IADIS International Journal on Computer Science and Information Systems
Vol. 14, No. 2, pp. 41-58
ISSN: 1646-3692

ASSURING DATA PRIVACY WITH *PRIVAS* – A TOOL FOR DATA PUBLISHERS

Joana Margarida Miguel[1], Maria João Varanda Pereira[2], Pedro Rangel Henriques[1] and Mario Berón[3]
[1]*Centro ALGORITMI – Universidade do Minho, Portugal*
[2]*CeDRI – Instituto Politécnico de Bragança, Portugal*
[3]*University of San Luis, Argentina*

ABSTRACT

The technology of nowadays allows to easily extract, store, process and use information about individuals and organizations. The increase of the amount of data collected and its value to our society was, at first, a great advance that could be used to optimize processes, find solutions and support decisions but also brought new problems related with lack of privacy and malicious attacks to confidential information.
In this paper, a tool to anonymize databases is presented. It can be used by data publishers to protect information from attacks controlling the desired privacy level and the data usefulness. In order to specify these requirements a DSL (*PrivasL*) is used and the automatization of repository transformation, that is based on language processing techniques, is the novelty of this work.

KEYWORDS

Privacy, Repositories, PPDP, Anonymization, DSL

1. INTRODUCTION

Due to advances in information processing technology and storage capacity, modern organizations end up processing and storing a large volume of data — personal details of individuals and organizations — for multiple purposes. Currently this volume is 2.5 quintillion bytes every day and it tends to increase in the next years (Marr, 2018). In order to analyze such a huge amount of data and extract knowledge from that, there is a strong research line inside the computer science area: artificial intelligence techniques were further explored namely data mining, statistical models used to discover patterns and compare historical data and other machine learning approaches were implemented. (Chakrabarti *et al.*, 2006). Today's large companies, such as Amazon, Google, Facebook, Spotify, take advantages from the amount of data and information they have about their customers. There are also other important organizations that use the data in different ways like in test environments and simulations (Goldman, 2018).

However, the application of these techniques can cause undesirable effects such as viewing personal, private, sensitive and confidential data. In order to produce results from data exploration it is also necessary to raise privacy and information security concerns (Naik and Ghule, 2013). Even if these processes can bring a number of benefits, advantages and discoveries, it is fundamentally to ensure that privacy is guaranteed. For a long period of time, privacy of the user data was neglected (Corrigan, Craciun and Powell, 2014). Nowadays, regulators are starting to worry about such important topics. One of the examples of this newer preoccupation is the EU General Data Protection Regulation (GDPR) (Murphy, 2018), that is already in place since 25th May 2018, and tries to regulate the protection and privacy of user's data and sensitive information.

In order to protect information is crucial to identify all the actors of the data exploration process. The different types of users in this process are: Data Provider (that provides the data), Data Collector (that collects and stores the data provided), Data Publisher (that transforms data and publishes it to be explored) and Data Explorer (that explores the data and retrieves information). Figure 1 illustrates the data exploration process with all of these users and roles.

As privacy threats exist along and in every step of the data exploration process, each one of these users has privacy concerns and is able to ensure privacy with a set of methods and techniques. The Data Provider can protect its data by using external tools to provide fake data or even to limit the quantity and type of information provided when there is an intention to sell its data for some value. Data Collector can take some measures to first collect the data safely and then use some tools to store data while preserving privacy. The Data Publisher can assure the data privacy by adopting and applying the privacy-preserving data publishing (PPDP) techniques. Finally, Data Explorer can assure the privacy preserving by adopting the techniques according with the exploring purpose (for example, for the data mining process a set of privacy-preserving data mining techniques is available) (Xu *et al.*, 2014).

Figure 1. Global process of data exploration

A system called *Privas* has been developed to aid the Data Publisher in its data publishing process. This system accepts a repository and creates a copy maintaining the information to be explored (coherence in data to be analyzed) but assuring that involved individuals/organizations cannot not be identified by applying PPDP techniques. *Privas* offers *PrivasL*, a Domain Specific Language (*DSL*) that easily allows to specify: the original repository schema, the identification of the tables/columns that one wants to explore and the definition of the privacy level to be assured. The *PrivasL* specification is submitted to *Privas* processor that interprets it. Then *Privas* automatically chooses the best techniques to apply to the repository in order to transform it and improve its privacy level. The compilers' generator — ANTLR4 — is used to implement *PrivasL* processor. This brings novelty and value to our contribution comparing to the actual manual implementation of anonymization techniques.

