Data Leakage Quantification

Sokratis Vavilis, Milan Petković, Nicola Zannone

Data Leakage Quantification*

Sokratis Vavilis[1], Milan Petković[1,2], and Nicola Zannone[1]

[1] Eindhoven University of Technology, The Netherlands
{s.vavilis, m.petkovic, n.zannone}@tue.nl
[2] Philips Research Europe, High Tech Campus, The Netherlands
milan.petkovic@philips.com

Abstract. The detection and handling of data leakages is becoming a critical issue for organizations. To this end, data leakage solutions are usually employed by organizations to monitor network traffic and the use of portable storage devices. These solutions often produce a large number of alerts, whose analysis is time-consuming and costly for organizations. To effectively handle leakage incidents, organizations should be able to focus on the most severe incidents. Therefore, alerts need to be prioritized with respect to their severity. This work presents a novel approach for the quantification of data leakages based on their severity. The approach quantifies leakages with respect to the amount and sensitivity of the leaked information as well as the ability to identify the data subjects of the leaked information. To specify and reason on data sensitivity in an application domain, we propose a data model representing the knowledge in the domain. We validate our approach by analyzing data leakages within a healthcare environment.

Keywords: Data Leakage Detection, Severity Metrics, Data Sensitivity Model

1 Introduction

In the recent years the number of data breaches reported by public and private organizations has increased sharply. For instance, a study from Ponemon Institute in 2012 showed that 94% of US hospitals suffered serious data breaches [1]. The main cause is that IT systems often implement inadequate measures that allow users to have access on sensitive data, which they are not authorized to access. The problem is that it may not be always possible to specify fine-grained access control policies to protect from the disclosure of data. For example, access control policies in hospitals often do not pose restrictions on the amount of health records that doctors can access. Moreover, access to information should not be restricted under certain circumstances. For instance, doctors should be able to access patient records to face an emergency. Typically, this is addressed using the break-the-glass protocol [2], which allows users to bypass security mechanisms, thus leading to potential data misuse.

Timely detection and management of data leakages is becoming a serious challenge for organizations. According to the newly proposed EU data protection regulation, organizations are obliged to notify privacy authorities within 24 hours after the detection of a data breach [3]. To detect data leakages, organizations usually deploy data leakage

* This work has been funded by the Dutch national program COMMIT under the THeCS project.

detection (DLD) solutions. These solutions analyze the disclosed data and raise an alert when a leakage is detected. However, the number of alerts can be huge in certain situations, making difficult their analysis and management. For example, in hospitals a DLD solution might produce a large number of alerts due to the usage of the break-the-glass protocol. Before taking any action (e.g., notifying authorities), organizations typically evaluate a sample of the alerts manually. To effectively manage and mitigate the damage due to security incidents, organizations should be able to focus on the most severe incidents. To this end, data leakages should be quantified based on their severity.

Data leakage quantification, however, has not been properly addressed in the literature. Many proposals [4–6] are founded on quantitative information flow. In particular, they quantify data leakages in terms of the number of "sensitive" bits which have been disclosed. Thereby, they do not consider the semantics of the leaked information in the assessment of data leakages. From our knowledge, only M-Score [7] assesses the severity of data leakages on the basis of the semantics of the leaked information. In particular, M-Score uses the amount and sensitivity of leaked information as well as an identifiability factor to measure the severity of leakages. The amount and sensitivity of leaked information characterize the "quantity" and "quality" aspects of the leakage. These aspects are weighted with respect to the identifiability factor which represents the ability to obtain the identity of the individuals to whom the leaked data refer. However, M-Score requires defining the sensitivity for all pieces of information explicitly. Such a task is time-consuming and error-prone. In addition, M-Score is not able to accurately distinguish data leakages (see Section 7.2).

In this work we propose a novel approach to quantify data leakages on the basis of the content of the leaked information. In particular, we make the following contributions: (i) a new metric that evaluates the severity of data leakages based on the amount and sensitivity of the leaked data and an identifiability factor; (ii) a data model representing the knowledge of an application domain to specify and reason on the sensitivity of the information in the domain. Our metric uses the same factors used by M-Score. However, compared to M-Score, our metric provides a more accurate discrimination of data leakages with respect to their severity. In addition, differently from M-Score, our approach does not require specifying the sensitivity for every piece of information characterizing the application domain explicitly. The data model makes it possible to infer the sensitivity of every piece of information through a sensitivity propagation mechanism based on a small initial sensitivity assignment.

We validate our approach by analyzing a sample scenario in the healthcare domain. Healthcare is indeed an interesting domain to investigate as a large amount of sensitive data, such as patient healthcare records, has to be protected. Based on a given scenario, a group of security experts was asked to evaluate the severity of a number of data leakages. The severity measurements calculated using our metric have been analyzed against the evaluation provided by the security experts.

The remainder of the paper is organized as follows. The next section discusses related work. Section 3 motivates the need of approaches for data leakage quantification using a running example in the healthcare domain. Section 4 presents an overview of our approach. Section 5 defines the data model along with the machinery to reason on data sensitivity, and Section 6 describes how leaked information is mapped to the data model. Section 7 presents our metric for data leakage quantification along with a comparison with M-Score. A validation of the proposed metric is presented in Section 8. Finally, Section 9 concludes the paper by providing directions for future work.

2 Related Work

Several works aiming at data leakage detection and protection can be found in the literature [8–10]. Data leakage detection (DLD) solutions differ in the approach and technologies used to detect leakages. They are usually rule-based [10], behavior-based [11] or content-based [12–14]. Rule-based approaches set predefined policies (e.g., access control policies, firewall rules) that are used to define which operations are allowed or not. In behavior-based DLD solutions the permitted usage of data is defined by observing users' behavior. For instance, network behavior monitoring technologies, such as anomaly detection and extrusion detection systems [11], can be used to detect unusual behavior. Content-based DLD solutions analyze the values of the disclosed data to detect data leakages. Such approaches include the use of keywords, regular expressions, text classification [14], and information retrieval [12, 13] to detect the presence of sensitive data leaving the organization perimeter. However, most existing DLD solutions only focus on detecting leakages and do not assess their severity.

