MultiRelational k-Anonymity [*]

M. Ercan Nergiz — Chris Clifton
Department of Computer Sciences, Purdue University
{mnergiz, clifton}@cs.purdue.edu

A. Erhan Nergiz
Bilkent University
anergiz@ug.bilkent.edu.tr

Abstract

k-Anonymity protects privacy by ensuring that data cannot be linked to a single individual. In a k-anonymous dataset, any identifying information occurs in at least k tuples. Much research has been done to modify a single table dataset to satisfy anonymity constraints. This paper extends the definitions of k-anonymity to multiple relations and shows that previously proposed methodologies either fail to protect privacy, or overly reduce the utility of the data, in a multiple relation setting. A new clustering algorithm is proposed to achieve multirelational anonymity.

1 Introduction

The tension between the value of using personal data for research, and concern over individual privacy, is ever-increasing. Simply removing uniquely identifying information (SSN, name) from data is not sufficient to prevent identification because partially identifying information (age, gender . . .) can still be mapped to individuals by using external knowledge[16]. k-Anonymity[13] is one technique to protect against the linkage and identification of records. In k-anonymous table, each distinct tuple (in the projection over identifying attributes) occurs at least k times. Private tables are k-anonymized by the use of generalizations and suppressions, providing two key properties:

- In the anonymous dataset, an individual can only be linked to a group of at least k private entities.
- Every tuple of the anonymous dataset correctly represents a unique tuple in the private dataset (There is no false or noisy information.)

k-Anonymity does not enforce diversity on the sensitive information of equivalence classes (set of tuples with the same identifying attributes in k-anonymous dataset). This has lead to extended privacy definitions [6, 11].

To achieve k-anonymity in single-table datasets, numerous generalization (replacing data values with more general values) and suppression algorithms have been proposed [14, 7, 8, 9, 4, 10, 3, 5, 12]. These algorithms assume each private entity is stored as one row in a single attribute-value table. When information about a private entity is contained in multiple tables, and not easily represented in a single table, the existing definitions and algorithms are insufficient. Section 2 extends k-anonymity definitions for the multi-Relational setting; Section 3 discusses why multiR anonymity (multirelational k-anonymity) is a new problem that is not solved by previous k-anonymity algorithms. In Section 4, protected entities and associated relations will be abstracted by trees and a modification of a previously proposed clustering algorithm will be presented to provide multiR anonymity on snowflake schemas.

2 MultiR Anonymity

We now define notations and k-anonymity for the multiR setting. Given a table T, $T[c][r]$ refers to the value of column c, row r of T. $T[c]$ is the projection of column c

Definition 1 (MultiR schema) *A set of tables SU and a set of functional dependencies SF corresponds to a multiR schema if SU is a dependency preserving, lossless join decomposition with respect to SF and there exists one person specific table $PT \in SU$ where each row corresponds to an individual in population U. We say a database with such a schema has the transcript $MR(SF, U, PT, ST, vip)$, where vip is the unique identifier in PT and $ST = SU - \{PT\}$.*

Table 1 shows an example for a multiR database with transcript $MR(SF, U, T_p, \{T_1, T_2\}, Sid)$ where SF={Sid → GPA, SCid → {Sid, Course, Grade} } and U is the set of students. The schema is in BCNF and dependency preserving. The following quasi-identifier definition is a reformulation of the definition in [15].

Definition 2 (Quasi-identifier) *Let $MR(SF, U, PT, \{T_1, T_2, ...T_n\}, vip)$ be a multiR database, and $JT = PT \bowtie T_1 \bowtie \cdots \bowtie T_n$. Let $f_c : U \rightarrow JT$ and $f_g : JT \rightarrow U'$, where $U \subseteq U'$. A quasi-identifier of MR, written Q_{MR}, is a subset of attributes of*

[*]This material is based upon work supported by the National Science Foundation under Grant No. 0428168.

