ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Privacy Maintenance and Accuracy Constrained Access Control Mechanism

A.Asha priya
M.Tech student,
Department of CSE
Swarnandhra College of Engineering and Technology
Narsapuram, Andhra Pradesh, India.

N.Tulasi raju
Associate professor
Department of CSE
Swarnandhra College of Engineering and Technology
Narsapuram, Andhra Pradesh, India.

Abstract:
Access control mechanisms protect sensitive information from unauthorized users. However, when sensitive information is shared without presence of Privacy Protection Mechanism (PPM), an authorized user can still compromise the privacy of a person leading to identity disclosure. A PPM can use suppression and generalization of relational data to anonymize and satisfy privacy requir ements, e.g., k-anonymity and l-diversity, against identity and attribute disclosure. However, privacy is achieved at the cost of precision of authorized information. In this paper, we propose an accuracy-constrained privacy-preserving access control framework. The access control policies define selection predicates available to roles while the privacy requirement is to satisfy the k-anonymity or l-diversity. An additional constraint that needs to be satisfied by the PPM is the imprecision bound for each selection predicate. The techniques for workload-aware anonymization for selection pred icates have been discussed in the literature. However, to the best of our knowledge, the problem of satisfying the accuracy constraints for multiple roles has not been studied before. In our formulation of the aforementioned problem, we propose heuristics for anonymization algorithms and show empirically that the proposed approach satisfies imprecision bounds for more permissions and has lower total imprecision than the current state of the art.Along with the proposed system, we also gone through the extension for the article. As the extension of this article, we have gone through incremental data and cell level access control.

Index Terms—Access control, privacy, k-anonymity, query evaluation

I. INTRODUCTION

Different organizations collect and analyze consumer data to improve their quality of services. We use Access Control Mechanisms (ACM) to provide only authorized information to users. However, sensitive information can still be misused by authorized users to compromise the privacy of consumers. The concept of privacy-preservation for sensitive data can require the enforcement of privacy policies or the protection against identity disclosure by satisfying some privacy requirements. In this paper, we investigate privacy-preservation from the anonymity aspect. The sensitive information, even after the removal of identifying attributes, is still susceptible to linking attacks by the authorized users. This problem has been studied extensively in the area of micro data publishing and privacy definitions, e.g., k-anonymity , l-diversity, and variance diversity. Anonymization algorithms use suppression and generalization of records to satisfy privacy requirements with minimal distortion of micro data. The anonymity techniques can be used with an access control mechanism to ensure both security and privacy of the sensitive information. The privacy is achieved at the cost of accuracy and imprecision is introduced in the authorized information under an access control policy.

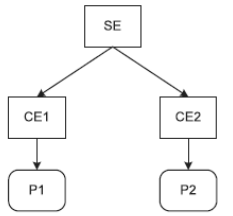
We use the concept of imprecision bound for each permission to define a threshold on the amount of imprecision that can be tolerated. Existing workload aware anonymization techniques, minimize the imprecision aggregate for all queries and the imprecision added to each permission/query in the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

anonym zed micro data is not known. Making the privacy requirement more stringent (e.g., increasing the value of k or l) results in additional imprecision for queries. However, the problem of satisfying accuracy constraints for individual permissions in a policy/workload has not been studied before. The heuristics proposed in this paper for accuracy-constrained privacy-preserving access control are also relevant in the context of workload-aware anonymization. In this paper the focus is on a static relational table that is anonym zed only once. To exemplify our approach, role-based access control is assumed. However, the concept of accuracy constraints for permissions can be applied to any privacy-preserving security policy, e.g., discretionary access control.

Example 1 (Motivating Scenario). Syndromic surveillance systems are used at the state and federal levels to detect and monitor threats to public health. The department of health in a state collects the emergency department data (age, gender, location, time of arrival, symptoms, etc.) from county hospitals daily. Generally, each daily update consists of a static instance that is classified into syndrome categories by the department of health. Then, the surveillance data is anonym zed and shared with departments of health at each county.