A number of proposals for the quantification of data breaches exist in the literature [15–17]. These proposals measure the impact of a security incident in financial terms. For instance, security incidents are quantified on the basis of the damage on the reputation of the organization and the losses on the revenue. Another approach for measuring the severity of security incidents is proposed in [2, 18]; this approach evaluates privacy infringements by quantifying deviations from the intended usage of data.

Data leakage quantification is studied in the field of quantitative information flow [4, 6]. These solutions measure the amount of information leaking from a high confidentiality input to a low confidentiality output. Leakages are usually quantified in terms of bits, using metric based on information theory and information entropy. Quantitative information flow has also been applied to quantify leakages at network level [5]. In particular, it has been used to measure the amount of leaked information (measured in bytes) in the hypertext transfer protocol. The major drawback of quantitative information flow methods is that they do not consider the semantics of the leaked information to quantify data leakages. In particular, the sensitivity of leaked data is not considered in the calculation of the severity of a leakage.

To the best of our knowledge, M-Score [7] is the only proposal that uses semantic information to compute the severity of data leakages. In particular, M-Score measures the severity of leakages in database environment on the basis of the amount and sensitivity of the data leaked. However, M-Score is not able to accurately distinguish data leakages. A detailed analysis of M-Score is presented in Section 7.1.

3 Running Example

Consider a local hospital where patients of a small region are treated. The hospital offers treatment for various diseases, ranging from flu to serious cases such as heart attack and infectious diseases. Patient information is stored in a central database at the hospital in the form of electronic health records (EHR). Typically, doctors and nurses can only access EHR of the patients they treat. However, in emergency situations doctors and nurses can bypass access control mechanisms by invoking the break-the-glass protocol. Therefore, they can have access to the EHRs of all patients. The hospital has also administrative personnel for financial management and to make appointments with patients. Moreover, the database is maintained by a database administrator.

To detect data leakages, the hospital employs a DLD solution. In a typical day hospital employees access thousands of patient records. In addition, the number of invocations of the break-the-glass protocol can be huge [2]. Therefore, the DLD system can generate hundreds of alerts, making difficult to evaluate their severity. In particular, the evaluation of a large number of data leakages can be time-consuming for organizations. Below we present three representative alerts of data leakages:

Alert 1 A query is made by a doctor requesting an unusual large number of patient records. In particular, the names and addresses of 10000 patients were retrieved.

Alert 2 A query for patient data is made by a doctor after his regular working hours. He retrieved 200 records containing the names and diseases of patients.

Alert 3 A query for data about patients affected by HIV is made by a medical researcher of the hospital, specialized on cardiovascular diseases. He retrieved 500 anonymized records containing the sex, age and treatment provided to patients.

To assist organizations in the evaluation of data leakages, leakages should be ranked on the basis of their severity. However, the quantification of data leakages is not a trivial task as the leakages may differ on several aspects. The *amount* of leaked information is a main aspect to discriminate data leakages. For instance, the leakage described in Alert 1 contains thousands of patient records, while in Alert 2 only a relatively small amount (200) of records is retrieved. Another difference is the information leaked itself. In particular, the *sensitivity* of the information (i.e., the impact that its disclosure has on the patient) can be different. For instance, disease information (Alert 2) is more sensitive than patient addresses (Alert 1). Finally, data leakages also differ on the extent that an individual related to the data is identifiable. According to the EU Data Protection Directive (95/46/EC), personal data should be protected. However, the principles defined in the directive do not apply to anonymous data. Therefore, the ability to identify the individuals related to the leaked data has an impact on the severity of a leakage. For instance, in Alerts 1 and 2 the leaked data can be directly linked to patients' identity, while in Alert 3 the data are anonymized. Therefore, the first two alerts should be considered more severe than the third alert.

In order to obtain a ranking of alerts we need a method to quantify the severity of data leakages. Such quantification should take into account the amount and sensitivity of leaked information, and the extent to which the identity of the individuals related to the leaked information can be ascertained.

4 Approach

DLD solutions are often deployed to detect data leakages. These solutions analyze the data leaving the system and raise an alert when a data leakage is detected. However, the

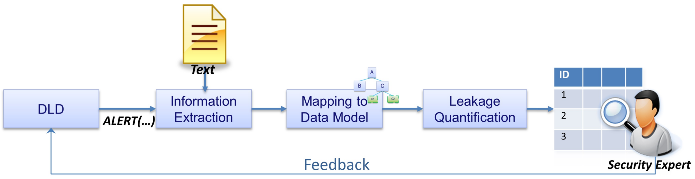

Fig. 1: Data Leakage Quantification Process

number of alerts can be very large, making their analysis costly and time-consuming for organizations. To enable organizations to focus on the most severe incidents, data leakages have to be ranked based on their severity.

To address this issue, we propose a new data leakage quantification system (Fig. 1). The system is connected to a DLD solution. In particular, it receives alerts of data leakages and analyzes the disclosed data to estimate their severity. Since leakages can originate from different sources, data can be structured or unstructured. For instance, data originating from a database are structured, while data in an e-mail are usually unstructured. In this work, we focus on structured data where portions of the database's tables are leaving the database as result of a user query. However, the system can be extended using technologies like natural language processing and information retrieval, to extract information from unstructured data.

The severity of data leakages depends both on the amount of the data leaked and on the data themselves. Therefore, the quantification of data leakages should consider both these factors. In particular, data leakage quantification should reflect the cost of data disclosure according to the data subject/owner or to the organization hosting the data. We represent such a cost in terms of the sensitivity of data. In particular, a sensitivity value should be assigned to every piece of data that may be leaked.

