International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume - 2, Issue - I ISSN (Online): 2454- 1966

Research Manuscript Title

AN LOCAL-RECODING ANONYMIZATION WITH MAPREDUCE FOR SCALABLE BIG DATA PRIVACY PRESERVATION IN CLOUD

S. Deepa[1], S. Muruganandham[2],

[1]M.phil Full Time Research Scholar, [2]Asst. Professor,
Department of Computer Science,
Vivekanandha College of Arts and Sciences for Women,

E-Mail: deephsri@gmail.com, mr.smanand@gmail.com

March – 2016

www.istpublications.com

International Journal of Future Innovative Science and Engineering Research (IJFISER), Volume - 2, Issue – I,
ISSN (Online): 2454- 1966 www.istpublications.com.

AN LOCAL-RECODING ANONYMIZATION WITH MAPREDUCE FOR SCALABLE BIG DATA PRIVACY PRESERVATION IN CLOUD

S. Deepa[1], S. Muruganandham[2],

[1]M.phil Full Time Research Scholar, [2]Asst. Professor,
Department of Computer Science,
Vivekanandha College of Arts and Sciences for Women,
E-Mail: deephsri@gmail.com, mr.smanand@gmail.com

ABSTRACT

Data privacy preservation is one of the most disturbed issues on the current industry. Data privacy issues need to be addressed urgently before data sets are shared on cloud. Data anonymization refers to as hiding complex data for owners of data records. Cloud computing provides promising scalable IT infrastructure to support various processing of a variety of big data applications in sectors such as healthcare and business. Data sets like electronic health records in such applications often contain privacy-sensitive information, which brings about privacy concerns potentially if the information is released or shared to third-parties in cloud. A practical and widely-adopted technique for data privacy preservation is to anonymize data via generalization to satisfy a given privacy model. However, most existing privacy preserving approaches tailored to small-scale data sets often fall short when encountering big data, due to their insufficiency or poor scalability. In this paper, we investigate the local-recoding problem for big data anonymization against proximity privacy breaches and attempt to identify a scalable solution to this problem. Specifically, we present a proximity privacy model with allowing semantic proximity of sensitive values and multiple sensitive attributes, and model the problem of local recoding as a proximity-aware clustering problem. A scalable two-phase clustering approach consisting of a t-ancestors clustering (similar to k-means) algorithm and a proximity-aware agglomerative clustering algorithm is proposed to address the above problem.

We design the algorithms with MapReduce to gain high scalability by performing data-parallel computation in cloud. Extensive experiments on real-life data sets demonstrate that our approach significantly improves the capability of defending the proximity privacy breaches, the scalability and the time-efficiency of local-recoding anonymization over existing approaches.

Key words: **Big data, cloud computing, mapreduce, data anonymization, proximity privacy.**

I.INTRODUCTION

Cloud computing and big data, two disruptive trends at present, pose a significant impact on current IT industry and research communities [1], [2]. Today, a large number of big data applications and services have been deployed or migrated into cloud for data mining, processing or sharing. The salient characteristics of cloud computing such as high scalability and pay as you go fashion make big data cheaply and easily accessible to various organizations through public cloud infrastructure. Data sets in many big data applications often contain personal privacy-sensitive data like electronic health records and financial transaction records. As the analysis of these data sets provides profound insights into a number of key areas of society (e.g., healthcare, medical, government services, research), the data sets are often shared or released to third party partners or the public. The privacy-sensitive information can be divulged with less effort by an adversary as the coupling of big data with public cloud environments disables some traditional privacy protection measures in cloud [3], [4]. This can bring considerable economic loss or severe social reputation impairment to data

owners. As such, sharing or releasing privacy-sensitive data sets to third-parties in cloud will bring about potential privacy concerns, and therefore requires strong privacy preservation.

