Efficient Method of Detecting Data Leakage Using Misusability Weight Measure

[1]K.Sundaramoorthy, [2],Dr.S.Srinivasa Rao Madhane
[1,]Research Scholar, St.Peter's University,Chennai ,Tamilnadu, India,
[2,]Adhiparasakthi college of Engineering, Kalavai Tamilnadu, India

Abstract

Users within the organization's perimeter perform various actions on this data and may be exposed to sensitive information embodied within the data they access. In an effort to determine the extent of damage to an organization that a user can cause using the information she has obtained, we introduce the concept of misuseability Weight. To calculate the M-Score, A misuseability weight measure, this calculates a score that represents the sensitivity level of the data exposed to the user and by that predicts the ability of the user to maliciously exploit the data. By assigning a score that represents the sensitivity level of the data that a user is exposed to, the misuseability weight can determine the extent of damage to the organization if the data is misused. Using this information, the organization can then take appropriate steps to prevent or minimize the damage.

***Index Terms**: Data leakage, data misuse, security measures, misuseability weight.*

I. INTRODUCTION

Sensitive information such as customer or patient data and business secrets constitute the main assets of an organization. Such information is essential for the organization's employees, subcontractors, or partners to perform their tasks. Conversely, limiting access to the information in the interests of preserving secrecy might damage their ability to implement the actions that can best serve the organization. Thus, data leakage and data misuse detection mechanisms are essential in identifying malicious insiders.The focus of this paper is on mitigating leakage or misuse incidents of data stored in databases (i.e., tabulardata) by an insider having legitimate privileges to access the data. There have been numerous attempts to deal with the malicious insider scenario. The methods that have been devised are generally based on user behavioral profiles that define normal user behavior and issue an alert whenever a user's behavior significantly deviates from the normal profile. The most common approach for representing user behavioral profiles is by analyzing the SQL statement submitted by an application server to the database (as a result of user requests), and extracting various features from these SQL statements. Another approach focuses on analyzing the actual data exposed to the user, i.e., theresult-sets. However, none of the proposed methods consider the different sensitivity levels of the data to which an insider is exposed. This factor has a great impact in estimating the damage that can be caused to an organization when data is leaked or misused. Security-related data measures including k-Anonymity, l-Diversity, and (_, k)-Anonymity is mainly used for privacy-preserving and is not relevant when the user has free access to the data. Therefore, we present a new concept, Misuseability Weight, which assigns a sensitivity score to data sets, thereby estimating the level of harm that might be inflicted upon the organization when the data is leaked. Four optional usages of the misuseability weight are proposed.

[1] Applying anomaly detection by learning the normal behavior of an insider in terms of the sensitivity level of the data she is usually exposed to.
[2] Improving the process of handling leakage incidents identified by other misuse detection systems by enabling the security officer to focus on incidents involving more sensitive data.
[3] Implementing a Dynamic Misuseability-Based Access Control (DMBAC), designed to regulate user access to sensitive data stored in relational databases;.
[4] Reducing the misuseability of the data.

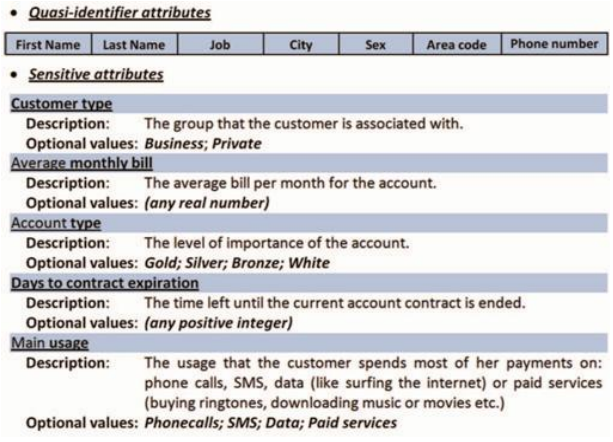
II. MISUSEABILITY WEIGHT CONCEPTS

Data stored in an organization's computers is extremely important and embodies the core of the organization's power. An organization undoubtedly wants to preserve and retain this power. On the other hand, this data is necessary for daily work processes. Users within the organization's perimeter (e.g., employees, subcontractors, or partners) perform various actions on this data (e.g., query, report, and search) and may be exposed to sensitive information embodied within the data they access.In an effort to determine the extent of damage to an organization that a user can cause using the information she has obtained, we introduce the concept of Misuseability Weight. By assigning a score that represents the sensitivity level of the data that a user is exposed to, the misuseability weight can determine the extent of damage to the organization if the data is misused. Using this information, the organization can then take appropriate steps to prevent or minimize the damage.