Role	Designation
SE	State Epidemiologist
CE1	County 1 Epidemiologist
CE2	County 2 Epidemiologist



Permission	Authorized Query Predicate
P1	Location = County 1 ∧ Age = 15-65 ∧ Syndrome = Influenza
P2	Location = County 2 ∧ Age = 15-65 ∧ Syndrome = Influenza

Fig. 1. Access control policy.

An access control policy is given in Fig. 1 that allows the roles to access the tuples under the authorized predicate, e.g., Role CE1 can access tuples under Permission P1. The epidemiologists at the state and county level suggest community containment measures, e.g., isolation or quarantine according to the number of persons infected in case of a flu outbreak. According to the population density in a county, an epidemiologist can advise isolation if the number of persons reported with influenza are greater than 1,000 and quarantine if that number is greater than 3,000 in a single day. The anonymization adds imprecision to the query results and the imprecision bound for each query ensures that the results are within the tolerance required. If the imprecision bounds are not satisfied then unnecessary false alarms are generated due to the high rate of false positives.

2 BACKGROUND

Here, In this section, role-based access control and privacy definitions based on anonymity are over-viewed. Query evaluation semantics, imprecision, and the Selection Mondrian algorithm are briefly explained.

Given a relation $\mathbf{T = \{A_1, A_2, \ldots, A_n\}}$, where $\mathbf{A_i}$ is an attribute, T_ is the anonym zed version of the relation T. We assume that T is a static relational table.

2.1 Access Control for Relational Data

Fine-grained access control for relational data allows to define tuple-level permissions, e.g., Oracle VPD and SQL. For evaluating user queries, most approaches assume a Truman model. In this model, a user query is modified by the access control mechanism and only the authorized tuples are returned. Column level access control allows queries to execute on the authorized column of the relational data only. Cell level access control for relational data is implemented by replacing the unauthorized cell values by NULL values.

Role-based Access Control (RBAC) allows defining permissions on objects based on roles in an organization. An RBAC policy configuration is composed of a set of Users (U), a set of Roles (R), and a set of Permissions (P). For the relational RBAC model, we assume that the selection predicates on the QI attributes define a permission. UA is a user-to-role (U X R) assignment relation and PA is a roleto-permission (R X P) assignment relation. A role hierarchy (RH) defines an inheritance relationship

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

among roles and is a partial order on roles (R X R). Each permission defines a hyper-rectangle in the tuple space and all the tuples enclosed by this hyper-rectangle are authorized to the role assigned to the permission. In practice, when a user assigned to a role executes a query, the tuples satisfying the conjunction of the query predicate and the permission are returned.

2.2 Anonymity Definitions

Definition 1 (Equivalence Class (EC)). *An equivalence class is a set of tuples having the same QI attribute values.*

Definition 2 (k-anonymity Property). *A table T_ satisfies the k-anonymity property if each equivalence class has k or more tuples.*

Definition 3 (Query Imprecision). *Query Imprecision is defined as the difference between the number of tuples returned by a query evaluated on an anonym zed relation T* and the number of tuples for the same query on the original relation T.*

The imprecision for query Q_i *is denoted by* imp_{Q_i},

$$imp_{Q_i} = |Q_i(T^*)| - |Q_i(T)|, \text{ where } |Q_i(T^*)| = \sum_{EC \text{ overlaps } Q_i} |EC|.$$

k-anonymity is prone to homogeneity attacks when the sensitive value for all the tuples in an equivalence class is the same. To counter this shortcoming, l-diversity has been proposed and requires that each equivalence class of T* contain at least l distinct values of the sensitive attribute. For sensitive numeric attributes, an l-diverse equivalence class can still leak information if the numeric values are close to each other. For such cases, variance diversity has been proposed that requires the variance of each equivalence class to be greater than a given variance diversity parameter.

	QI_1	QI_2	S_1
ID	Age	Zip	Disease
1	5	15	Flu
2	15	25	Fever
3	28	28	Diarrhea
4	25	15	Fever
5	22	28	Flu
6	32	35	Fever
7	38	32	Flu
8	35	25	Diarrhea

(a) Sensitive table

ID	QI_1 Age	QI_2 Zip	S_1 Disease
1	0-20	10-30	Flu
2	0-20	10-30	Fever
3	20-30	10-30	Diarrhea
4	20-30	10-30	Fever
5	20-30	10-30	Flu
6	30-40	20-40	Fever
7	30-40	20-40	Flu
8	30-40	20-40	Diarrhea

(b) 2-anonymous Table

Fig. 2. Generalization for k-anonymity and l-diversity.