Data anonymization has been extensively studied and widely adopted for data privacy preservation in noninteractive data sharing and releasing scenarios [5]. Data anonymization refers to hiding identity and sensitive data so that the privacy of an individual is effectively preserved while certain aggregate information can be still exposed to data users for diverse analysis and mining tasks. A variety of privacy models and data anonymization approaches have been proposed and extensively studied recently. However, applying these traditional approaches to big data anonymization poses scalability and efficiency challenges because of the "3Vs", i.e., Volume, Velocity and Variety. The research on scalability issues of big data anonymization has come to the picture [14], but they are only applicable to the sub-tree or multidimensional scheme. Following this line, we investigate the local-recoding scheme herein and attempt to identify a scalable solution to big data localrecoding anonymization. Recently, differential privacy has attracted plenty of attention due to its robust privacy guarantee regardless of an adversary's prior knowledge [16].

II. RELATED WORK

Recently, data privacy preservation has been extensively investigated [5]. We briefly review existing approaches for local-recoding anonymization and privacy models to defense against attribute linkage attacks. In addition, research on scalability issues in existing anonymization approaches is shortly surveyed. Recently, clustering techniques have been leveraged to achieve local-recoding anonymization for privacy preservation.

Xu et al. [9] studied on the anonymization of local recoding scheme from the utility perspective and put forth a bottom-up greedy approach and the top-down counterpart. The former leverages the agglomerative clustering technique while the latter employs the divisive hierarchical clustering technique, both of which pose constraints on the size of a cluster.

Byun et al. [19] formally modeled localrecoding anonymization as the k-member clustering problem which requires the cluster size should not be less than k in order to achieve k-anonymity, and proposed a simple greedy algorithm to address the problem.

Li et al. [18] investigated the inconsistency issue of local-recoding anonymization in data with hierarchical attributes and proposed KACA (K-Anonymization by Clustering in Attribute hierarchies) algorithms. We proposed a set of constant factor approximation algorithms for two clustering based anonymization problems, i.e., r-GATHER and r-CELLULAR CLUSTERING, where cluster centers are published without generalization or suppression. However, existing clustering approaches for local-recoding anonymization mainly concentrate on record linkage attacks, specifically under the k-anonymity privacy model, without paying any attention to privacy breaches incurred by sensitive attribute linkage. On the contrary, our research takes both privacy concerns into account. attribute linkage attacks. However, the data utility of the resultant anonymous data is heavily influenced by the choice of splitting attributes and values, while local recoding does not involve such factors. Our approach leverages clustering to accomplish local recoding because it is a natural and effective way to anonymize data sets at a cell level.

III. PROBLEM ANALYSIS

A.Local-Recoding Anonymization Scheme:

To facilitate subsequent discussion, we briefly introduce the concept of local-recoding anonymization as background knowledge. Local recoding, also known as cell generalization, is one of the schemes outlined in [5]. Other schemes

include full-domain, sub-tree and multidimensional anonymization. Local recoding generalizes a data set at the cell level, while global recoding generalizes them at the domain level. The last three schemes mentioned above are global recoding. Generally, local recoding minimizes the data distortion incurred by anonymization, and therefore produces better data utility than global recoding.

B. MapReduce Basics:

MapReduce [28], a parallel and distributed large-scale data processing paradigm, has been extensively researched and widely adopted for big data applications recently [29]. Integrated with infrastructure resources provisioned by cloud systems, MapReduce becomes much more powerful, elastic and cost-effective due to the salient characteristics of cloud computing. A typical example is the Amazon Elastic Map- Reduce service.

C. Motivation and Problem Analysis:

In this section, we analyze the problems of existing approaches for local-recoding anonymization from the perspectives of proximity privacy and scalability. Further, challenges of designing scalable MapReduce algorithms for proximity-aware local recoding are also identified.

IV. PROXIMITY-AWARE CLUSTERING PROBLEM OF LOCAL-RECODING ANONYMIZATION

Due to the non-monotonicity property of proximity-aware privacy models and characteristics of local recoding, clustering is a natural and promising way to group both quasiidentifier attributes and sensitive attributes.