2.1 Dimensions of Misuseability

Assigning a misuseability weight to a given data set is strongly related to the way the data is presented (e.g., tabular data, structured or free text) and is domain specific. Therefore, one measure of misuseability weight cannot fit all types of data in every domain. In this section, we describe four general dimensions of misuseability. These dimensions, which may have different levels of importance for various domains, can serve as guidelines when defining a misuseability weight measure. While the first two dimensions are related to entities (e.g., customers, patients, or projects) that appear in the data, the last two dimensions are related to the information (or properties) that are exposed about these entities. The four dimensions are: Number of entities. This is the data size with respect to the different entities that appear in the data. Having dataabout more entities obviously increase the potential damage as a result of a misuse of this data. Anonymity level. While the number of different entities in the data can increase the misuseability weight, the anonymity level of the data can decrease it. The anonymity level is regarded as the effort that is required in order to fully identify a specific entity in the data. Number of properties. Data can include a variety of details, or properties, on each entity (e.g., employee salary or patient disease). Since each additional property can increase the damage as a result of a misuse, the number of different properties (i.e., amount of information on each entity) should affect the misuseability weight. Values of properties. The property value of an entity can greatly affect the misuseability level of the data. For example, a patient record with disease property equals to HIV should probably be more sensitive than a record concerning patient with a simple flu. In the context of these four dimensions, we claim that PPDP measures are only effective in a limited way through their capability of measuring the anonymity level dimension of the data. These measures, however, lack any reference to the other important dimensions that are necessary for weighting misuseability. For example, consider a table that shows employee names and salaries. Even if we double all the salaries that appear in the table, there may not be any change in neither of these measures' scores, and therefore no reference to the values of properties dimension. As a result of this lack, as well as others, we conclude that PPDP measures are not sufficiently expressive to serve as a misuseability weight measure and that a new measure is needed. In the following section, we introduce our proposal for addressing this need.

Fig. 1. An example of quasi-identifier and sensitive attributes.



- ***Quasi-identifier attributes***

First Name	Last Name	Job	City	Sex	Area code	Phone number

- ***Sensitive attributes***

Customer type
Description: The group that the customer is associated with.
Optional values: *Business; Private*

Average monthly bill
Description: The average bill per month for the account.
Optional values: *(any real number)*

Account type
Description: The level of importance of the account.
Optional values: *Gold; Silver; Bronze; White*

Days to contract expiration
Description: The time left until the current account contract is ended.
Optional values: *(any positive integer)*

Main usage
Description: The usage that the customer spends most of her payments on: phone calls, SMS, data (like surfing the internet) or paid services (buying ringtones, downloading music or movies etc.)
Optional values: *Phonecalls; SMS; Data; Paid services*

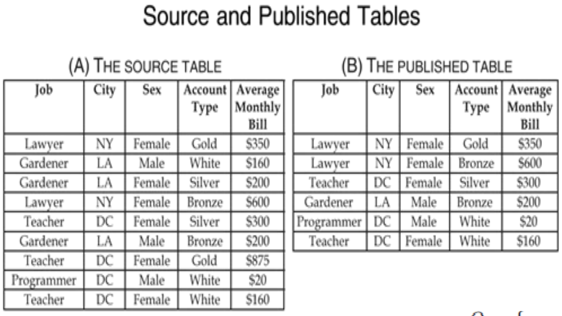
III. THE M-SCORE MEASURE

To measure the misuseability weight, we propose a new algorithm—the M-score. This algorithm considers and measures different aspects related to the misuseability of the data in order to indicate the true level of damage that can result if an organization's data falls into wrong hands. The M-score measure is tailored for tabular data sets (e.g., result sets of relational database queries) and cannot be applied to no tabular data such as intellectual property, business plans, etc. It is a domain independent measure that assigns a score, which represents the misuseability weight of each table exposed to the user, by using a sensitivity score function acquired from the domain expert.

3.1 Formal Definition

In this section, we provide the formal definitions for the M-score. Without loss of generality, we assume that only a single database exists. Nevertheless, the measure can be easily extended to cope with multiple databases. The first definition discusses the building blocks of our measure— table and attributes.