In Section 2 we shortly present the most relevant 'privacy-preserving data publishing (PPDP)' methods or techniques proposed in the literature for data anonymization after being collected and before exploration, balancing privacy assurance and information preservation. In

that section three different perspectives are discussed. We did not include specifically a related work section because all the tools available do not automatically anonymize data and either focus on masking the data or require knowledge, expertise and configuration to apply the anonymization techniques. Our proposal, *Privas* tool, is introduced in Section 3. In that Section 3, after an overview of the system features and a brief description on how *Privas* is integrated in the flow of the global process for data exploration, *Privas* architecture is explained in detail. Then a description of *PrivasL*, designed to describe privacy concerns, is presented along with how to measure the percentage of information loss; the way how anonymization techniques are implemented is also referred in the same section. Before concluding the paper in Section 5, a Case Study is discussed in Section 4. In that section, *Sakila* database is presented highlighting the privacy problems arising in that context; then the desired transformation for privacy preservation is shown written in *PrivasL* and the tables transformed according to that description are also shown; a discussion about the results obtained closes Section 4.

2. DATA ANONYMIZATION - TECHNIQUES AND METHODS

As it was discussed, the Data Publisher is in charge of select and transform the data to be provided. Typically, this process brings a loss of information's usefulness. The process of balancing the privacy with the loss of information is commonly referred as *privacy-preserving data publishing (*PPDP*).* It will always be necessary, and it will always be the biggest challenge, that Data Publisher factors these two weights during the *PPDP* process to ensure that the collected data is useful so it can be later explored. This challenge raises three questions (Wong and Fu, 2010).

2.1 How Does the Data Publisher Prepares the Data to be Modified?

In order to be able to answer this question, it is necessary to understand some fundamental concepts used in the *PPDP*. The existing information and its parts can be classified into different attribute types (Wong and Fu, 2010; Sharma Amita *et al.*, 2014; Xu *et al.*, 2014):

- **Identifier (ID)**: It is an attribute or set of attributes, such as name, telephone number, social security number, which contains information that allows to directly and uniquely identify an individual;
- **Quasi-identifier (QID)**: a set of attributes that can potentially lead to the identification of record owners (e.g. in (Sweeney, 2000), the report stated that in a U.S. Census the set of 5-digit Zip, gender and birth date, allowed the identification of 87% of the population);
- **Sensitive Attribute (SA)**: consist of information specific to each individual they wish to enclose, such as illness, salary value, level of disability, etc.;
- **Non-Sensitive Attributes (NSA)**: all attributes that do not fit in the three previous categories are non-sensitive attributes.

Based on this categorization a set of *data anonymization* techniques can be applied. (Fung *et al.*, 2010): Generalization, Bucketization, Suppression, Anatomization, Permutation, and Perturbation. Each of these techniques ends up being used inside the algorithms developed to implement the anonymization.

2.2 How Does the Data Owner Guarantee that the Modified Data are Protected from Attacks?

The assurance that the modified data is protected can be given by quantifying the preservation of privacy according to the type of privacy threats (Fung *et al.*, 2010).

Table 1. Main Privacy models with associated attack models. Adapted from (Fung et al., 2010; Mendes and Vilela, 2017)