A.Privacy Model:

In big data scenarios, multiple sensitive attributes are often contained in data sets, while existing proximity-aware privacy models assume only one single sensitive attribute, either categorical or numerical. Hence, we assume multiple sensitive attributes in our privacy model, including both categorical and numerical attributes. As the discussion of proximity privacy attacks stems from numerical attributes, existing proximity-aware privacy models assume that categorical attribute values have no sense of semantic proximity [12], [21]. That is, categorical values are only examined whether they are exactly identical or different. Also, privacy models for categorical attributes only aims at avoiding exact reconstruction of sensitive values via limiting the number or distribution of sensitive values without considering semantic proximity [7], [8]. However, sensitive categorical values often have the sense of semantic proximity in real-life applications because the values are usually organized in a taxonomy tree in terms of domain specific knowledge. For instance, a taxonomy tree of diseases is presented in [27].

V. TWO-PHASE PROXIMITY-AWARE CLUSTERING USING MAPREDUCE

Except where otherwise noted, the proximity-aware clustering problem refers to the single-objective proximity-aware clustering problem hereafter. To address the SPAC problem in big data scenarios, we propose a two-phase clustering approach where agglomerative clustering method and point-assignment clustering method are employed in the two phases, respectively. We outline the sketch of the twophase clustering approach in Section a. Then, the algorithmic details of the two phases are elaborated in Sections b and c, respectively. We illustrate the execution process of our approach and analyze the performance in Section d.

A.Sketch of Two-Phase Clustering:

In order to choose proper clustering methods for the SPAC problem, some observations of clustering problems for data anonymization should be taken into account. First, the parameter k in the k-anonymity privacy model is relatively small compared with the scale of a data set in big data scenarios. Since the upper-bound of the size of a cluster for local-recoding anonymization is 2k 1, the size of clusters is also relatively small. Accordingly, the number of clusters will be quite large. Second, under the condition that the size of any cluster is not less than k, the smaller a cluster is, the more it is preferred. The reason is that this tends to incur less data distortion. Ideally, the size of all clusters is exactly k. Third, the intrinsic clustering architecture in a data set is helpful for local-recoding anonymization, but building such an architecture is not the final purpose. Given the observations above, the agglomerative clustering method is suitable for local-recoding anonymization, as the stopping criterion can be set as whether the size of a cluster reaches to k. Moreover, the agglomerative clustering method can achieve minimum data distortion in the sense of the defined distance measure. Most existing approaches for k-anonymity mentioned in Section 2 employ greedy agglomerative clustering approaches. But they construct clusters in a greedy manner rather than combine the two clusters that have the minimum distance in each round, which results in more data distortion. But the optimal agglomerative clustering method suffers the scalability problem when handling large-scale data sets. Its time complexity is Oðn2 log nÞ with utilizing a priority queue. Worse still, the agglomerative method is serial, which makes it difficult to be adapted to parallel environments like MapReduce.

B. t-Ancestor Clustering for Data Partitioning:

One core problem in the point-assignment method is how to represent a cluster. Similar to t-medians [32], We propose to leverage the ‘ancestor’ of the records in a cluster to represent the cluster. More precisely, an ancestor of a cluster refers to a data record whose attribute value of each categorical quasiidentifier is the lowest common ancestor of the original values in the cluster. Each numerical quasi-identifier of an ancestor record is the median of original values in the cluster.
The notion of ancestor record also attempt to capture the logical centre of a cluster like t-means/medians, but t-ancestors clustering is more suitable for anonymization due to categorical attributes in the clustering problem herein. To facilitate t-ancestors clustering, we take quasi-identifier attributes but sensitive ones into consideration. This will rarely affect the proximity of sensitive values in a final cluster, because the clustering granularity in the first phase is rather coarse. Accordingly, we leverage the distance measure (8) to calculate the distance between a data record and an ancestor. Usually, an ancestor is not a real data record in the data set, but the (8) can still be employed to calculate the distance between two vectors of attribute values. Except where otherwise noted, a record r in this section refers to the quasi-identifier part.

C. Proximity-Aware Agglomerative Clustering:

we leverage the proximity-aware distance measure (9) for the agglomerative clustering in this section. In the agglomerative clustering method, each data record is regarded as a cluster initially, and then two clusters are picked to be merged in each round of iteration until some stopping criteria are satisfied. Usually, two clusters with the shortest distance are merged. Thus, one core problem of the agglomerative clustering method is how to define the distance between two clusters. To coincide with the objective in the SPAC problem, we leverage the complete- linkage distance in our agglomerative clustering algorithm, i.e., the distance between two clusters equals to the weighted distance between those two records (one in each cluster) that are farthest away from each other. In fact, after merging such two clusters, the distance between them is the diameter of the new cluster.