Assigning a sensitivity value to all pieces of data, however, is time-consuming and error prone. We employ a data model representing the knowledge of the application domain to reason on data sensitivity. The data model makes it possible to specify the sensitivity of some pieces of information and infer the sensitivity for the other pieces of information based on this initial assignment (Section 5). To calculate the severity of data leakages, leaked data are mapped to the data model. Intuitively, the attributes and values in the leaked tables are mapped to the corresponding piece of information in the data model (Section 6). The sensitivity of data along with a discrimination factor, which determines to what extent data can be related to an individual, and the amount of leaked data is used to quantify the severity of data leakages (Section 7).

Data leakages are ranked on the basis of their severity. Security experts thus can evaluate data leakages focusing on the more severe incidents. Based on this analysis, organizations can take the appropriate actions to prevent or mitigate the losses. If the analysis reveals that a leakage is a false positive (i.e., wrongly recognized by the DLD solution as a leakage), feedback explaining the assessment is sent to the DLD system to reduce the number of false alerts in the future.

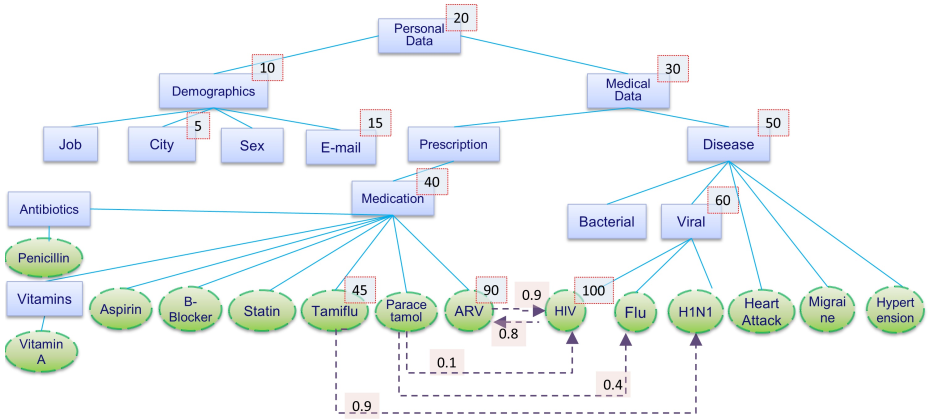

Fig. 2: Data Model Example

5 Modeling and Reasoning on Data Sensitivity

To determine the sensitivity of data we employ a data model. It provides a description of the data within an application domain along with the necessary semantic information.

Definition 1. *A data model is a tuple $DM = (T, I, HR, IR, SL, PL)$, where:*

- *T is a set of data types, and I is a set of data instances.*
- *$HR \subset T \times T \cup I$ is a hierarchy relation representing a specialization relationship.*
- *$IR \subset I \times I$ represents an inference relation on I.*
- *$SL : T \cup I \to \mathbb{Z}_{\geq 0}$ is a labeling function that assigns a sensitivity value to data types and instances.*
- *$PL : I \times I \to [0, 1]$ is a labeling function that defines the probability to infer knowledge about a data instance having knowledge about another data instance.*

Fig. 2 shows an example of data model for the healthcare domain. Data types are nodes represented by rectangles, while data instances by ovals. Hierarchy relations are represented with straight edges between two nodes. For instance, the hierarchy relation between Viral diseases and Flu (denoted as $(Viral, Flu)$) nodes indicates that flu is a viral disease. Inference relations are represented by dashed edges. For example, the inference relation between Anti-RetroViral (ARV) and HIV indicates that a patient treated with ARV medication is likely infected with the HIV virus.

Nodes can be annotated with a sensitivity label that indicates the sensitivity of the data represented by the respective node. For instance, the sensitivity label of node HIV is $SL(HIV) = 100$. Inference relations are annotated with probability labels to indicate the probability of the inference. For instance, label $PL(ARV, HIV) = 0.9$ indicates that a patient treated with ARV medication is very likely (90%) infected by HIV.

Both domain and security experts need to be involved in the construction of the data model for a given domain. Domain experts should define data types and instances along with the hierarchy relations between them. Moreover, they should determine the inference relations between instances with the respective probability labels. On the other hand, security experts should annotate the data model with sensitivity labels.

The annotation of the data model with sensitivity labels, however, can be difficult as the number of nodes can be large. Ideally, security experts should assign sensitivity labels to few nodes and the system determines the sensitivity of the other nodes based on this initial assignment. To this end, we introduce the notion of *sensitivity propagation*.

Definition 2. *Let $DM = (T, I, HR, IR, SL, PL)$ be a data model.* Sensitivity propagation *is a function $SP : T \cup I \rightarrow \mathbb{Z}_{\geq 0}$ such that given a node $x \in T \cup I$*

$$SP(x) = \begin{cases} SL(x) & if\ SL(x)\ exists \\ SP(y) & otherwise \end{cases} \tag{1}$$

with $y \in T$ such that $(y, x) \in HR$.

Sensitivity propagation is used to assign a sensitivity value to the nodes in the data model based on hierarchy relations. Intuitively, if a node does not have a sensitivity label, then its sensitivity is inherited from the node higher in the hierarchy.

Example 1. In Fig. 2 the sensitivity label of Paracetamol is not defined. Therefore, this node inherits the sensitivity value of the parent node (Medication), i.e., $SP(Paracetamol) = SP(Medication) = 40$.

Although sensitivity propagation simplifies the task of assigning sensitivity values to nodes, it may lead to an inaccurate assignment. For instance, a security expert might underestimate the sensitivity of some pieces of information, which is propagated through the data hierarchy. To this end, we use inference relations to validate the propagated values and eventually adjust the sensitivity of the nodes to a higher value.