Table 1. T_p:Student has GPA; T_1:Student takes courses; T_2:Books bought by student for course

Sid	GPA
S1	3.72
S2	2.34
S3	3.12
S4	4.00

SCid	Sid	Course	Grade
SC1	S1	Math	93
SC2	S1	Physics	91
SC3	S1	History	85
SC4	S2	CS	78
SC5	S2	Physics	62
SC6	S2	Religion	42
SC7	S3	History	85
SC8	S3	Religion	75
SC9	S3	Physics	77
SC10	S4	History	98
SC11	S4	Religion	96

SCid	Book	Price
SC1	Discrete	\$63
SC2	Calculus	\$89
SC2	Dynamics	\$42
SC3	Relg. H.	\$33
SC4	Discrete	\$65
SC5	Dynamics	\$51
SC6	Yodaism	\$38
SC7	Ottomans	\$49
SC8	Yodaism	\$39
SC9	Calculus	\$84
SC10	Am. Hist	\$54

Table 2. One anonymization of Table 1 where $k = 2$

Sid	GPA
S1	3.72
S2	2.34
S3	3.12
S4	4.00

SCid	Sid	Course	Grade
SC1	S1	*Science*	93
SC2	S1	Physics	91
SC3	S1	*Social*	85
SC4	S2	*Science*	78
SC5	S2	Physics	62
SC6	S2	*Social*	42
SC7	S3	History	85
SC8	S3	Religion	77
*	*	*	*
SC10	S4	History	98
SC11	S4	Religion	96

SCid	Book	Price
SC1	Discrete	\$63
SC2	Dynamics	\$42
*	*	*
SC3	*Relg Book*	\$33
SC4	Discrete	\$65
SC5	Dynamics	\$51
SC6	*Relg Book*	\$38
SC7	*Hist Book*	\$49
*	*	*
SC10	*Hist Book*	\$54

JT where $\exists p_i \in U$ such that $f_g(f_c(p_i)[Q_{MR}]) = p_i$, and an adversary knows the values of Q_{MR} for p_i.

Informally a quasi-identifier for a schema is the set of attributes in JT that can be used to externally link or identify a given tuple in PT. In Table 1, Course and Book attributes can be considered quasi-identifiers since colleagues of a student may know this information about their friend. The attributes GPA, Grade, Price are the sensitive attributes of the private entity Sid. An attacker knows the quasi-identifiers about an entity and tries to discover other (sensitive) information in the data. E.g., in Table 1, we assume the attacker knows that some individual George in U takes the courses History and Religion and uses the text book American History for History course. The attacker wants to discover George's (sensitive) GPA or his grade in History course. If the data is released as it is, even though George's name is hidden, the attacker can easily link George to student S4 and GPA 4.00 or SCid SC10 and grade 98. We also have other join keys in Table 1 like the vip attribute Sid or SCid that are not part of the quasi-identifier set.

To simplify notation, given database $\mathbf{MR_i}$ we will use the notation $\mathbf{vip_i}$ for a private entity in MR_i, $\mathbf{PT_i}$ for the person specific table of MR_i (table where vip_i is the primary key), $\mathbf{ST_i}$ for the set of all tables in MR_i excluding PT_i, $\mathbf{JT_i}$ for the join of all tables in MR_i, $\mathbf{Q_{MR_i}}$ for set of quasi identifier attributes, and $\mathbf{S_{MR_i}}$ for the set of sensitive attributes of MR_i.

Definition 3 (Structurally Equivalent) *Two databases MR_1 and MR_2 have structurally equivalent schemas if and only if $vip_1 = vip_2$, PT_1 has the same set of attributes as PT_2, and there exist bijective mapping between the set of tables ST_1 and ST_2 such that tables mapped have the same set of attributes. Structurally equivalent schemas have the same func. dependencies, population, QI, sensitive and non-QI joining attribute sets.*

Definition 4 (k-anonymity for multiR databases) *Let MR_1 and MR_2 be two multiR databases with the same set of QI set Q_{MR} and set of sensitive attributes S_{MR}. We say MR_2 is a k-anonymization of MR_1 if and only if $\forall v(JT_2)$ (views on JT_2) the following properties hold:*