The table in Fig. 2a does not satisfy k-anonymity because knowing the age and zip code of a person allows associating a disease to that person. The table in Fig. 2b is a 2-anonymous and 2-diverse version of table in Fig. 2a. The ID attribute is removed in the anonym zed table and is shown only for identification of tuples. Here, for any combination of selection predicates on the zip code and age attributes, there are at least two tuples in each equivalence class. In Section 4, algorithms are presented for k-anonymity only. However, the experiments are performed for both l-diversity and variance diversity using the proposed heuristics for partitioning.

2.3 Top Down Selection Mondrian (TDSM)

The objective of TDSM is to minimize the total imprecision for all queries while the imprecision bounds for queries have not been considered. TDSM starts with the whole tuple space as one partition and then partitions are recursively divided till the time new partitions meet the privacy requirement. To divide a partition, two decisions need to be made, i) Choosing a split value along each dimension, and ii) Choosing a dimension along which to split. In the TDSM algorithm [5], the split value is chosen along the median and then the dimension is selected along which the sum of imprecision for all queries is minimum.

3. ANONYMIZATION WITH IMPRECIS ION BOUNDS

3.1 Definitions

Definition (Query Imprecision Slack). *The query imprecision slack, denoted by sQi for a Query, say Qi, is defined as the difference between the query imprecision bound and the actual query imprecision.*

$$s_{Q_i} = \begin{cases} B_{Q_i} - imp_{Q_i}, & \text{if } imp_{Q_i} \le B_{Q_i}, \\ 0, & \text{otherwise.} \end{cases}$$

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Definition (Query Imprecision Bound). *The query imprecision bound, denoted by* $\boldsymbol{BQ_i}$ *, is the total imprecision acceptable for a query predicate* $\boldsymbol{Q_i}$ *and is preset by the access control administrator.*

Example 3. Assume two range queries as given in Fig. 3. The queries are the shaded rectangles with solid lines while the partitions are the regions enclosed by rectangles with dashed lines. The imprecision bounds for Queries Q1 and Q2 are preset to 2 and 0. The partitioning given in Fig. 2b does not satisfy the imprecision bounds. However, the partitioning given in Fig. 3 satisfies the bounds for Queries Q1 and Q2 as the imprecision for Q1 and Q2 is 2 and 0, respectively.

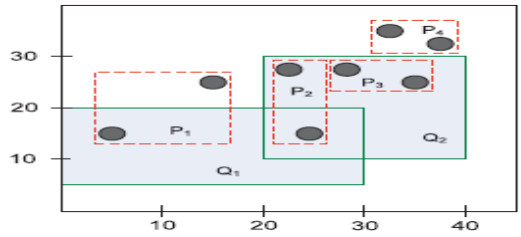


Fig. 3. Anonymization satisfying imprecision bounds.

Definition (Query Cut). *A query cut is defined as the splitting of a partition along the query interval values. For a query cut using Query Qi, both the start of the query interval* $(a_j^{Q_i})$ *and the end of the query interval* $(b_j^{Q_i})$ *are considered to split a partition along the jth dimension.*

Example 4. A comparison of median cut and query cut is given in Fig. 4 for 3-anonymity. The rectangle with solid lines represents Query Q1. While, the rectangles with dotted lines represent partitions. In Fig. 4a the tuples are partitioned according to the median cut and even after dividing the tuple space into four partitions there is no reduction in imprecision for the Query Q1. However, for query cuts in Fig. 4b the imprecision is reduced to zero as partitions are either non-overlapping or fully enclosed inside the query region.