Definition 3. *Let $DM = (T, I, HR, IR, SL, PL)$ be a data model.* Node sensitivity *is a function $NS : T \cup I \rightarrow \mathbb{Z}_{\geq 0}$ such that given a node $x \in T \cup I$*

$$NS(x) = \max\{SP(x), IS(x)\} \tag{2}$$

where $SP(x)$ is the sensitivity derived through sensitivity propagation (Definition 2) and the inferred sensitivity *$IS(x)$ is computed using function $IS : T \cup I \rightarrow \mathbb{Z}_{\geq 0}$:*

$$IS(x) = \begin{cases} \sum_{(x,y) \in IR} PL(x, y) \times NS(y) & if\ x \in I \\ 0 & if\ x \in T \end{cases} \tag{3}$$

Intuitively, the sensitivity of data types is obtained through sensitivity propagation. On the other hand, the sensitivity of data instances also depends on the sensitivity of the information that can be inferred through the inference relations.

Example 2. Consider the sensitivity value of node Paracetamol calculated in Example 1. The node has an inference relation with nodes HIV and Flu, which have sensitivity 100 and 60 respectively. The inferred sensitivity is 34. As this value is lower than the sensitivity obtained through propagation, the node sensitivity for Paracetamol is 40.

Note that the computation of inferred sensitivity can be problematic as inference relations can form a cyclic graph. This issue can be addressed by representing inference relations as Markov chains. For the lack of space, we omit details of such an approach.

6 Mapping Information on the Data Model

To quantify the severity of data leakages, the leaked data have to be mapped onto the data model in order to determine their sensitivity. Recall that in this work we focus on structured data leaving a database. Thus, the mapping consists in determining, for each entry in the leaked table, the corresponding node in the data model. In this section we first introduce the notation used to represent data; then, we present the mapping.

Let $\mathcal{A}$ be a set of attributes. Attributes can be divided in two types: *quasi-identifiers* and *sensitive* attributes. Quasi-identifiers $Q = \{q_1, \ldots, q_k\} \subseteq \mathcal{A}$ can be used to reveal the identity of an individual, possibly using an external data source (any subset of the quasi-identifiers is a quasi-identifier itself). Sensitive attributes $S = \{s_1, \ldots, s_m\} \subseteq \mathcal{A}$ are the attributes that need to be protected. Certain attributes may belong to both sets. For instance, the sex of a person is a quasi-identifier, as it can be used to partly reveal an individual's identity. Moreover, according to the EU Data Protection Directive (Directive 95/46/EC) the sex of an individual is considered to be sensitive personal information. Note that the distinction between quasi-identifiers and sensitive attributes is related to the purpose and context of use. One may consider all attributes in a table as both quasi-identifiers and sensitive attributes.

A database table $D(a_1, \ldots, a_n)$ is a set of records over a set of attributes $\{a_1, \ldots, a_n\} \subseteq \mathcal{A}$. We denote the records in $D(a_1, \ldots, a_n)$ as $R^{D(a_1,\ldots,a_n)}$. Given a record $r \in R^{D(a_1,\ldots,a_n)}$, $a_i[r]$ represents the value of attribute a_i in r. Attributes take values from a close set of values defined by the domain. Given an attribute $a \in \mathcal{A}$, C_a denotes the domain of a.

Example 3. Consider the scenario in Section 3. The database includes table D(Job, City, Sex, Disease, Medication). Attributes Job and City are quasi-identifiers, while attributes Disease and Medication are sensitive attributes. Attribute Sex belongs to both sets. Each sensitive attribute takes values from a pre-specified domain. For instance, Disease can take a value from {HIV, Heart Attack, Hypertension, Migraine, H1N1, Flu}, and Medication from {ARV, b-Blocker, Tamiflu, Statin, Antibiotics, Aspirin, Paracetamol, Vitamins}. We assume that a doctor can prescribe antibiotics without referring explicitly to a particular medical product, allowing the patient and/or pharmacist to choose an antibiotic from a list of equivalent medication.

In addition to the attributes and values contained in the leaked table, we also consider pre-acquired knowledge as part on the leaked information. In particular, conditional clauses such as WHERE clauses in SQL may leak information. For instance, consider a user query requesting the medication prescribed to patients infected by HIV (i.e., WHERE Disease = 'HIV'). Although the leaked table only contains values concerning attribute Medication, we also assume that value HIV is leaked.

The attributes of a table correspond to data types in the data model presented in Section 5.1. Formally, $\mathcal{A} \subseteq T$. The values of an attribute can correspond either to an instance or a data type node, which is located in the subtree of the data model rooted in the node corresponding to the attribute. Formally, $\{C_a\}_{a \in \mathcal{A}} \subseteq T \cup I$.

To obtain the sensitivity of the leaked data, the values of the attributes in the leaked table need to be mapped onto the data model. For the mapping, search methods can be employed. However, the efficiency of the search methods depends on the size of the data model. To facilitate the search process, the attributes in the leaked table can be first mapped to the corresponding data type node in the data model. The value of the attribute can be then mapped starting the search from the data type node corresponding to the attribute and continuing downward the hierarchy defined by the data model.

Example 4. Consider table D(Job, City, Sex, Disease, Medication) in Example 3 and the data model in Fig. 2. Suppose that a leaked record contains value Hypertension for attribute Disease. First, Disease is mapped by searching from node Personal Data downward the hierarchy until a data type node with the same name is found. Then, value Hypertension is mapped by searching the corresponding node from node Disease.

7 Data Leakage Quantification

The estimation of the severity of data leakages requires metrics that assess the sensitivity and the amount of the data leaked. In this section, we present an overview of M-Score [7] and study its accuracy by applying it to some data leakages. Based on this analysis, we present our proposal for data leakage quantification.