1. anonymized*: any query of the type $\Pi_{att}(v(JT_2))$ where $att \in S_{MR}$ returns either zero tuples or at least k (not necessarily distinct) tuples,*
2. anonymized w.r.t. individuals*: any query of the type $\Pi_{vip}(v(JT_2))$ returns either zero tuples or at least k distinct tuples, and*
3. correct*: tuples in JT_1 and JT_2 can be ordered such a way that for all possible j, $JT_2[att][j]$ is equal to or some generalization of $JT_1[att][j]$ if $att \in Q_{MR}$ and $JT_2[att][j]$ is equal to $JT_1[att][j]$ if $att \in S_{MR}$*

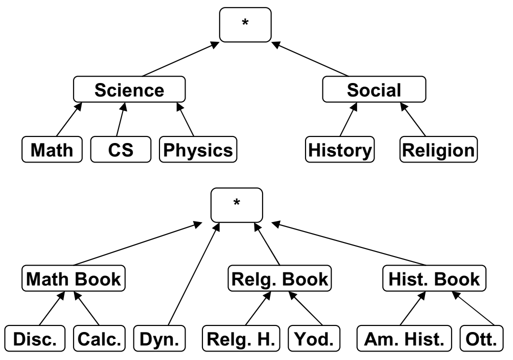

Figure 1. Course, Book DGH structures

The part 'k not necessarily distinct tuples' in requirement 1 can be changed to 'k distinct tuples' if we assume all sensitive information in the MR_1 is unique. MR_1 and the k-anonymous MR_2 need not be structurally equivalent, however, we will see that equivalence eases the anonymization process and can improve utility of the dataset.

The example in Table 1 is clearly not k-anonymous even for $k = 2$, as Π_{Sid} $(\sigma_{Course=``History'' \wedge Book=``Am.Hist''}(JT)) = \{S4\}$. Table 2 shows a 2-anonymization of Table 1 using generalizations from the domain generalization hierarchies given in Figure 1; the same query on Table 2 returns no tuples.

Theorem 1 *Let MR be a k-anonymous multiR database where $ST = \{T_1, T_2, ...T_n\}$ and $k \leq 2$. Then for every vip value vp, there exist some $k-1$ distinct vip values vp_1, vp_2, $\cdots vp_{k-1}$ such that for every view v possible if $vp \in \Pi_{vip}(v(JT))$ then $vp_1, vp_2, \cdots vp_{k-1} \in \Pi_{vip}(v'(JT))$ also. We say the set $S_{vp} = \{vp, vp_1, vp_2, \cdots vp_{k-1}\}$ is the equivalence class of vp and write $EC_{MR}(vp) = S_{vp}$.*

PROOF. Suppose this is not the case and let the set of views $V_{vp} = \{v_i | vp \in \Pi_{vip}(v_i(JT)) \wedge |\Pi_{vip}(v_i(JT))| \geq k\}$. Since there are no common $k-1$ vip values (other than vp) over all views then we have $|\cap_{v_i \in V_{vp}} \Pi_{vip}(v_i(JT))| < k$. Constructing the view $v^{\cap} = \cap_{v_i \in V_{vp}} v_i$ gives $|\Pi_{vip}(v^{\cap}(JT))| \leq k$ and $vp \in \Pi_{vip}(v^{\cap}(JT))$, violating the k-anonymity constraint. This gives a contradiction. □

Theorem 1 can be modified for only sensitive attributes if we have unique sensitive values. Every sensitive value s in the data belongs to a set EC_s of at least k sensitive values such that if s is in a query result then every element in EC_s is also in that query result.