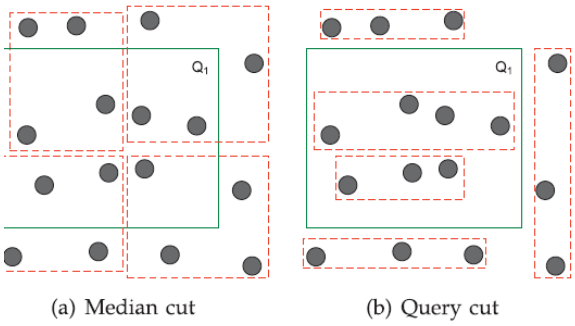


Fig. 4. Comparison of median and query cut.

3.3.1 Access Control Enforcement

The exact tuple values in a relation are replaced by the generalized values after the anonymization. In this case, access control enforcement over the generalized data needs to be defined. In this section, we discuss the Relaxed and Strict access control enforcement mechanisms over anonymized data. The access control enforcement by reference monitor can be of the following two types:

1. Relaxed. Use overlap semantics to allow access to all partitions that are overlapping the permission.
2. Strict. Use enclosed semantics to allow access to only those partitions that are fully enclosed by the permission.

Relaxed enforcement violates the authorization predicate by giving access to extra tuples but is beneficial for applications where low cost of a false alarm is tolerable as compared to the risk associated with a missed event. Examples include epidemic surveillance and airport security. On the other hand, strict enforcement is suitable for applications where a high risk is associated with a false alarm as compared to the cost of a missed event. An example is a false arrest in case of shoplifting. Here in this paper, we first focus on relaxed enforcement. However the proposed methods for anonymization are also valid for strict enforcement because the proposed heuristics reduce the overlap between partitions and queries.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4 HEURISTICS FOR PARTITIONING

4.1 Top-Down Heuristic 1 (TDH1)

Algorithm 1: TDH1

Input : T, k, Q, and B_{Q_j}
Output: P

1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** ($CP_i \in CP$) **do**
3 Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4 Sort queries QO in increasing order of B_{Q_j}
5 **while** *(feasible cut is not found)* **do**
6 Select query from QO
7 Create query cuts in each dimension
8 Select dimension and cut having least overall imprecision for all queries in Q
9 **if** *(Feasible cut found)* **then**
10 Create new partitions and add to CP
11 **else**
12 Split CP_i recursively along median till anonymity requirement is satisfied
13 Compact new partitions and add to P
14 return (P)

The TDH1 algorithm is listed in Algorithm 1. In the first line, the whole tuple space is added to the set of candidate partitions. In the Lines 3-4, the query overlapping the candidate partition with least imprecision bound and imprecision greater than zero is selected. The while loop in Lines 5-8 checks for a feasible split of the partition along query intervals. If a feasible cut is found, then the resulting partitions are added to CP. Otherwise, the candidate partition is checked for median cut in Line 12. A feasible cut means that each partition resulting from split should satisfy the privacy requirement.

4.2 Top-Down Heuristic 2 (TDH2)

In the TDH2, the query bounds are updated as the partitions are added to the output. This update is carried out by subtracting the $ic_{P_i}^{Q_j}$ value from the imprecision bound B_{Q_j} of each query, for a Partition, say Pi, that is being added to the output. For example, if a partition of size k has imprecision 5 and 10 for Queries Q1 and Q2 with imprecision bound 100 and 200, then the bounds are changed to 95 and 190, respectively. The best results are achieved if the kd-tree traversal is depth-first (preorder). Preorder traversal for the kd-tree ensures that a given partition is recursively split till the leaf node is reached.

The algorithm for TDH2 is listed in Algorithm 2. There are two differences compared to TDH1. First, the kd-tree traversal for the for loop in Lines 2-14 is preorder. Second, in Line 14, the query bounds are updated as the partitions are being added to the output (P). The time complexity of TDH2 is $\mathcal{O}(d|Q|^2n^2)$, which is the same as that of TDH1.