Privacy Model	Description	Application and Domains	Attack Model			
			Record Linkage	Attribute Linkage	Table Linkage	Probabilistic Attack
k-Anonymity (P. Samarati and Sweeney, 1998; Pierangela Samarati and Sweeney, 1998)	Anonymity is guaranteed by the existence of at least other k-1 undistinguishable (w.r.t. the QID) records for each record in a database. This group of k undistinguishable records is referred to as equivalence class.	Wireless Sensor Networks (Groat, Hey and Forrest, 2011), Location-based services (Bamba *et al.*, 2008), Cloud (He *et al.*, 2016), E-health (Gal, Chen and Gangopadhyay, 2008)	✓			
l-Diversity (Machanavajjhala *et al.*, 2006)	Expands the k-anonymity model by requiring every equivalence class to have at least one "well-represented" value for the sensitive attributes.	E-health (Gal, Chen and Gangopadhyay, 2008; Kim, Sung and Chung, 2014), Location-based services (Bamba *et al.*, 2008; Liu, Hua and Cai, 2009)	✓	✓		
t-Closeness (Ninghui, Tiancheng and Venkatasubramanian, 2007)	Extends the l-diversity model by treating the values of a sensitive-attribute distinctly by taking into account the sensitive-attribute's distribution of data values.	Location-based services (Riboni *et al.*, 2009)		✓		✓
Personalized Privacy (Xiao and Tao, 2006)	Achieved by creating a taxonomy tree using generalization, and by allowing the record owners to define a guarding node. Owners' privacy is breached if an attacker is allowed to infer any sensitive value from the subtree of the guarding node with a probability (breach probability) greater that a certain threshold.	Social Networks (Yuan, Chen and Yu, 2010), Location-based services (Agir *et al.*, 2014; Ghasemi Komishani, Abadi and Deldar, 2016)]		✓		
ε-Differential Privacy (Dwork, 2006)	Ensures that a single record does not considerably affect (adjustable through the value ε) the outcome of the analysis of the dataset. In this sense, a person's privacy will not be affected by participating in the data collection since it will not make significant difference in the final outcome.	E-health (Dankar and El Emam, 2013; Lin *et al.*, 2016), Smart meters (Zhang *et al.*, 2017), Location-based services (Elsalamouny and Gambs, 2016)			✓	✓

According to (Fung et al., 2010), threats to privacy can be classified into two categories:

- The first category considers that the adversary or attacker is capable of identifying the record of a target individual by linking the record to data from other sources, such as

linking the record to a record in a published data table (this is called record linkage method), to a sensitive attribute in a published data table (this is called attribute linkage method), or to the published data table itself (this is called table linkage method);

- The second category aims at achieving the uninformative principle (Machanavajjhala *et al.*, 2006): consider that the attacker or adversary has enough background knowledge to execute a probabilistic attack, that is, the adversary is able to make a confident inference about whether the target's record exist in the table or which value the target's sensitive attribute would take and because of that, the publish data, cannot disclose additional information beyond the background knowledge that may already have.

According to the attack models and to measure the quantification of privacy preservation, different privacy models were proposed. In the Table 1 it is possible to consult the main models of privacy according to the different types of attack, as well as a small description. New privacy models have appeared recently but contain small differences from the models presented.

Each privacy model in its definition uses techniques from different categories presented previously. Multiple algorithms were developed over the years to achieve such techniques.

Figure 2 gives a general overview of how privacy models are linked to the privacy attacks, privacy attributes, privacy techniques and algorithms. The Figure helps to understand how the concepts around the privacy context are connected.

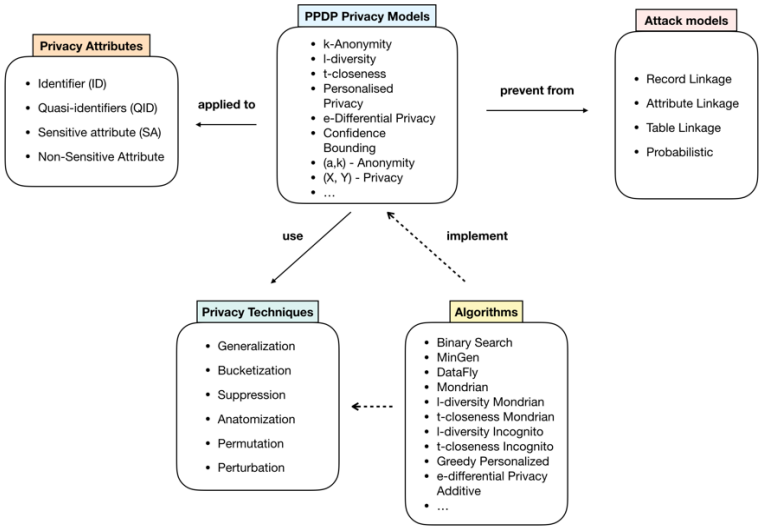

Figure 2. Privacy domain context. Different concepts of the privacy domain with its connections

2.3 How Much Does the Data Needs to be Modified so that no Sensitive Information Remains?

In order to transform the data and generate new data without sensitive information, the Data Publisher can use several techniques. One needs to remember that when the data is changed, exists an impact in its usefulness. There is always a trade-off between privacy and usefulness. The transformation of data to ensure privacy can be done in multiple ways and with several techniques and it will result in information with different utility.

