42nd South Eastern Symposium on System Theory
University of Texas at Tyler
Tyler, TX, USA, March 7-9, 2010

Database Security Issues in Rough Relational Databases

Theresa Beaubouef
Computer Science Department
Southeastern Louisiana University
Hammond LA

Frederick E. Petry
Naval Research Laboratory
Stennis Space Center MS

Abstract— **In this paper we consider security issues that arise in imprecise databases based on rough set theory. The aspect of security considered is similar to that in statistical databases for which a combination of queries cannot reveal exact values of attributes. Information theory measures are used to characterize security for imprecise databases.**

Keywords - rough sets; database security; information theory; entropy; rough relational database

I. INTRODUCTION

Databases continue to grow in size and complexity, and they are used in many diverse applications. For many real world applications, it is necessary to incorporate some type of uncertainty management into the underlying data model. One characteristic of many imprecise databases is that they allow sets of values in their tuples. This is referred to as a non-first form or nested database [1,2]. If the value of an attribute is non-atomic, i.e. set-valued, then there is uncertainty as to which one of the values in the set corresponds to the attribute, or whether more than one do. There are specific aspects in different uncertain database models but all share use of set values. Of particular interest in this research is the rough relational database, a model based on rough sets [3].

It is also the case that security is becoming more and more of an issue with database applications [4], especially considering the widespread problems associated with identity theft and fraud, website visit history trackers, privacy and data mining applications, and the plethora of SPAM. In this paper we investigate the area of security for rough databases, which have security issues similar to that of statistical databases. We are not talking about the general protection of the data from unauthorized use, but in controlling the type of data that may be accessed by a valid user. For example, a user must be prevented from deducing a specific non-key attribute value associated with the key value.

There will always be some tradeoff between the benefits of information sharing and that of privacy, and while we often want to maximize the sharing and use of data, we can not allow protected data to be compromised. In this paper we discuss security issues in rough relational databases and measurements for determining relative security of rough relations.

II. BACKGROUND

A. Rough Set Theory

Rough set theory [5] is a mathematical formalism for representing uncertainty. An approximation region in rough sets partitions some universe into equivalence classes. This partitioning can be adjusted to increase or decrease its granularity, to group items together that are considered indiscernible for a given purpose, or to "bin" ordered domains into range groups.

U is the *universe*, which cannot be empty,

R : *indiscernibility relation*, or equivalence relation,

$A = (U,R)$, an ordered pair, called an *approximation space*,

$[x]_R$ denotes the equivalence class of R containing x, for any element x of U,

elementary sets in A - the equivalence classes of R,

definable set in A - any finite union of elementary sets in A.

Any finite union of these elementary sets is called a definable set. A *rough set* $X \subseteq U$, however, is defined in terms of the definable sets by specifying its lower ($\underline{R}X$) and upper ($\overline{R}X$) approximation regions:

$\underline{R}X = \{x \in U \mid [x]_R \subseteq X\}$ and $\overline{R}X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$.

$\underline{R}X$ is the positive region, $U - \overline{R}X$ is the negative region, and $\overline{R}X - \underline{R}X$ is the boundary or borderline region of the rough set X, allowing for the distinction between certain and possible inclusion in a rough set.

For example: Let U = {medium, small, little, tiny, big, large, huge, enormous}, and let the equivalence relation R be defined as follows:

R* = {[medium], [small, little, tiny], [big, large], [huge, enormous]}.

A given set X = {medium, small, little, tiny, big, huge}, can be defined in terms of its lower and upper approximations:

$\underline{R}X$ = { medium, small, little, tiny }, and $\overline{R}X$ = { medium, small, little, tiny, big, large, huge, enormous }.

The major rough set concepts of interest are the use of an indiscernibility relation to partition domains into equivalence classes and the concept of lower and upper approximation regions to allow the distinction between certain and possible, or partial, inclusion in a rough set. The indiscernibility relation allows the grouping of items based on some definition of 'equivalence' as it relates to the application domain. Those equivalence classes included in their entirety in X belong to the lower approximation region. The upper approximation region includes those equivalence classes that are included either entirely or partially in X. The results in the lower approximation region are certain, corresponding to exact matches. The boundary region of the upper approximation contains results that are possible, but not certain.

B. Rough Relational Databases

The rough relational database model [3] is an extension of the standard relational database model of Codd [6]. It captures all the essential features of rough sets theory including indiscernibility of elements denoted by equivalence classes and lower and upper approximation regions for defining sets which are indefinable in terms of the indiscernibility.

Every attribute domain is partitioned by some equivalence relation designated by the database designer or user. Within each domain, those values that are considered indiscernible belong to an equivalence class. This is compatible with the traditional relational model since every value belongs to its own class. This information is used by the query mechanism to retrieve information based on equivalence with the class to which the value belongs rather than equality, resulting in less critical wording of queries as shown in [3].