Algorithm 2: TDH2

Input : T, k, Q, and B_{Q_j}
Output: P

1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** ($CP_i \in CP$) **do**
// Depth-first(preorder) traversal
3 Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4 Sort queries QO in increasing order of B_{Q_j}
5 **while** *(feasible cut is not found)* **do**
6 Select query from QO
7 Create query cuts in each dimension
8 Select dimension and cut having least overall imprecision for all queries in Q
9 **if** *(Feasible cut found)* **then**
10 Create new partitions and add to CP
11 **else**
12 Split CP_i recursively along median till anonymity requirement is satisfied
13 Compact new partitions and add to P
14 Update B_{Q_j} according to $ic_{P_i}^{Q_j}$, $\forall Q_j \in Q$
15 return (P)

4.3 Top-Down Heuristic 3 (TDH3)

In the TDH3 algorithm, we modify TDH2 so that the time complexity of $\mathcal{O}(d|Q|nlgn)$ can be achieved at the cost of reduced precision in the query results. Given a partition, TDH3 checks the query cuts only for the query having the lowest imprecision bound. Also, the second constraint is that the query cuts are feasible only in the case when the size ratio of the resulting partitions is not highly skewed. We use a skew ratio of 1:99 for TDH3 as a threshold. If a query cut results in one partition h having a size greater than hundred times the other, then that cut is ignored. TDH3 algorithm is listed in Algorithm 3. In Line 4 of Algorithm 3, we use only one query for the candidate

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

cut. In Line 6, the partition size ratio condition needs to be satisfied for a feasible cut. If a feasible query cut is not found, then the partition is split along the median as in Line 11.

5 IMPROVING THE NUMBER OF QUERIES SATISFYING THE IMPRECISION BOUNDS

In Section 3, the query imprecision slack is defined as the difference between the query bound and query imprecision. This query imprecision slack can help

Algorithm 3: TDH3

Input : T, k, Q, and B_{Q_j}
Output: P

1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** ($CP_i \in CP$) **do**
  // Depth-first(preorder) traversal
3 Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4 Select query from QO with smallest B_{Q_j}
5 Create query cuts in each dimension
6 Reject cuts with skewed partitions
7 Select dimension and cut having least overall imprecision for all queries in Q
8 **if** *(Feasible cut found)* **then**
9 Create new partitions and add to CP
10 **else**
11 Split CP_i recursively along median till anonymity requirement is satisfied
12 Compact new partitions and add to P
13 Update B_{Q_j} according to $ic_{P_i}^{Q_j}, \forall Q_j \in Q$
14 return (P)

satisfy queries that violate the bounds by only a small margin by increasing the imprecision of the queries having more slack. In repartitioning step, we consider only the first two groups of queries that fall within 10 percent and 10-25 percent of the bound only and these queries are added to the Candidate Query set (CQ), while all queries satisfying the bounds are added to the query set SQ. The output partitions are all the leaf nodes in the kd-tree. For repartitioning, we only consider those pairs of partitions from the output that are siblings in the kd-tree and have imprecision greater than zero for the queries in the candidate query set. These pairs of partitions are then added to the candidate partition set for repartitioning. Merging such a pair of sibling leaf nodes ensures that we still get a hyper-rectangle and the merged partition is non-overlapping with any other output partition. The repartitioning is first performed for the set of queries within 10 percent of the bound. The partitions that are modified are removed from the candidate set and then the second group of queries is checked. The algorithm for repartitioning is listed as Algorithm 4. In Lines 6-9, we check if a query cut along any dimension exists that reduces the total imprecision for the queries in CQ Set while still satisfying the bounds of the queries in SQ. If such a cut exists, then the old partitions are removed and the new ones are added to Output P in Lines 11-12. After every iteration, the imprecision of the queries in Set CQ is checked. If the imprecision is less than the bound for any query, then as in Line 15 that query is moved from Set CQ to SQ. The proposed algorithm in the experiments satisfies most of the queries from the first group and only a few queries from the second group. This repartitioning step is equivalent to partitioning all the leaf nodes that in the worst case can take $\mathcal{O}(|Q|n)$ time for each candidate query set.