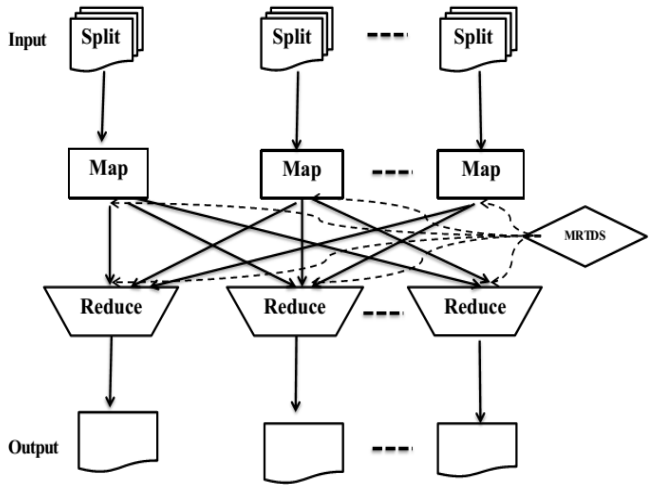

Fig. 1. Execution process overview of two-phase clustering.

D. Execution Process and Performance Analysis:

In order to demonstrate the proposed two-phase proximityaware clustering approach visually, its execution process overview is illustrated in Fig. 1. The bold solid arrow line in the right indicates the timeline of the process. Seen from Fig. 1, the three MapReduce jobs are coordinated together to accomplish the local-recoding anonymization. From the perspective of control flow, our approach is partially parallel because the first phase is sequential with iteration of the SeedUpdate job, while the second phase is parallel. However, our approach is fully parallel from the perspective of data flow. The light solid arrow lines in Fig. 1 represent data flows in the canonical MapReduce framework, while the dashed arrow lines stand for data flows of dispatching seeds to distributed caches and the data flow of updating seeds. An Original data set is read by Map functions and its splits are processed in a parallel manner. As such, the twophase clustering approach can handle large-scale data sets. Note that the amount of seeds (or ancestors) in the Seed- Update job is relatively small with proper parameter t, so that they can be delivered to distributed caches efficiently.

VI. EXPERIMENT EVALUATION

A. Overall Comparison:

To evaluate the effectiveness and efficiency of the Proximity- Aware Clustering approach, we compare it with the kmember Clustering approach proposed in [18], which also represents the approaches for local recoding in [9], [19]. The kMC approach is the state-of-the-art approach for localrecoding anonymization with clustering techniques. As to effectiveness, we consider three factors, namely, the resistibility to proximity breaches, data distortion and scalability.

B.Experiment Evaluation:

Our experiments are conducted in a cloud environment named U-Cloud [4]. The Hadoop cluster is built on Ucloud. For more details about U-Cloud, please refer to [4]. The data set Census-Income (KDD) [35] is utilized in our experiments. Its subset Adult data set has been commonly used as a de facto benchmark for testing anonymization algorithms [9], [18], The data set is sanitized via removing records containing missing values and attributes with extremely skewed

distributions. We obtain a sanitized data set with 153,926 records, from which data sets in the following experiments are sampled. Twelve attributes are chosen out of the original 40 ones, including nine (four numerical and five categorical) quasi-identifier ones and three (two numerical and one categorical) sensitive ones.
Both PAC and kMC are implemented in Java. Further, PAC is implemented with the standard Hadoop MapReduce API and executed on a Hadoop cluster built on Open- Stack in U-cloud. The Hadoop cluster consists of 20 virtual machines with type m1.medium which has two virtual CPUs and 4 GB Memory. kMC is executed on one virtual machine with type m1.medium. The maximum heap size of Java VM is set as 4 GB when running kMC. Each round of experiment is repeated 10 times. The mean of the measured results is regarded as the representative.