Definition 1 (Table and Attribute). A table T(A_1;...,A_n) is a set of r records. Each record is a tuple of n values. The value i of a record, is a value from a closed set of values defined by A_i, the i's Attribute of T. Therefore, we can define A_i either as the name of the column i of T, or as a domain of values. We define three, nonintersecting types of attributes: quasi-identifier attributes; sensitive attributes; and other attributes, which are of no importance to our discussion. To exemplify the computation of the M-score, we use throughout this paper the database structure of a cellular company as represented in Fig. 1.



Source and Published Tables

(A) THE SOURCE TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Gardener	LA	Male	White	$160
Gardener	LA	Female	Silver	$200
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Teacher	DC	Female	Gold	$875
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

(B) THE PUBLISHED TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

Definition 2 (Quasi-Identifier Attributes). Quasi-identifier attributes are $Q = \{q_{i1}, \ldots, q_{ik}\} \subseteq \{A_i, \ldots, A_n\}$| attributes that can be linked, possibly using an external data source, to reveal a specific entity that the specific information is about. In addition, any subset of the quasi-identifiers (consisting of one or more attributes of Q) is a quasi-identifier itself. $q_1 = First\ Name; q_2 = Last\ Name; q_3 = Job; q_4 = City;$ $q_5 = Sex; q_6 = Area\ Code; and\ q_7 = Phone\ Number.$

Definition 3 (Sensitive Attributes). Sensitive attributes

$S_j = \{s_{j1}, \ldots, s_{jk}\} \subseteq \{A_i, \ldots, A_n\}$ are attributes that are used to evaluate the risk derived from exposing the data. The sensitive attributes are mutually excluded from the quasi-identifier attributes (i.e., $\forall j S_j \cap Q = \emptyset$). .In our example; we have five different sensitive attributes—$s_1 = Customer\ Group$ to $s_5 = Main\ Usage.$.

Definition 4 (Sensitivity Score Function). The sensitivity score function f: [0,1] assigns a sensitivity score to each possible value x of S_j, according to the specific context c 2 C in which the table was exposed. For each record r, we denote the value x_r of S_j as $S_j[x_r]$. The sensitivity score function should be defined by the data owner (e.g., the organization) and it reflects the data owner's perception of the data's importance in different contexts. When defining this function, the data owner might take into consideration factors such as privacy and legislation, and assign a higher score to information that eventually can harm others (for example, customer data that can be used for identity theft and might result in compensatory costs). In addition, the data owner should define the exact context attributes.

3.2 Calculating the M- Score

The M-score incorporates three main factors.

[1] Quality of data—the importance of the information.

[2] Quantity of data—how much information is exposed.

[3] The Distinguishing Factor (DF)—given the quasiidentifiers, the amount of efforts required in order to discover the specific entities that the table refers to. In order to demonstrate the process of calculating the M-score, we use the example presented in Table 1. Table 1 a represents our source table (i.e., our "database") while Table 1b is a published table that was selected from the source table and for which we calculate the M- score. In the following sections, we explain each step in the proposed measure calculation.

3.3 Calculating Raw Record Score

The calculation of the raw record score of record i (or RRS_i), is based on the sensitive attributes of the table, their value in this record, and the table context. This score determines the quality factor of the final M-score, using the sensitivity score function f, defined in Definition 4.

Definition 5 (Raw Record Score).

$$RRS_i = \min\left(1, \sum_{S_j \in T} f(c, S_j[x_i])\right).$$

S_j2T For a record i, RRS_i will be the sum of all the sensitive values score in that record, with a maximum of 1. When comparing two tables with different number of attributes, the table with the larger number of sensitive attributes will tend to have a higher sensitivity value for each individual record. In order to be able to compare the sensitivity of tables having different number of attributes, we need to eliminate this factor. Therefore, we have set an upper bound on the RRS_i by taking the minimum between 1 and the sum of sensitivity scores of the sensitive attributes.

3.4 Calculating Record Distinguishing Factor

Using the distinguishing factor, the M-score incorporates the uniqueness of the quasi-identifier's value in the table when weighting its misuseability. The DF measures to what extent a quasi-identifier reveals the specific entity it represents (e.g., a customer). It assigns a score in the range of [0,1], when the lower the score is, the harder it is to distinguish one entity from another, given this quasi-identifier. In other words, the DF of record indicates the effort a user will have to invest in order to find the exact entity she is looking for.