Algorithm 4: Repartitioning

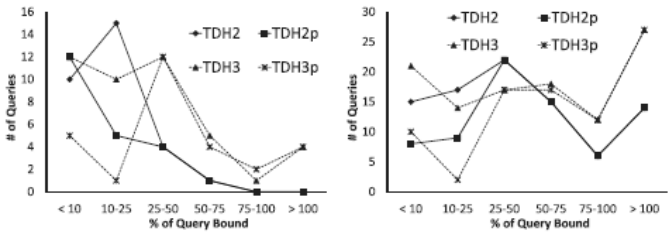
Input : T, k, Q, P, and B_q
Output: P

1 Initialize SQ, CQ, and CP
2 Add $q \in Q$ satisfying bound to SQ
3 Add $q \in Q$ violating bound by 10% to Candidate Query set(CQ)
4 Add all sibling leaf node pairs having $\sum_{q \in CQ}(ic_{P_i}^{q_j} + ic_{P_{i+1}}^{q_j}) > 0$ to Candidate Partition(CP)
5 **for** ($CP_i \in CP$) **do**
6 Merge the first pair CP_i and CP_{i+1}
7 Select q from CQ with the least imprecision greater than the imprecision bound
8 Create the candidate cuts in each dimension
9 Select the cut and the dimension satisfying all $q \in SQ$ with the minimum imprecision $\forall q \in CQ$
10 **if** *(feasible cut found)* **then**
11 Remove CP_i and CP_{i+1} from CP and P
12 Add new partitions to P
13 **for** ($q \in CQ$) **do**
14 **if** ($Imp_q < B_q$) **then**
15 Remove q from CQ and add to SQ
16 return (P)

In the experiments, we set the value of k to 5 and 7 with a query imprecision bound of 30 percent of the query size. The results for repartitioning are given in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Fig. 18. TDH2p and TDH3p are the results after the repartitioning step. Observe that most of the queries in the 10 percent group have been satisfied, while for the 10-25 percent group, some of these have been satisfied while the others have moved into the first group. Repartitioning of the other groups of queries reduces the total imprecision but the gains in terms of having more queries satisfying bounds are not worthwhile.



(a) k = 5 and Bound = 30% (b) k = 7 and Bound = 30%

Fig. 5 Improvements after repartitioning for k-anonymity for the Census data set.

6. EXTENSION FOR PROPOSED SYSTEM

In this extended model, a user query is modified by the access control mechanism and only the authorized tuples are returned. Since Cell level access control is proving a best solution than column level access, along with Column level access we also implemented cell level access control and incremental data for our system.

If given a relation **T = {A_1,A_2, . . .,A_n, . . A_{2n}}**, where Ai is an attribute, T* is the anonym zed version of the relation T. We assume that T is a static relational table. Her on this relation T, we performed cell level access control over it. Cell level access control for relational data is implemented by replacing the unauthorized cell values by NULL values.

7. CONCLUSIONS

Our System is a combination of secrecy protection and accuracy restrained mechanisms. The access control mechanism allows only authorized query predicates on sensitive data. The secrecy protection module anonymizes the data to meet privacy requirements and imprecision constraints on predicates set by the access control mechanism. We formulate this interaction as the problem of k-anonymous Partitioning with Imprecision Bounds (k-PIB). We give hardness results for the k-PIB problem and present heuristics for partitioning the data to the satisfy the privacy constraints and the imprecision bounds. Along with this , we also implemented Cell level access control over incremental data of a relational data.

REFERENCES

[1] E. Bertino and R. Sandhu, “Database Security-Concepts, Approaches, and Challenges,” IEEE Trans. Dependable and Secure
Computing, vol. 2, no. 1, pp. 2-19, Jan.-Mar. 2005.
[2] P. Samarati, “Protecting Respondents’ Identities in Microdata
Release,” IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6,
pp. 1010-1027, Nov. 2001.
[3] B. Fung, K. Wang, R. Chen, and P. Yu, “Privacy-Preserving Data Publishing: A Survey of Recent Developments,” ACM Computing Surveys, vol. 42, no. 4, article 14, 2010.
[4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “L-Diversity: Privacy Beyond k-anonymity,” ACM Trans. Knowledge Discovery from Data, vol. 1, no. 1, article 3, 2007.
[5] K. LeFevre, D. DeWitt, and R. Ramakrishnan, “Workload-Aware Anonymization Techniques for Large-Scale Datasets,” ACMTrans. Database Systems, vol. 33, no. 3, pp. 1-47, 2008.