7.1 M-Score

M-Score has been proposed to estimate data misuse in a database environment. It is based on the calculation of the severity of a (portion of) table, which may have been leaked. M-Score evaluates the severity of a data leakage by evaluating three main aspects of the leaked data: the sensitivity, quantity and distinguishing factor (Section 3). The sensitivity of data is defined through a *sensitivity score* function.

Definition 4. *Let $\mathcal{A}$ be a set of attributes and C_{a_i} the domain of an attribute $a_i \in \mathcal{A}$. The sensitivity score function $f : C_{a_i} \to [0,1]$ assigns a sensitivity value to each value in C_{a_i}.*

Given a record $r \in R^{D(a_1,\ldots,a_n)}$, the sensitivity score of a value $a_i[r] \in C_{a_i}$ is denoted by $f(a_i[r])$. The sensitivity of a record is captured by the *raw record score*. In particular, the calculation of the raw record score of a record r, denoted as RRS_r, encompasses the sensitive attributes of a table and their values in r.

Definition 5. *Let $D(a_1,\ldots,a_n)$ be a table, $S = \{s_l,\ldots,s_m\} \subseteq \mathcal{A}$ the set of sensitive attributes in $D(a_1,\ldots,a_n)$, and f the sensitivity score function. Given a record $r \in R^{D(a_1,\ldots,a_n)}$, the* raw record score *of r is*

$$RRS_r = \min\left(1, \sum_{s_i \in S} f(s_i[r])\right) \tag{4}$$

Intuitively, the raw record score of a record is obtained by summing the sensitivity score of every piece of sensitive information in the record, with a maximum of 1.

The *distinguishing factor* of a record r with respect to a table, denoted as $DF_r^{D(a_1,\ldots,a_n)}$, is the amount of efforts required to identify the individual which r refers to. The distinguishing factor of a record is calculated on the basis of quasi-identifier attributes.

Definition 6. *Let $D(a_1,\ldots,a_n)$ be a table, $Q=\{q_l,\ldots,q_k\}\subseteq\mathcal{A}$ the set of quasi-identifier attributes in $D(a_1,\ldots,a_n)$ and $r\in R^{D(a_1,\ldots,a_n)}$ a record in $D(a_1,\ldots,a_n)$. Given $\{q_1[r],\ldots,q_k[r]\}$ with $q_i[r]\in C_{q_i}$ the set of quasi-identifier values in r, the* distinguishing factor *of r with respect to $D(a_1,\ldots,a_n)$ is*

$$DF_r^{D(a_1,\ldots,a_n)}=\frac{1}{|R'|} \tag{5}$$

where $R'=\{r_i|\forall q_i\in Q\ q_i[r]=q_i[r_i]\}$, i.e. the set of records in $D(a_1,\ldots,a_n)$ that have $\{q_1[r],\ldots,q_k[r]\}$ as quasi-identifier values and $|R'|$ is the number of such records.

The *final record score* for a leaked table $L(a_1,\ldots,a_m)$, denoted as RS_L, is calculated based on the raw record score and distinguishing factor.

Definition 7. *Let $ST(a_1,\ldots,a_n)$ be a source table and $L(b_1,\ldots,b_m)$ a leaked table with $\{b_1,\ldots,b_m\}\subseteq\{a_1,\ldots,a_n\}$. Given the raw record score RRS_r and distinguishing factor $DF_r^{ST(a_1,\ldots,a_n)}$ for every record $r\in R^{L(b_1,\ldots,b_m)}$, the* final record score *of $L(b_1,\ldots,b_m)$ is*

$$RS_L=\max_{r\in R^{L(b_1,\ldots,b_m)}}(RRS_r\times DF_r^{ST(a_1,\ldots,a_n)}) \tag{6}$$

It is worth noting that the distinguishing factor is calculated with respect to the source table. To capture the quantity aspect of the leakage, M-Score determines the severity of leakages based on the final record score and the number of records disclosed.

Definition 8. *Let $L(a_1,\ldots,a_n)$ be a leaked table. Given the final record score RS_L of $L(b_1,\ldots,b_m)$, the* M-Score *of $L(a_1,\ldots,a_n)$ is*

$$M\text{-}Score_L=|R^{L(a_1,\ldots,a_n)}|^{\frac{1}{x}}\times RS_L \tag{7}$$

where $|R^{L(a_1,\ldots,a_n)}|$ represents the number of records in $L(a_1,\ldots,a_n)$ and $x\in\mathbb{Z}_{>0}$ is a weighting factor for the number of records.

7.2 Application of M-Score

In this section we study the accuracy of M-Score by applying it to a number of leakage examples. The examples are based on table D(Job, City, Sex, Disease, Medication) presented in Section 6. In the examples we analyze the calculation of the severity of leakages with respect to the sensitivity of the leaked data. Therefore, we assume that the amount of records and distinguishing factor are the same for all leakages.

The sensitivity score function used to assess the severity of leakages is shown in Table 1. The sensitivity score assigned to diseases is related to the impact the disclosure

Disease		**Medication**	
$f(HIV) = 1$	$f(Migraine) = 0.3$	$f(ARV) = 1$	$f(Antibiotics) = 0.4$
$f(HeartAttack) = 0.7$	$f(Flu) = 0.1$	$f(b\text{-}Blocker) = 0.8$	$f(Aspirin) = 0.3$
$f(Hypertention) = 0.6$		$f(Statin) = 0.6$	$f(Paracetamol) = 0.1$
$f(H1N1) = 0.4$		$f(Tamiflu) = 0.5$	$f(Vitamins) = 0.1$

Table 1: Sensitivity score function

of disease information has on the life of an individual. In particular, diseases whose disclosure has a major impact on the life of the patient (e.g., HIV) are assigned a higher sensitivity than diseases with less critical impact (e.g., Flu). The sensitivity of medication is related to its degree of specialization; medication can be general and specialized. General medication is prescribed to treat mild symptoms of different diseases, such as headache. This category includes medication such as Antibiotics, Aspirin and Paracetamol. Specialized medication is prescribed to treat symptoms related to a particular disease. For instance, ARV is usually prescribed to patients infected with HIV. We assume that specialized medication has higher sensitivity than general medication.