Recall is also improved in the rough relational database because rough relations provide *possible* matches to the query in addition to the *certain* matches which are obtained in the standard relational database. This is accomplished by using set containment in addition to equality of attributes in the calculation of lower and upper approximation regions of the query result.

The rough relational database has several features in common with the ordinary relational database. Both models represent data as a collection of *relations* containing *tuples*. These relations are sets. The tuples of a relation are its elements, and like elements of sets in general, are unordered and nonduplicated. A tuple $\mathbf{t_i}$ takes the form $(d_{i1}, d_{i2}, ..., d_{im})$, where d_{ij} is a *domain value* of a particular *domain set* D_j. In the ordinary relational database, $d_{ij} \in D_j$. In the rough database, however, as in other non-first normal form extensions to the relational model [7, 8], $d_{ij} \subseteq D_j$, and although it is not required that d_{ij} be a singleton, $d_{ij} \neq \varnothing$. Let $P(D_i)$ denote the powerset(D_i) - $\varnothing$.

Definition. A *rough relation R* is a subset of the set cross product $P(D_1) \times P(D_2) \times \cdots \times P(D_m)$.

A rough tuple **t** is any member of *R*, which implies that it is also a member of $P(D_1) \times P(D_2) \times \cdots \times P(D_m)$. If $\mathbf{t_i}$ is some arbitrary tuple, then $\mathbf{t_i} = (d_{i1}, d_{i2}, ..., d_{im})$ where $d_{ij} \subseteq D_j$. A tuple in this model differs from that of ordinary databases in that the tuple components may be sets of domain values rather than single values. The set braces are omitted from singletons for notational simplicity.

Definition. An *interpretation* $\alpha = (a_1, a_2, ..., a_m)$ of a rough tuple $t_i = (d_{i1}, d_{i2}, ..., d_{im})$ is any value assignment such that $a_j \in d_{ij}$ for all j.

The interpretation space is the cross product $D_1 \times D_2 \times \cdots \times D_m$, but is limited for a given relation *R* to the set of those tuples which are valid according to the underlying semantics of *R*. In an ordinary relational database, because domain values are atomic, there is only one possible interpretation for each tuple $\mathbf{t_i}$, the tuple itself. In the rough relational database, this is not always the case when there are a set of values.

Let $[d_{xy}]$ denote the equivalence class to which d_{xy} belongs. When d_{xy} is a set of values, the equivalence class is formed by taking the union of equivalence classes of members of the set; if $d_{xy} = \{c_1, c_2, ..., c_n\}$, then $[d_{xy}] = [c_1] \cup [c_2] \cup ... \cup [c_n]$.

Definition. Tuples $\mathbf{t_i} = (d_{i1}, d_{i2}, ..., d_{im})$ and $\mathbf{t_k} = (d_{k1}, d_{k2}, ..., d_{km})$ are *redundant* if $[d_{ij}] = [d_{kj}]$ for all j = 1,..., m.

In the rough relational database, redundant tuples are removed in the merging process since duplicates are not allowed in sets, the structure upon which the relational model is based.

There are two basic types of relational operators. The first type arises from the fact that relations are considered sets of tuples. Therefore, operations which can be applied to sets also apply to relations. The most useful of these for database purposes are *set difference*, *union*, and *intersection*. Operators which do not come from set theory, but which are useful for retrieval of relational data are *select*, *project*, and *join*. In the rough relational database, relations are rough sets as opposed to ordinary sets. Therefore, new rough operators (—, $\cup$, $\cap$, σ, π, $\bowtie$), comparable to the standard relational operators, were developed for the rough relational

database. Properties of the rough relational operators can be found in [5].

III. Security in Rough Databases

There are many advantages to database technology such as the ability to share data and information and to allow controlled access to data for the purpose of data mining. However, with these advantages also come disadvantages. In particular, there are security issues. Security is usually defined as the protection of data against unauthorized access [6]. However, we must also protect the data from users who are authorized to access the data by controlling what can be accessed and how.

Several researchers have studied issues related to this type of database security [9-15]. In [16] security in a fuzzy database is addressed. The rough relational database, like the fuzzy database, allows for non-first normal form tuples, so much of the analysis for fuzzy set database also applies for the rough relational database.

Each tuple in a rough database can potentially represent a large number of interpretations since the interpretation is an element of the cross product of the sets of domain values. This inherent security occurs because of the ability of the rough relational database to allow sets of values for attributes. When data are merged into these sets, the specific association of values based on the interpretations become blurred. Therefore if some data item $b \in D_i$ is protected, then the value of $x \in D_j$ that is associated with b cannot be determined. So with b and x it should not be possible to derive singleton sets.