Usually, the DF is not easily acquired, and therefore we use the record distinguishing factor (D_i) as an approximation. The record distinguishing factor (D_i) is a kanonymity-like measure, with a different reference table from which to calculate k. While k-anonymity calculates, for each quasi-identifier, how many identical values are in the published table, the distinguishing factor's reference is "Yellow Pages." This means that an unknown data source, denoted by R_0, contains the same quasi-identifier attributes that exist in the organization's source table, denoted by R_1 (for example, Table 1a). In addition, the quasi-identifier values of R_1 are a subset of the quasi-identifier values in R0, or more formally—quasi-identifierR1 quasi-identifier R0.

In the example presented in Table 1b, the distinguishing factor of the first record is equal to two (i.e., $D_1 = 2$) since the tuple {Lawyer, NY, Female} appears twice in Table 1a. Similarly, D_3 =3, ({Teacher, DC, Female} appears three times in Table 1a); $D_4 = 2$; and D_5 =1. If there are no quasi-identifier attributes in the published table, we define that for each record i, D_i equals to the published table size.

As previously mentioned, the k-anonymity may suffer from the common sensitive attribute problem in which an adversary may not be able to match a record with its true entity, but she can still know the sensitive values. We opt to use the variation of the k-anonymity measure since it is well known and widely used in various tasks and implementations. However, other PPDP measures such as l-Diversity and (, k)-Anonymity can be used as well.

3.5 Calculating the Final Record Score (RS)

The Final Record Score uses the records' RSS_i and D_i, in order to assign a final score to all records in the table. Definition 6 (Final Record Score). Given a table with r records, RS is calculated as follows:

$$RS = \max_{0 \le i \le r}(RS_i) = \max_{0 \le i \le r}\left(\frac{RRS_i}{D_i}\right).$$

For each record i, RS calculate the weighted sensitivity score RS_i by dividing the Record's Sensitivity Score (RRS_i) by its distinguishing factor (D_i). This ensures that as the record's distinguishing factor increases (i.e., it is harder to identify the record in the reference table) the weighted sensitivity score decreases. The RS of the table is the maximal weighted sensitivity score.

3.6 Calculating the M-Score

Finally, the M-score measure of a table combines the sensitivity level of the records defined by RS and the quantity factor (the number of records in the published table, denoted by r). In the final step of calculating the M-score, we use a settable parameter x(x >1). This parameter sets the importance of the quantity factor within the table's final M-score. The higher we set x, the lower the effect of the quantity factor on the final M-score.

Definition 7 (M-Score). Given a table with r records, the table's M-score is calculated as follow:

$$MScore = r^{1/x} \times RS = r^{1/x} \times \max_{0 \le i \le r}\left(\frac{RRS_i}{D_i}\right),$$

Where r is the number of records in the table, x is a given parameter and RS is the final Record Score presented in Definition 6.The derived M-score value is not bounded. Thus, it is difficult to understand the meaning of the derived value and in particular the level of threat that is reflected by the M- score value. Therefore, we propose the following procedure for normalizing the M-score to the range [0,1]. Assume that T is thepublishedtablewhichisderivedbyapplyingtheselection operator on the source table S, given a set of conditions, and then the projection operator: $T = \Pi_{a1,a2,\ldots,an}(\sigma_{\text{condition}}(S)).$ Let T be the projection of a_1, a2,...,an.on the source table: $T^* = \Pi_{a1,a2,\ldots,an}(S).$

The M-score of table T can be normalized by dividing the M-score of T by the M-score of T :
$NormM\text{-}Score(T) = M\text{-}Score(T)/M\text{-}Score(T^{*})$.

IV. EXTENDING THE M-SCORE

Until now, we were describing how the M-score can measure the misuseability weight of a single publication, without considering the information the user already has; i.e., "prior knowledge." Prior knowledge can be: 1) previous publications (previous data tables the user was already exposed to); and 2) knowledge on the definition of the publication (e.g., the user can see the WHERE clause of the SQL query). In this section, we extend the M-score basic definition and address these issues.

4.1 Multiple Publications

A malicious insider can gain valuable information from accumulated publications by executing a series of requests. The result of each request possibly revealing information about new entities, or enriching the details of entities already known to her. Here, we focus on the case where the user can uniquely identify each entity (e.g., customer) in the result-set, i.e., the distinguishing factor is equal to 1 (D_i =1) Fig. 3 depicts nine optional cases resulting from two fully identifiable sequential publications. Each Case is determined by the relation (equal, overlapping, or distinct) between the two publications with respect to the publications' sensitive attributes (marked in shades of green) and the exposed entities which are the distinct identifier values (marked in red). For example, in Case 1 on Fig. 3, the publications share the same schema (i.e., include the same attributes in all tuples), but have no common entities; Case 6 presents two publications that share some of the entities, but each publication holds different attributes on them.