We apply M-Score to two cases. We focus on the impact of data sensitivity on the severity of leakages. Thus, we consider the same number of leaked records and set parameter x of M-Score equal to 1. At the end of this section, we discuss the impact of the amount of records and x on the severity of data leakages.

Case 1: Consider the leakages in Tables 2a and 2b. In *Case* 1.1 the records contain general medication prescribed to patients suffering from serious health issues. In *Case* 1.2 the records contain information about specialized medication prescribed to patients suffering from serious health issues. We expect *Case* 1.2 to be more severe than *Case* 1.1 as it contains more sensitive information. However, M-Score calculates the same severity value (2.000) in both cases. The problem lies in the use of the *min* function in the calculation of *RRS*. In particular, this measure has an upper bound equal to 1, which leads to the same *RRS* for all records whose sensitivity is greater than 1.

Case 2: Consider the leakages in Tables 2c and 2d. In *Case* 2.1 the records contain general medication. In contrast, the records in *Case* 2.2 contain information about specialized medication. In both cases we consider only a small percentage of records (1 record) about patients suffering from a serious health issue. Therefore, *Case* 2.2 should be estimated more severe than *Case* 2.1, as it contain more sensitive information. In contrast, M-Score calculates the same severity value (2.000) in both cases. The problem lies in the use of the *max* function in the calculation of *RS*. In particular, *RS* uses the sensitivity value of the record that has the highest sensitivity. Since parameter x is equal to 1, M-Score is the product of *RS* and the number of rows.

As shown above, M-Score may not be able to accurately estimate the severity of leakages. In particular, M-Score is not able to discriminate data leakages that contain at least one highly sensitive record (i.e., $\sum_{s_i \in S} f(s_i[r]) \geq 1$), regardless of the sensitivity of the other records. This is due to the calculation of *RRS* and *RS* and, in particular, to the use of the *min* and *max* functions respectively. The *min* function allows a maximum sensitivity score of 1 per record; the *max* function leads to consider only the record

Job	City	Sex	Disease	Medication
Lawyer	LA	Male	HIV	Vitamins
Lawyer	LA	Male	Heart Attack	Aspirin
Lawyer	LA	Male	Migraine	Paracetamol
Lawyer	LA	Male	Hypertension	Aspirin
M-Score: 2.000				

(a) Case 1.1

Job	City	Sex	Disease	Medication
Lawyer	LA	Male	HIV	ARV
Lawyer	LA	Male	Hypertension	Statin
Lawyer	LA	Male	Heart Attack	b-Blocker
Lawyer	LA	Male	Migraine	b-Blocker
M-Score: 2.000				

(b) Case 1.2

Job	City	Sex	Disease	Medication
Lawyer	LA	Male	HIV	Vitamins
Lawyer	LA	Male	Flu	Paracetamol
Lawyer	LA	Male	Flu	Aspirin
Lawyer	LA	Male	Migraine	Aspirin
M-Score: 2.000				

(c) Case 2.1

Job	City	Sex	Disease	Medication
Lawyer	LA	Male	HIV	ARV
Lawyer	LA	Male	H1N1	Tamiflu
Lawyer	LA	Male	H1N1	Antibiotics
Lawyer	LA	Male	Flu	Antibiotics
M-Score: 2.000				

(d) Case 2.2

Table 2: M-Score evaluation

with the highest *RS* when calculating M-Score. Thus, the discrimination of the severity of leakages relies on the amount of records leaked. The importance of this factor is expresses by parameter x of M-Score. For low values of x (i.e., $x \approx 1$) considerable importance is given to the amount of records. Thus, leakages with a larger number of records result to have a higher severity. Otherwise, for $x \gg 1$, more importance is given to sensitivity. Therefore, M-Score converges to the value of the record with highest *RS*.

7.3 L-Severity

This section presents L-Severity, a new metric for quantifying data leakages that addresses M-Score's drawbacks. Similarly to M-Score, L-Severity assesses the severity of data leakages based on the sensitivity, distinguishing factor and amount of leaked data.

Definition 9. *Let $ST(a_1,\ldots,a_n)$ be a source table, $L(b_1,\ldots,b_m)$ a leaked table with $\{b_1,\ldots,b_m\} \subseteq \{a_1,\ldots,a_n\}$, $S = \{s_l,\ldots,s_m\} \subseteq \mathcal{A}$ the set of sensitive attributes in $L(b_1,\ldots,b_m)$ and $DM = (T,I,HR,IR,SL,PL)$ a data model. Given $r \in R^{L(b_1,\ldots,b_m)}$ a record in $L(b_1,\ldots,b_m)$ and $DF_r^{ST(a_1,\ldots,a_n)}$, the* record sensitivity *of r is*

$$RSENS_r = DF_r^{ST(a_1,\ldots,a_n)} \times \sum_{s_i \in S} NS(s_i[r]) \tag{8}$$

where NS is the node sensitivity of the node in the data model that corresponds to the value $s_i[r]$ of a sensitive attribute s_i.

In the calculation of record sensitivity we make use of the data model (Section 5). In particular, we use *NS* to calculate the sensitivity of each sensitive attribute value in a record. To calculate the severity of data leakages, we introduce L-Severity metric.