Since there are available several ways to transform data, the Data Publisher should choose the one that seems to be the most useful. Generally the one that contains more valuable information for the data analysis (Wong and Fu, 2010), but that criteria can change depending on the purpose of the exploration phase.

3. *PRIVAS* - AUTOMATIC *ANONYMIZATION* OF DATA REPOSITORIES

There are lots of techniques to provide data privacy protection at the publisher stage. The main focus of the literature so far has been finding and/or creating new and better techniques to apply privacy to data. Due to already exist a large number of techniques and ways of protecting privacy and as they require some study on how they should be applied, the choice and use of these techniques is still an ad-hoc choice in accordance with business solutions and types of data.

As a way to help and to promote privacy protection, this work aims to present a solution that helps to choose and use the various techniques and methods of privacy protection in data repositories in the data publishing phase (PPDP). This solution creates a tool - named *Privas* - that enables to:

- Specify the type of data repository to be treated;
- Identify the existing information in the repository and classify its type (ID, QID, sentive, none) – this step is currently manual;
- Set desired privacy level (choosing the type of attacks to prevent);
- Produce a metrics of the utility of data still present with the desired level of privacy;
- Apply privacy protection techniques (PPDP) and methods to the specified repository.

Figure 3. Global process of data exploration with *Privas*

The goal is that, as shown in Figure 2, the Data Publisher user has a tool available that, after configuring the parameters, will automatically transform their data and prepare them for publication with the type privacy requested. To achieve this goal, the *Privas* tool and architecture was thought as a set of components/parts that serve different purposes.

3.1 Tool Architecture

Privas' architecture can be seen in Figure 3. The different components present on *Privas* are:

1. *PrivasL*, a domain-specific language (*DSL*) that allows the description of the repository type, the desired privacy level and the data types classification for that privacy level. This language must be expressive enough to allow all the needed specifications, and simple and intuitive to be easily learned by anyone;

2. A core software unit, agnostic to the kind of repository, which will have the techniques and modes of privacy protection and the logic to add privacy to the data of the repository - by having this core unit, the goal is to allow an easier evolution of the tool to other repositories and techniques, and make it a task specific and simplified;
3. A specific connector for each type of repository that will be responsible to transform the information into a format that the core software unit can then process and later generate the new transformed repository;
4. A web interface that offers users a visual tool to obtain *PrivasL* descriptions. Within the web platform the user defines all the details about the repository and the desired transformation level and, in the end, a *PrivasL* description is generated.

These parts/components are an integral part of a process (Figure 3) of defining the repository data and its types and enable the transformation of data in a way that guarantees privacy.

By zooming into the core of *Privas* tool, one can see that *Privas* is developed through several independent and perfectly separable components, which are:

- The *PrivasL* parser;
- The connector to each repository type;
- The set of PPDP techniques and models;
- A core unit that connects everything with a decision engine to automatically apply the techniques.

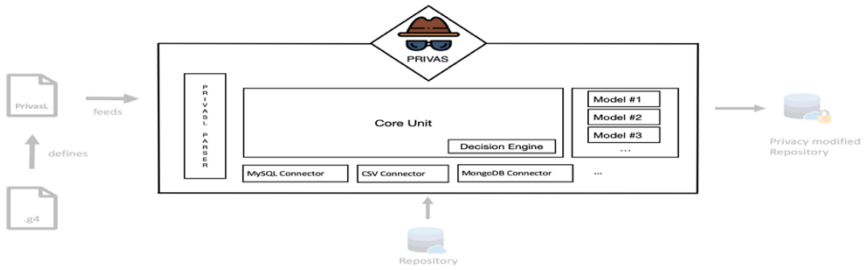

Figure 4. *Privas* Core architecture

The main idea behind this division is to allow the *Privas* tool to be modular and extensible, and therefore, easily scalable. Thus, new techniques can be easily added, and new repositories can be configured in few steps.

In the next subsection, a *DSL* to classify and configure the desired privacy level is shown.