One area of database security deals with the overlap of query results allowing an inference to be made. It might be possible to manipulate the data in a relation to result in explicit associations for protected values. For example, if Smith's salary is to be protected, a security violation occurs if the tuples

(… {Jones, Smith, Green}…{55000, 72000} …)

and

(.. {Smith, White, Blum}…{44000, 65000, 72000} …)

are intersected, resulting in

{…Smith …72000…).

In a single query this type of security violation is caused by set intersection resulting in a single tuple. In reality it is difficult to deal with this problem completely since queries may be made at different times or by different users, and each in itself might not be a security problem, but if taken together, might violate some privacy of data. However, in a rough relational database, the intersection of tuples in a single relation cannot produce a security violation. Because redundant tuples are not allowed in a rough relation, there cannot be two tuples having the same interpretation. In [16], this was proved for fuzzy databases. The proof for rough relational database follows similarly:

THEOREM: *The intersection of tuples in a single rough relation R cannot lead to a security violation.*

PROOF: Consider the intersection of tuples t_1, t_2, t_3, …in R over domains D_i and D_j . For a security violation to occur it must be true that

$|d_{1i} \cap d_{2i} \cap d_{3i} \cap \ \ldots| = 1$ and $|d_{1j} \cap d_{2j} \cap d_{3j} \cap \ \ldots| = 1$.

Here the resulting sets are singletons, {b} and {x}, for example. This means that $b \in d_{ki}$ and $x \in d_{kj}$ for all the tuples in the intersection. The interpretation associating b and x must be an interpretation of all the tuples intersected. However, a rough relation cannot have more than one tuple having the same interpretation. Hence, the intersection of tuples in single rough relation cannot produce a security violation. ■

Security violations in a rough relational database in terms of the access protected data items relates directly to uncertainty about specific associations of data items. Information-theoretic measures [17] have often been used to "measure" uncertainty, and they have been used in statistical databases [18], and for fuzzy databases [19]. In the rough relational database information-theoretic measures for uncertainty were defined for rough schemas and rough relations [20]:

Definition. The *rough schema entropy* for a rough relation schema S is $E_s(S) = - \Sigma_j [\Sigma \ Q_i \log(P_i)]$ for i = 1,... n; j = 1,..., m

where there are n equivalence classes of domain j, and m attributes in the schema $R(A_1, A_2, ...,A_m)$.

Definition. The *rough relation entropy* of a particular extension of a schema is $E_R(R) = -\Sigma_j D\rho_j(R) [\Sigma \ DQ_i \log(DP_i)]$ for i = 1,... n; j = 1,..., m

where $D\rho_j(R)$ represents a type of database roughness for the rough set of values of the domain for attribute j of the relation, m is the number of attributes in the database relation, and n is the number of equivalence classes for a given domain for the database.

The schema entropy provides a measure of the uncertainty inherent in the definition of the rough relation schema taking into account the partitioning of the domains on which the attributes of the schema are defined. The entropy of an actual rough relation instance $E_R(R)$ of some database D is an extension of the schema entropy obtained by multiplying each term in the product by the roughness of the rough set of values for the domain of that given attribute.

We obtain the $D\rho_j(R)$ values by letting the non-singleton domain values represent elements of the boundary region, computing the original rough set

accuracy and subtracting it from one to obtain the roughness. DQ_i is the probability of a tuple in the database relation having a value from class i, and DP_i is the probability of a value for class i occurring in the database relation out of all the values which are given.

Consider the sample database below where domains for soil color and size have been defined as

COLOR = {[black, ebony], [brown, tan, sienna],[white], [gray], [orange]}, and

PARTICLE-SIZE = {[big, large], [huge, enormous], [medium], [small, little, tiny]}.

TABLE I. SAMPLE-114

BIN	COLOR	PARTICLE - SIZE
P21	brown	medium
P22	{black, tan}	large
P23	gray	{medium, small}
T01	black	tiny
T04	{gray, brown}	large

TABLE II. SAMPLE-115

BIN	COLOR	PARTICLE-SIZE
M43	{black, tan, white}	{big, huge, medium}
M46	{brown, orange, white, gray}	{medium, small}

The rough relation entropy of the relations SAMPLE-114 and SAMPLE-115 shown in the tables are calculated as follows:

E_R(SAMPLE-114) = -(4/7)[(2/5)log(2/7) + (3/5)log(3/7) + 0 + (2/5)log(2/7) + 0] - (2/6)[(2/5log(2/6) + 0 + (2/5)log(2/6) + (2/5)log(2/6)] = **.56**

E_R(SAMPLE-115) = -(7/7)[(1/2)log(1/7) + (2/2)log(2/7) + (2/2)log(2/7) + (1/2)log(1/7) + (1/2)log(1/7)] – (5/5)
[(1/2)log(1/5) + (1/2)log(1/5) + (2/2)log(2/5) + (1/2)log(1.5) = **3.7821**

From this example it is clear that our concept of security in the rough relational database corresponds to uncertainty in this sense, so we can use these measures of entropy as a quantitative measure for security in a rough relational database.

