD Explorations 4(2), 12–19 (2002)

[29] Sung, S.Y., Liu, Y., Xiong, H., Ng, P.A.: Privacy Preservation for Data Cubes. Knowledge and Information Systems 9(1), 38–61 (2006)

[30] Thompson, S.K., Seber, G.A.F.: Adaptive Sampling. John Wiley & Sons (1996)

[31] Tong, Y., Sun, G., Zhang, P., Tang, S.: Privacy-Preserving OLAP based on Output Perturbation Across Multiple Sites. In: Proc. of PST, p. 46 (2006)

[32] Wang, L., Jajodia, S., Wijesekera, D.: Securing OLAP Data Cubes against Privacy Breaches. In: Proc. of SP, pp. 161–175 (2004)

[33] Wang, L., Wijesekera, D., Jajodia, S.: Cardinality-based Inference Control in Data Cubes. Journal of Computer Security 12(5), 655–692 (2004)

[34] Zhang, N., Zhao, W., Chen, J.: Cardinality-based Inference Control in OLAP Systems: An Information Theoretic Approach. In: Proc. of DOLAP, pp. 59–64 (2004)

[35] Zipf, G.K.: Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology. Addison-Wesley (1949)

[36] Schmidt, A., Waas, F., Kersten, M.L., Carey, M.J., Manolescu, I., Busse, R.: XMark: A Benchmark for XML Data Management. In: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 974–985 (2002)

[37] Transaction Processing Performance Council (2004), `http://www.tpc.org/tpch/default.asp`

[38] Yao, B.B., Özsu, M.T., Khandelwal, N.: XBench Benchmark and Performance Testing of XML DBMSs. In: Proceedings of the 20th IEEE International Conference on Data Engineering, pp. 621–632 (2004)

[39] University of Pennsylvania, The Penn Treebank Project (2002), `http://www.cis.upenn.edu/~treebank/`

[40] Swiss Institute of Bioinformatics, Swiss-Prot Protein Knowledgebase (2005), `http://www.expasy.ch/sprot/`

[41] GSFC/NASA XML Project, NASA (2003), `http://xml.gsfc.nasa.gov`

[42] Cuzzocrea, A., Bertino, E., Saccà, D.: Towards A Theory for Privacy Preserving Distributed OLAP. In: Proceedings of the EDBT/ICDT Workshops, pp. 221–226 (2012)