Definition 10. *Let $ST(a_1,\ldots,a_n)$ be a source table and $L(b_1,\ldots,b_m)$ a leaked table with $\{b_1,\ldots,b_m\} \subseteq \{a_1,\ldots,a_n\}$. Given the record sensitivity $RSENS_r$ for each record $r \in R^{L(b_1,\ldots,b_n)}$, the* leakage severity (L-Severity) *of $L(b_1,\ldots,b_m)$ is*

Case	**L-Severity**	**M-Score**
Case 1.1	2.050	2.000
Case 1.2	2.900	2.000
Case 2.1	1.150	2.000
Case 2.2	2.100	2.000

Table 3: Comparison between L-Severity and M-Score

$$L\text{-}Severity_L = \sum_{r \in R^{L(b_1,\dots,b_m)}} RSENS_r \tag{9}$$

To demonstrate L-Severity we applied it to the same cases used to evaluate M-Score (Section 7.2). To make a fair comparison with M-Score we use the sensitivity score function in Table 1 to determine the sensitivity of data. A summary of the severity scores obtained by L-Severity and M-Score is shown in Table 3. Accordingly, the value of L-Severity is 2.050 for *Case* 1.1 and 2.900 for *Case* 1.2. Hence, L-Severity is higher for *Case* 1.2 than for *Case* 1.1. Similarly, the L-Severity value for *Case* 2.2 (2.100) is higher that the value for *Case* 2.1 (1.150). Thus, L-Severity provides values that better characterize the severity of leakages with respect to the intuition (Section 7.2).

8 Validation

In this section we validate L-Severity using a number of representative data leakages in a healthcare environment. First, we discuss the construction of the data model using existing ontologies. Then, we apply L-Severity to assess the severity of the data leakages and validate the results against the evaluation provided by a group of security experts.

8.1 Determining Data Sensitivity

The data model presented in Section 5 provides a description of the data characterizing an application domain along with the necessary semantic information, such as the sensitivity of the data. Ontologies [19] are often adopted to capture the knowledge of a specific domain. The basic elements of ontologies are *Classes, Individuals, Attributes and Relationships*. Classes are abstract groups of objects, while individuals represent instances of classes. Attributes are used to represent properties and characteristics of classes and individuals. Relationships represent ways in which classes and individuals are related to one another. Ontologies can be used as a basis for the definition of a data model. Table 4 shows the correspondence between the elements of the data model and the elements of an ontology.

Several ontologies have been proposed for the healthcare domain [20–25]. In this work we adopted and extended SNOMED-CT [22] as a basis for our data model. In particular, we added an attribute to classes and individuals to represent sensitivity labels and an attribute to relations for the specification of probability labels. SNOMED-CT uses several relations to relate individuals. For instance, relation *ASSOCIATED WITH* is used to relate an individual of class Disease to an individual of class Substance.

Data Model	**Ontology**
Data type	Class
Data instance	Individual
Hierarchy relation	*IS-A* (is-a-subclass-of) relations
Inference relation	Relations between individuals
Sensitivity label	Attribute of classes and individuals
Probability label	Attribute of relations between individuals

Table 4: Correspondence between the Data Model and Ontologies

Relation *CAUSATIVE AGENT* is used to relate an individual of class Disease to an individual of class Organism. These relations can be seen as instances of our inference relation, as they make it possible to obtain additional information based on the knowledge of a specific instance.

To define the sensitivity of data, we rely on HL7 Healthcare Privacy and Security Classification System (HCS). HL7 HCS provides guidelines and a tagging system for automated labeling and segmentation of protected health care information. Security labels in HL7 HCS are a structured representation of the sensitivity of a piece of information. Relying on HL7 HCS for the definition of data sensitivity has the advantage that the human intervention and judgment is limited and thus the overall outcome of the approach is not affected by the consequent subjectivity. The tagging system provided by HL7 HCS is based on SNOMED-CT (and other code systems). In particular, it provides a partial classification of concepts and individuals in SNOMED-CT. We use this partial classification as the initial assignment and derive the sensitivity for all other classes and individuals using the approach described in Section 5. In particular, *IS-A* relations in SNOMED-CT are used for sensitivity propagation, and the domain relations mentioned above to compute inferred sensitivity.

8.2 Assessing Data Leakage Severity Validation

We evaluated the applicability of L-Severity in a real setting based on the scenario in Section 3. We implemented the hospital database using GNU Health (`http://health.gnu.org`), a healthcare management system used by several healthcare providers worldwide. The system was used to generate a number of data leakages, which have been validated by our industry partner, Roessingh Hospital in the Netherlands.

The generated leakages were manually analyzed by a group of security experts to evaluate the output of L-severity. In particular, we developed a questionnaire describing these leakages; each leakage was described along with its key features. The security experts were invited to answer the questionnaire and evaluate the severity of each data leakage on the basis of the amount and sensitivity of the leaked information as well as the ability to identify the patients to whom the leaked information refers. The security experts assessed the severity of the leakages using a three-valued scale (i.e., low, medium, high severity). Based on this assessment, we built a ground truth data set of leakages along with their severity. For some leakages there was no clear majority in the experts' assessment; thus we considered two additional values namely, low/medium and medium/high. Low/medium is used to represent the experts' assessment when it ranges

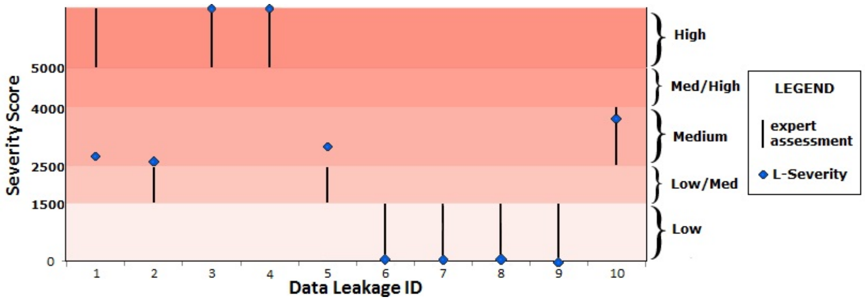

Fig. 3: Evaluation of L-Severity against experts' assessment

in low and medium severity. Similarly, medium/high is used to represent the experts' assessment when it ranges in medium and high severity.