The k-anonymity definition for a multiR database is not arbitrary. If an attacker faces the same set of private entities in every possible set of queries, it can only map its external knowledge to that set. Requirement 3 for k-anonymity prevents false information being included in the anonymization of the original database. (Otherwise there would be trivial solutions for k-anonymization such as replication of tuples. This requirement holds also for classical, single-table k-anonymity, although it was not included explicitly in its definition.) Note that the definitions and concepts given here subsume the definitions of single-table k-anonymity.

3 Single Table Algorithms for MultiR Anonymity

We now explore some obvious approaches to achieving multiR anonymity using single table k-anonymity algorithms. The main idea is to convert the multiR database into one or more single tables and anonymize these. For each approach, we describe why it does not give satisfactory results; the insights are useful in understanding the algorithm we will give in Section 4.

One solution would be to construct the universal relation from the multiR database and anonymize this relation. The problem is that a private entity may become multiple rows in the universal relation, which will likely anonymize with each other, making the relation "k-anonymous" but failing to protect individual identity. In Table 1, the join of T_p and T_1 will already be 2-anonymous w.r.t. QI attributes when we anonymize the entry CS with entry Math to create two entries of Science. But if an attacker knows that Chris is taking History, Math and Physics, then it will map Chris to $S1$ since $S1$ is the only one taking Physics, History and a Science course. Eliminating the join keys would help, but will damage the relational structure we want to preserve. If we instead blindly apply anonymizations to each single dataset we have the same shortcomings. E.g., applying local anonymization on T_p and T_1 will create semanticly the same output datasets as the above example.

Some multiR databases can be converted to a boolean vector "bitmap" format with every private entity as a single row, and distinct attributes used to reflect different values. Table 3 shows the bitmap version of the MR database given in Table 1 and its 2-anonymization. Classical k-anonymity algorithms can be run on such datasets. The anonymized data will then satisfy both multiR anonymity requirements for certain types of relations, however:

- Schemas containing tables that map one entity to another entity an arbitrary number times cannot be converted to bitmap format without information loss. (E.g., a student taking n different Physic classes where n is arbitrarily large cannot be readily expressed.)
- Anonymization would only be through suppression, as generalizing "S1 is taking a Math course and S2 is taking a CS course" into "S1 and S2 are both taking a science course" would correspond to merging columns in the schema rather than generalization of data.
- Conversion to bitmap format produces datasets of high dimensionality. The difficulty of anonymizing a high

dimension table without significant amount of information loss is discussed in [1].

Additional shortcomings of bitmap anonymization include lack of flexibility for certain heuristics, functional dependency inference attacks, and for some databases, insufficient sensitive information protection; further discussion is omitted due to space limitations.

4 Clustering-based MultiR Anonymity

We now develop a multiR anonymity algorithm that overcomes the shortcomings of the approaches described in the previous section, although it places certain (reasonable) restrictions on the schemas supported. Algorithms for arbitrary schemas are left as future work. We first give key properties about the database that the algorithm is expected to preserve, then detail the assumptions about the schema.

Schema Preservation: The schemas of the input database MR and the k-anonymous output MR^* will be structurally equivalent (Definition 3).

Dependency Preservation: The anonymized database preserves the atomicity of join keys and functional dependencies of the original database, so that:

1. the semantics of the data are better preserved, and
2. inference attacks, by an adversary who *knows* a functional dependency that fails to hold in the anonymized data, are prevented.

We require that the schema be normalized to enforce dependencies; this obviates the need to provide dependencies separately as input to the anonymization algorithm.

Snowflake Schema: The algorithm we present is limited to schemas satisfying the following constraints:

1. No connection keys (primary/foreign keys) between tables in MR are quasi-identifiers.
2. Every table in ST contains only one foreign key. Table PT does not contain a foreign key.
3. We say a table T_2 belongs to the family of T_1 and write $T_2 \in F(T_1)$ if T_2 has a foreign key attribute which is a primary key attribute either in T_1 or in another family member of T_1. We restrict ourselves to schemas with $F(PT) = ST$.