IV. CONCLUSION

Security is an important problem in databases, and applications involving statistical databases and data mining are especially critical in preserving the security of protected data. Aspects related to this type of database security also apply to rough relational databases.

We have shown how the nature of the rough relational database provides some inherent security through its use of non-first normal form structure. Moreover, we provided measures of database security based on information-theoretic measures that allow for the evaluation of numeric measures for entropy. We are currently investigating the extension of this work for fuzzy rough and intuitionistic relational databases [21].

ACKNOWLEDGMENT

The authors would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

REFERENCES

[1] A. Makinouchi "A Consideration on normal form of not-necessarily normalized relation in the relational data model". Proc. of the 3rd Int. Conf.VLDB,1977, pp 447-453.

[2] A. Yazici, A. Soysal, B. Buckles and F. Petry, "Uncertainty in a Nested Relational Database Model", Data and Knowledge Engineering, 30, #3, pp275-301, 1999.

[3] T.Beaubouef, F. Petry and B. Buckles "Extension of the Relational Database and its Algebra with Rough Set Techniques". Computational Intelligence. 11,1995, 233-245.

[4] W. Stallings and. L, Brown. Computer Security: Principles and Practice, Prentiss Hall, 2007.

[5] Z. Pawlak,. Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Norwell, MA 1991

[6] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 5th ed., Addison Wesley, 2006

[7] M. Roth, H. Korth, and D. Batory. "SQL/NF: A query language for non-1NF databases." .Information Systems. 12, 99-114 1987

[8] S. Thomas and P. Fischer, "Nested Relational Structures,"Advances in Computing Research , 3, 269-307, JAI Press, Greenwich CT, 1989

[9] D. Denning,. "Secure Statistical Databases with Random Sample Queries," ACM Transactions on Database Systems, 5:3, September, 1980, pp. 291-315.

[10] Chin, F. and Ozsoyoglu, G., "Statistical Database Design," ACM Transactions on Database Systems, 6:1, March 1981, pp. 113-139.

[11] E. Leiss, "Randomizing: A Practical Method for Protecting Statistical Databases Against Compromise", Proc. Very Large Databases Conference, Mexico, 1982, pp. 189-196.

[12] H. Wong, "Micro and Macro Statistical/Scientific Database Management," Int. Conf. on Data Engineering, 1984, pp. 104-106

[13] M. McLeish,. "Further Results on the Security of Partitioned Dynamic Statistical Database," ACM Transactions on Database Systems, 14:1, March 1989, pp. 98-113.

[14] A. Motro., D. Marks, and S. Jajodia, "Aggregation in Relational databases: Controlled Disclosure of Sensitive Information," in Proceedings of ESORICS 94, Third European Symposium on Research in Computer Security, Brighton, UK, November 1994. Lecture Notes in Computer Science No. 875, Springer-Verlag, pp. 431–445.

[15] C. Clifton and D. Marks. "Security and Privacy Implications of Data Mining," Proc. ACM SIGMOD Workshop on Data Mining and Knowledge Discovery, Canada, 1996, pp. 15-19

[16] B. Buckles and F. Petry, "Security and Fuzzy Databases," Proceedings of 1982 IEEE International Conference on Cybernetics and Society, pp 622-625, Seattle WA 1982

[17] C. Shannon "A Mathematical Theory of Communication." Bell System Tech Jour 27, pp 379-423 (July), pp. 623-656 (Oct), 1948.

[18] M. Thomason "On Applications of Probabilistic Information Theory to Relational Databases," SPIE Tech. Symposium, Huntsville, AL, May, 1979, pp78-82.

[19] B. Buckles and F. Petry, "Information-Theoretic Characterization of Fuzzy Databases," IEEE Trans. Systems, Man, and Cybernetics, 13 # 1 pp. 74-77, 1983

[20] T. Beaubouef, F. Petry and G. Arora "Information-Theoretic Measures of Uncertainty for Rough Sets and Rough Relational Databases". Information Sciences, 109, 1998, 185-195

[21] T. Beaubouef and F. Petry, "Uncertainty Modeling for Database Design Using Intuitionistic and Rough Set Theory," Journal of Intelligent and Fuzzy Systems, Vol. 20, No. 3, 2009, pp. 105-117.