The severity of the same leakages was evaluated using L-Severity. The calculated severity for the leakages is presented in Fig. 3 along with experts' evaluation (in the figure data leakages are identified by an ID). One can observe that the calculated severity matches experts' assessment in nine out of ten cases. Specifically, in seven cases there is a complete match, while in two cases there is a partial match. Therefore, in most cases L-Severity evaluates the data leakages severity correctly. Only one case (#1) presents a notable difference: L-Severity evaluates it to medium severity, whereas the experts evaluated it high. This can be explained by the different weight given to different aspects of the leakage. L-Severity returned medium severity because of the small amount of records leaked. In contrast, the experts weighted more the sensitivity of the leaked information. In cases #2 and #5, the difference between the L-Severity score and experts' assessment is negligible. In both cases L-Severity evaluates it to medium severity, whereas the experts evaluated it low/medium. Therefore, the calculated severity matches the evaluation provided by some experts.

9 Conclusions

In this work we presented a novel approach for the quantification of data leakages with respect to their severity. The assessment of the severity of data leakages considers the amount and sensitivity of the leaked information together with the ability to identify the individuals related to the leaked information. To specify and reason on data sensitivity, we defined a data model representing the knowledge in a given domain. We validated the approach by analyzing data leakages in a typical healthcare environment.

L-Severity as well as M-Score uses a distinguishing factor to determine the level of data anonymization. This factor is based on the number of occurrences of quasi identifiers in the dataset. An interesting direction for future work is to integrate L-Severity with other approaches to data anonymization like differential privacy. Moreover, the alerts generated by a DLD solution may not correspond to data misuses, i.e. alerts may turn out to be false positive. Therefore, the severity of a leakage may not correspond to its risk level (risk is usually defined as the combination of the severity and probability of an event). An interesting direction for further investigation is the integration of our approach with DLD solutions able to determine the probability that an alert is indeed a data breach. This would allow a risk-based ranking of leakages.

References

1. Ponemon Institute: Third annual benchmark study on patient privacy & data security (2012)
2. Banescu, S., Zannone, N.: Measuring privacy compliance with process specifications. In: International Workshop on Security Measurements and Metrics, IEEE (2011) 41–50
3. Information Age: New EU data laws to include 24hr breach notification (2012)
4. Backes, M., Kopf, B., Rybalchenko, A.: Automatic discovery and quantification of information leaks. In: IEEE Symposium on Security and Privacy, IEEE (2009) 141–153
5. Borders, K., Prakash, A.: Quantifying information leaks in outbound web traffic. In: IEEE Symposium on Security and Privacy, IEEE (2009) 129–140
6. Smith, G.: On the foundations of quantitative information flow. In: FOSSACS. LNCS 5504. Springer (2009) 288–302
7. Harel, A., Shabtai, A., Rokach, L., Elovici, Y.: M-score: A misuseability weight measure. IEEE Transactions on Dependable and Secure Computing **9**(3) (2012) 414–428
8. Abbadi, I.M., Alawneh, M.: Preventing insider information leakage for enterprises. In: SECURWARE, IEEE (2008) 99–106
9. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In: Insider Attack and Cyber Security. Adv. Inf. Secur. 39, Springer (2008) 69–90
10. Takebayashi, T., Tsuda, H., Hasebe, T., Masuoka, R.: Data loss prevention technologies. Fujitsu Scientific and Technical Journal **46**(1) (2010) 47–55
11. Koch, R.: Towards next-generation intrusion detection. In: ICCC, IEEE (2011) 1–18
12. Gessiou, E., Vu, Q.H., Ioannidis, S.: IRILD: an Information Retrieval based method for Information Leak Detection. In: EC2ND, IEEE (2011) 33–40
13. Gómez-Hidalgo, J., Martın-Abreu, J., Nieves, J., Santos, I., Brezo, F., Bringas, P.: Data leak prevention through named entity recognition. In: SocialCom, IEEE (2010) 1129–1134
14. Hart, M., Manadhata, P., Johnson, R.: Text classification for data loss prevention. In: Privacy Enhancing Technologies. LNCS 6794, Springer (2011) 18–37
15. Farahmand, F., Navathe, S.B., Enslow, P.H., Sharp, G.P.: Managing vulnerabilities of information systems to security incidents. In: ICEC, ACM (2003) 348–354
16. Garg, A., Curtis, J., Halper, H.: Quantifying the financial impact of it security breaches. Information Management & Computer Security **11**(2) (2003) 74–83
17. Blakley, B., McDermott, E., Geer, D.: Information security is information risk management. In: NSPW, ACM (2001) 97–104
18. Adriansyah, A., van Dongen, B.F., Zannone, N.: Privacy analysis of user behavior using alignments. it - Information Technology **55**(6) 255–260
19. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies **43**(5) (1995) 907–928
20. Doulaverakis, C., Nikolaidis, G., Kleontas, A., Kompatsiaris, I., et al.: GalenOWL: Ontology based drug recommendations discovery. J. Biomedical Semantics **3** (2012) 14
21. OpenGALEN. `http://www.opengalen.org/` (accessed February 24, 2014).
22. SNOMED - CT. `http://www.ihtsdo.org/snomed-ct/` (accessed February 24, 2014).
23. The Open Biological and Biomedical Ontologies Foundry. `http://www.obofoundry.org/` (accessed February 24, 2014).
24. Open Clinical: Ontologies. `http://www.openclinical.org/ontologies.html` (accessed February 24, 2014).
25. The Gene ontology. `http://www.geneontology.org/` (accessed February 24, 2014).