Schemas with these constraints are similar to snowflake relations where the fact table is the table PT (see Figure 2), although we do support one to many relationships between PT and other tables. Any table in the schema can contain sensitive attributes, anonymity constraint 1 will hold for all of them. This family of schemas is expressive enough for many database applications (XML, spatio-temporal databases, data warehouses, ...)

We now present a MiRaCle anonymization algorithm

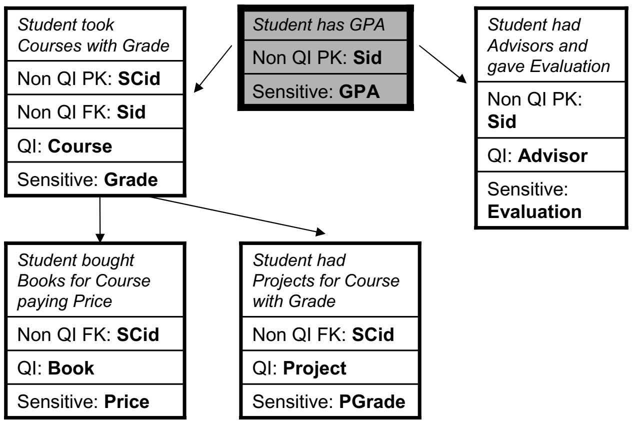

Figure 2. Schema graph

that anonymizes a given multiR database under the assumptions given in the previous section. MiRaCle is a clustering-based anonymity algorithm; any distance-based clustering k-anonymity algorithm [5, 12, 2] can be used as a basic skeleton for MiRaCle anonymizations. Due to space constraints, we only sketch such modification.

The main observation is that all clustering based anonymity algorithms make use of two basic operations on private entities: anonymization and calculation of the distance between two entities. The latter can be generally defined as the cost of the anonymization of two entities. The assumptions given in the previous section enables us to abstract private entities of multiR databases as trees where each level of a given entity tree corresponds to levels of the nested relation for a particular vip entity (Figure 3 gives an example.) The challenge is to anonymize two trees of similar structure with respect to each other.

Algorithm 1 anonymize($tree(s_1)$, $tree(s_2)$)

```
Require: For a tree node $s$; $tree(s_1)$ returns the tree rooted from
    $s$ and $v_s$ returns the QI attribute values associated with node
    $s$. For two values of the same domain $v_1$ and $v_2$, $gen(v_1, v_2)$
    returns the lowest cost generalization of $v_1$ and $v_2$ w.r.t. dgh
    structure defined over the associated domain
1: let $C_1$ be the set of child nodes of node $S_1$
2: let $C_2$ be the set of child nodes of node $S_2$
3: find a low cost pairing of nodes in $S_1$ and $S_2$
4: for all pairs of nodes $(c_1 \in C_1, c_2 \in C_2)$ matched do
5:     $v_{c_1}, v_{c_2} = gen(v_{c_1}, v_{c_2})$
6:     anonymize($tree(c_1)$, $tree(c_2)$)
7: for all nodes $(c \in C_1 \cup C_2)$ unmatched do
8:     suppress every value in nodes of $tree(c)$
```

Algorithm 1 shows how to anonymize two entity trees. Anonymization occurs top-down. Each tree root has a set of child nodes. (In Figure 3, children of $S1$ and $S2$: $C_1 = \{Math, Physics, History\}$, $C_2 = \{CS, Physics, Religion\}$.) The algorithm chooses pairings of nodes between these sets to minimize the local cost

Table 3. Bitmap version of MR without some of the sensitive attributes and its 2-anonymization, attribute T in each course shows whether the student has taken that course or not.