C.Experiment Process and Results:

We conduct two groups of experiments in this section to evaluate the effectiveness and efficiency of our approach. In the first one, we explore whether proximity-aware clustering leads to larger dissimilarity by comparing PAC with kMC from the perspectives of resistibility to proximity breaches and data distortion. The other investigates the scalability and efficiency.

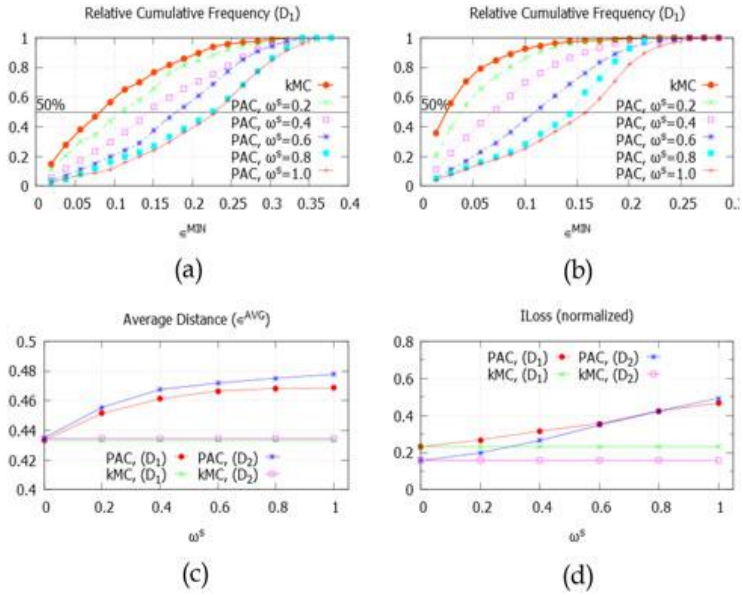

Fig. 2. Change of RCF .

Meanwhile, it can be seen from Fig. 2d that the normalized value of ILoss rises as well when vs grows, indicating that more data distortion is incurred. In fact, the gain of dissimilarity is at the cost of data utility, which. Fortunately, one can choose a proper weight vs to make a good trade-off between the capability of defending proximity attacks and data utility. For example, vs ¼ 0:6 seems to be a good choice via observing Fig. 2.

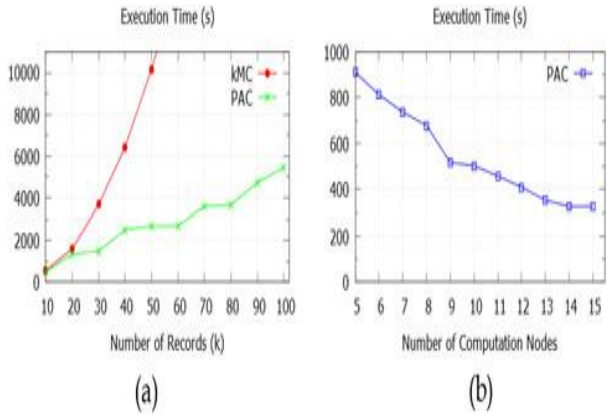

Fig. 3. Change of execution time w.r.t. data size and the number of computation nodes.

From Fig. 3, we can see the execution time of both PAC and kMC go up when the number of records increases although some slight fluctuations exist. The fluctuations, mainly incurred by the data distribution of each data set, will not affect the trends of execution time. Notably, the execution time of kMC surges from hundreds of seconds to more than 10,000 s within only four steps, while that of PAC goes linearly and stably. The dramatic increase of PAC execution time illustrates that the intrinsic time complexity of kMC makes it hard to scale over big data. The difference of execution time between kMC and PAC becomes larger and larger when the data size is growing. This trend demonstrates that our approach becomes much more scalable and efficient compared with kMC in big data scenarios. Fig. 3 exhibits the change of execution time of PAC with respect to the number of computation nodes ranging from 5 to 15. The number of data records set as 10,000, and other settings are the same as Fig. 5a. It can be seen from Fig. 3b. that the execution time decreases in a nearly linear manner when the number of computation nodes is getting larger. In terms of the tendency, we maintain that PAC is linearly scalable with respect to the number of computation nodes. Hence, PAC is able to manage to handle big data local recoding in a timely fashion in cloud where computation resources are offered on demand. Above all, the experimental results demonstrate that our approach integrating proximity of sensitive attributes
into clustering, significantly improves the capability of defending proximity attacks, the scalability and efficiency of local-recoding anonymization over existing approaches.