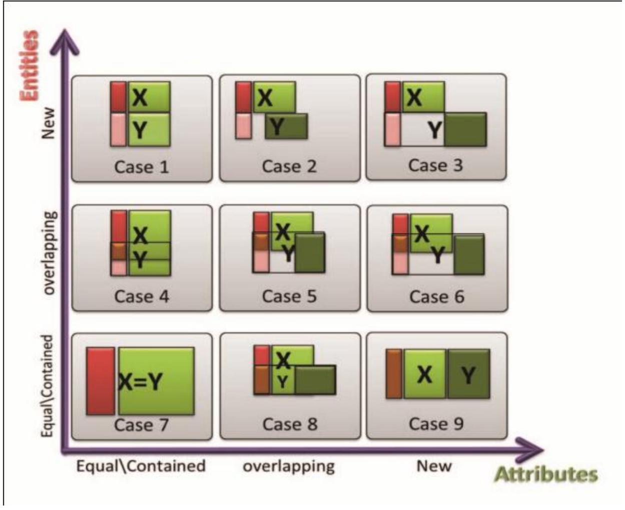


Fig. 3. Nine cases resulting from two fully identifiable publications.

Based on these nine possible cases, we introduce the Construct Publication Ensemble procedure. The Construct Publication Ensemble procedure is recursive. For each new publication, the procedure first creates an ensemble set X of all the previous publications that are within the time frame F. Then, the procedure checks which case in Fig. 3 fits the current publications and acts according. Finally, on the resulting ensemble set is returned

4.2 Multirelational Schema

In this section, we address the scenario of multirelational schema in which more than one table is released. In particular, following Nergiz et al. [18], we assume that we are given a multirelational schema that consists of a set of tables T_1;...T_n, and one main table PT, where each tuple corresponds to a single entity (for example in Table 1 the main entity is the customer). The joined table JT is defined as $JT = PT \bowtie T_1 \bowtie \cdots \bowtie T_n$. Note that the quasi-identifier set can span across various tables, namely the "quasi-identifier set for a schema is the set of attributes in JT that can be used to externally link or identify a given tuple in PT". The various ingredients of the M-score can be calculated on an individually basis by using JT. For each entity i in PT we calculate the RRS_i by summing the scores of all sensitive values that appear in all her records in JT after eliminating duplicate values (for example if there are two records in JT that correspond to the same customer and each one of these records redundantly indicate that the customer is living in NY, then the sensitive score for the city NY will be counted only once). The D_i for an entity should be calculated by first calculating the D_i for each record in J. Then, the entity's D_i is set to the minimum among all her records' D_i in JT. Finally, the M-score is calculated.

4.3 Knowledge on Request Definition

A user may have additional knowledge on the data she receives emanating from knowing the structure of the request created this data, such as the request's constraints. In such cases, the basic M-score does not consider such knowledge. For example, a user might submit the following request:
"Select "Name" of customers with "Account type" ¼ "Gold". In this case, the user knows that all customers are "gold" customers. However, since the result-set of this request will only include the names, the M-score cannot correctly compute its misuseability weight. In order to extend the M-score to consider this type of prior knowledge, RES(R) and COND(R) operators are defined.

V. CONCLUSIONS AND FUTURE WORK

We introduced a new concept of misuseability weight and discussed the importance of measuring the sensitivity level of the data that an insider is exposed to. We defined four dimensions that a misuseability weight measure must consider. To the best of our knowledge and based on the literature survey we conducted, there is no previously proposed method for estimating the potential harm that might be caused by leaked or misused data while considering important dimensions of the nature of the exposed data. Consequently, a new misuseability measure, the M-score, was proposed. We extended the M-score basic definition to consider prior knowledge the user might have and presented four applications using the extended definition. Finally, we explored different approaches for efficiently acquiring the knowledge required for computing the M-score, and showed that the M-score is both feasible and can fulfill its main goals.

Two important issues, which relate to the knowledge elicitation and representation, should be further investigated: the temporal aspect of the M-score and the validity of the knowledge, acquired from the experts, over time; and the knowledge acquisition that might be subjective and not consistent among different experts which, in turn, may lead to an inaccurate sensitivity function.