Sid	Math		Physics			CS		History				Religion		GPA
	T	Di	T	Ca	Dyn	T	Di	T	RH	Ot	AH	T	Yo	
S1	1	1	1	1	1	0	0	1	1	0	0	0	0	3.72
S2	0	0	1	0	1	1	1	0	0	0	0	1	1	2.34
S3	0	0	1	1	0	0	0	1	0	1	0	1	1	3.12
S4	0	0	0	0	0	0	0	1	0	0	1	1	0	4.00

S1	*	*	1	*	1	*	*	*	*	0	0	*	*	3.72
S2	*	*	1	*	1	*	*	*	*	0	0	*	*	2.34
S3	0	0	*	*	0	0	0	1	0	*	*	1	*	3.12
S4	0	0	*	*	0	0	0	1	0	*	*	1	*	4.00

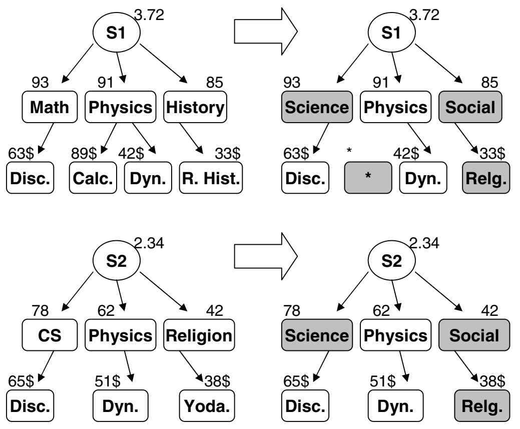

Figure 3. Anonymization of students S1 and S2 from the example MR database in Table 1

in the current level or the overall cost of the anonymized trees. (In Figure 3, Math is paired with CS, Physics with Physics, and History with Religion, producing the set nodes {Science, Physics, Social} which are the least costly sets in terms of cost metrics such as LM.) Since each pair are two trees to be anonymized, values of the roots are anonymized and function is called on the subtrees. (In Figure 3, Math and CS values are changed to Science and a second call is made on $(tree(Science_1), tree(Science_2))$. Unpaired nodes are suppressed (e.g., node Calc.)

References

[1] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *VLDB '05: Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 901–909.

[2] G. Agrawal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering," in *PODS '06: Proc. of the 25th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, Chicago, IL, USA, June 26-28 2006, pp. 153–162.

[3] K. W. B. Fung and P. Yu, "Top-down specialization for information and privacy preservation," in *Proc. of the 21st Int'l Conf. on Data Engineering*, 2005.

[4] R. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. of the 21st Int'l Conf. on Data Engineering*, 2005.

[5] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous k-anonymity through microaggregation," *Data Min. Knowl. Discov.*, vol. 11, no. 2, pp. 195–212, 2005.

[6] A. O. hrn and L. Ohno-Machado, "Using boolean reasoning to anonymize databases," *Artificial Intelligence in Medicine*, vol. 15, no. 3, pp. 235–254, Mar. 1999. http://dx.doi.org/10.1016/S0933-3657(98)00056-6

[7] A. Hundepool and L. Willenborg, "μ and t-argus: software for statistical disclosure control," in *Third International Seminar on Statistical Confidentiality*, 1996.

[8] V. Iyengar, "Transforming data to satisfy privacy constraints," in *Proc., the Eigth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2002, pp. 279–288.

[9] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proc. of the 2005 ACM SIGMOD Int'l Conf. on Management of Data*, Baltimore, MD, June 13-16 2005.

[10] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Multidimensional k-anonymity," University of Wisconsin, Madison, Tech. Rep. 1521, June 2005. http://www.cs.wisc.edu/techreports/2005/TR1521.pdf

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *Proc. of the 22nd IEEE Int'l Conf. on Data Engineering (ICDE 2006)*, Atlanta Georgia, Apr. 2006.

[12] M. E. Nergiz and C. Clifton, "Thoughts on k-anonymization," in *ICDEW '06: Proc. of the 22nd Int'l Conf. on Data Engineering Workshops*. Atlanta, GA, USA: IEEE Computer Society, 2006, p. 96.

[13] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[14] L. Sweeney, "Guaranteeing anonymity when sharing medical data, the datafly system," in *Proc., Journal of the American Medical Informatics Association*. Hanley & Belfus, Inc., 1997.

[15] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, 2002.

[16] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.