VII.CONCLUSIONS AND FUTURE WORK

In cloud environment, the privacy preservation for data analysis, share and mining is a challenging research issue due to increasingly larger volumes of datasets, thereby requiring intensive investigation. Based on the contributions herein, we plan to integrate our approach with Apache Mahout, a MapReduce based scalable machine learning and data mining library, to achieve highly scalable privacy preserving big data mining or analytics.

REFERENCES

[1] S. Chaudhuri, “What next?: A half-dozen data management research goals for big data and the cloud,” in Proc. 31st Symp. Principles Database Syst., 2012, pp. 1–4.
[2] L. Wang, J. Zhan, W. Shi, and Y. Liang, “In cloud, can scientific communities benefit from the economies of scale?” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 2, pp. 296–303, Feb. 2012.
[3] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, “Data mining with big data,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 1, pp. 97–107, Jan. 2014.

[4] X. Zhang, C. Liu, S. Nepal, S. Pandey, and J. Chen, "A privacy leakage upper bound constraint-based approach for cost-effective privacy preserving of intermediate data sets in cloud," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1192–1202, Jun. 2013.

[5] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," ACM Comput. Survey, vol. 42, no. 4, pp. 1–53, 2010.

[6] L. Sweeney, "K-anonymity: A model for protecting privacy," Int. J. Uncertainty Fuzziness, vol. 10, no. 5, pp. 557–570, 2002.

[7] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," ACMTrans. Knowl.Discovery Data, vol. 1, no. 1, 2007.

[8] N. Li, T. Li, and S. Venkatasubramanian, "Closeness: A new privacy measure for data publishing," IEEE Trans. Knowl. Data Eng., vol. 22, no. 7, pp. 943–956, Jul. 2010.

[9] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W. C. Fu, "Utilitybased anonymization using local recoding," in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data, 2006, pp. 785–790.

[10] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization techniques for large-scale datasets," ACM Trans. Database Syst., vol. 33, no. 3, pp. 1–47, 2008.

[11] G. Aggarwal, R. Panigrahy, T. Feder, D. Thomas, K. Kenthapadi, S. Khuller, and A. Zhu, "Achieving anonymity via clustering," ACM Trans. Algorithms, vol. 6, no. 3, 2010.

[12] T. Wang, S. Meng, B. Bamba, L. Liu, and C. Pu, "A general proximity privacy principle," in Proc. IEEE 25th Int. Conf. Data Eng., 2009, pp. 1279–1282.

[13] T. Iwuchukwu and J. F. Naughton, "K-Anonymization as spatial indexing: Toward scalable and incremental anonymization," in Proc. 33rd Int. Conf. Very Large Data Bases, 2007, pp. 746–757.

[14] X. Zhang, L. T. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using Mapreduce on cloud," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 363–373, Feb. 2014.

[15] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, and J. Chen, "A hybrid approach for scalable sub-tree anonymization over big data using Mapreduce on cloud," J. Comput. Syst. Sci., vol. 80, no. 5, pp. 1008–1020, 2014.

[16] Y. Yang, Z. Zhang, G. Miklau, M. Winslett, and X. Xiao, "Differential privacy in data publication and analysis," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2012, pp. 601–606.

[17] J. Lee and C. Clifton, "Differential identifiability," in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 1041–1049. [18] J. Li, R. C.-W. Wong, A. W.-C. Fu, and J. Pei, "Anonymization by local recoding in data with attribute hierarchical taxonomies," IEEE Trans. Knowl. Data Eng., vol. 20, no. 9, pp. 1181–1194, Sep. 2008.

[19] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient K-anonymization using clustering techniques," in Proc. 12th Int. Conf. Database Syst. Adv. Appl., 2007, pp. 188–200.

[20] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in Proc. IEEE 23rd Int. Conf. Data Eng., 2007, pp. 116–125.