In regards to the time factor, we assumed that the sensitivity level of an attribute's value will change in rare cases and especially the order of the values with respect to their sensitivity level. However, we are aware of the need to validate and reacquire the knowledge from timeto-time, and although we showed in the experiments that the knowledge can be acquired accurately with relatively minimal effort (in terms of experts time) using the pairwise comparison approach, we plan to explore methods for incremental learning, or postlearning fine tuning of the elicited sensitivity score function in future work.

With respect to the subjectivity of the elicited scoring function, our experiments indicate that the methods used ensure that the acquired knowledge is not biased. In fact, we showed that using knowledge acquired from one expert is sufficient in order to calculate sound M-scores for the entire domain. We plan to further investigate this important issue and check the effect of combining knowledge from several experts (e.g., ensemble of knowledge models) on the quality of the acquired knowledge and the accuracy of the M- score. In addition, in some cases the value of customers can be calculated by using known knowledge on the customer (e.g., how much she spends) and by predicting future revenue from the customer. In such cases, the sensitivity level of sensitive attributes can be objectively obtained by using machine learning techniques; in particular by fitting the sensitive parameter values to the customer value.

REFERENCES

[1] 2010 Cyber Security Watch Survey, http://www.cert.org/ archive/pdf/ecrimesummary10.pdf, 2012.

[2] A. Kamra, E. Terzi, and E. Bertino, "Detecting Anomalous Access Patterns in Relational Databases," Int'l J. Very Large Databases, vol. 17, no. 5, pp. 1063-1077, 2008.

[3] S. Mathew, M. Petropoulos, H.Q. Ngo, and S. Upadhyaya, "DataCentric Approach to Insider Attack Detection in Database Systems," Proc. 13th Conf. Recent Advances in Intrusion Detection, 2010.

[4] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge Based Systems, vol. 10, no. 5 , pp. 571-588, 2002.

[5] A. Machanavajjhala et al., "L-Diversity: Privacy beyond K-Anonymity," ACM Trans. Knowledge Discovery from Data, vol. 1, no.1, article 1, 2007.

[6] R.C. Wong, L. Jiuyong, A.W. Fu, and W. Ke, "(, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2006.

[7] E. Celikel et al., "A Risk Management Approach to RBAC," Risk and Decision Analysis, vol. 1, no. 2, pp. 21-33, 2009.

[8] B. Carminati, E. Ferrari, J. Cao, and K. Lee Tan, "A Framework to Enforce Access Control over Data Streams," ACM Trans. Information Systems Security, vol. 13, no. 3, pp. 1-31, 2010.

[9] Q. Yaseen and B. Panda, "Knowledge Acquisition and Insider Threat Prediction in Relational Database Systems," Proc. Int'l Conf. Computational Science and Eng., pp. 450-455, 2009.

[10] G.B. Magklaras and S.M. Furnell, "Insider Threat Prediction Tool: Evaluating the Probability of IT Misuse," Computers and Security, vol. 21, no. 1, pp. 62-73, 2002.

[11] M. Bishop and C. Gates, "Defining the Insider Threat," Proc. Ann. Workshop Cyber Security and Information Intelligence Research, pp. 13, 2008.

[12] C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey on Recent Developments," ACM Computing Surveys, vol. 42, no. 4, pp. 1-53, 2010.

[13] A. Friedman and A. Schuster, "Data Mining with Differential Privacy," Proc. 16th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 493-502, 2010.

[14] C. Dwork, "Differential Privacy: A Survey of Results," Proc. Fifth Int'l Conf. Theory and Applications of Models of Computation, pp. 1-19, 2008.

[15] T. Dalenius, "Finding a Needle in a Haystack or Identifying Anonymous Census Records," J. Official Statistics, vol. 2, no. 3, pp. 329-336, 1986.

[16] B. Berendt, O. Gu¨nther, and S. Spiekermann, "Privacy in e-Commerce: Stated Preferences vs. Actual Behavior," Comm. ACM, vol. 48, no. 4, pp. 101-106, 2005.

[17] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum, "Privacy and Contextual Integrity: Framework and Applications," Proc. IEEE Symp. Security and Privacy, pp. 184-198, 2006.

[18] M.E. Nergiz et al., "Multirelational k-Anonymity," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 8, pp. 1104-1117, Aug. 2009.