3.2 *PrivasL* - A *DSL* to Classify Original Data

The description written in *PrivasL* allows to configure all the transformation process:

- Configure the repository type and its basic information (connection string, path, etc.);
- Classify each data type of information present on the repository being handled (he can choose between ID, QID, SA, NSA);
- Choose what type of attack models the repository should be protected from. These attack models can be *Record Linkage*, *Attribute Linkage*, *Table Linkage* and *Probabilistic Attack*;
- Optionally define generalization trees to give domain context when anonymizing.

Actually, there are combinations of type of attacks that are not covered by any of the Privacy Models available. The *PrivasL* processor detects this as an invalid configuration sending an alert to the user.

What the user has to provide through *PrivasL* is represented in Listing 1:

- Database name;
- Specify the tables where the privacy techniques will be applied, and for each of them classify the attributes in *PPDP* ('@': attribute identifier (ID); '&': attribute quasi-identifier (QID); '~': attribute sensitive (SA); and nothing: none of the others);
- Type of attack (or set of attacks) that is intended to protect which offers some degree of privacy. For example, the user can choose to protect from the record linkage attack;
- Optionally define generalization trees for the attributes to obtain a domain anonymization.

```
privas : repositoryOptions
dataDescription privacyOptions
generalizationTrees?;


repositoryOptions
        : 'Relational'
relationalRepositoryOptions
        | 'CSV' csvRepositoryOptions
        ;

relationalRepositoryOptions
        : 'Connection String:'
STRING_LITERAL;

csvRepositoryOptions : 'Path:'
STRING_LITERAL;

dataDescription : NAME entities;

entities : entity ( entity* );

entity : NAME '[' attributeList ']';

attributeList : attribute ( ','
attribute )*;

attribute : attributeValue
attributeTree?;

attributeTree : '::' NAME;

attributeValue : '@' NAME   | '&'
NAME
               | '~' NAME   | NAME
;

privacyOptions : 'Prevent from:' '['
attackModelList ']';
```

```
attackModelList : attackModel ( ','
attackModel )*;

attackModel: attackModelName
attackModelParameters?;

attackModelParameters: '<' NAME '='
NUM (',' NAME '=' NUM)* '>';

attackModel : 'recordLinkage' |
'attributeLinkage'
            | 'tableLinkage'. |
'probabilisticAttack'
            ;

generalizationTrees :
generalizationTreeHeader+;

generalizationTreeHeader
    : '+' NAME ('as' NAME)?
generalizationTree+
    ;

generalizationTree
    : '+' NAME generalizationTree{2,}
    | '>' STRING_LITERAL ':'
setOfValues
    | '>' STRING_LITERAL ':' range;

setOfValues
    : '{' STRING_LITERAL (','
STRING_LITERAL)* '}' ;

range : (']' | '[') NUM? ',' NUM? (']'
| '[') ;


NAME : [a-zA-Z][-_a-zA-Z0-9]*;
NUM  : [1-9][0-9]*;
STRING_LITERAL:   '"' (~["\\\r\n])*
'"';
```

Listing 1. *PrivasL* DSL BNF definition - extended AntLR

When anonymizing the repository, the algorithms can transform the value in different ways. When a generalization occurs, *Privas* has two types of generalizations already built-in:

- For numbers and dates, as they are continuous values, *Privas* offers out-of-the-box generalization in ranges between values. The number ranges are built from the values. The date ranges start by generalizing the month and can go through the year part.
- For every other value not specified, Privas treats them as strings. So, when generalizing two values Privas tries to join the values with `~' (e.g. 'value1~value2'). When more than two values are involved, *Privas* generalizes to a more generic form to avoid disclosing much information while helping the generalization process: `*' character.

Besides these built-in generalizations, and to allow a better and more adapted to the domain context generalization, *Privas* also offers the user the possibility of defining generalization trees. This tree should be composed by different levels. Each level represents a possible generalization and holds its value. In the leaves it will be the raw values present in the repository. Figure 4 depicts a generalization tree for a numerical value (votes for instance a price attribute. Listing 2 contains the tree represented in *PrivasL* description.

Figure 5. Example of a tree generalization for a price attribute

```
+ Price
  > Low: [0, 0.5[
  + Medium
    > 'Medium Low': [0.5, 1[
    > 'Medium High': [1, 2[
  > High: [2,[
```

Listing 2. Generalization of Price attribute written in *PrivasL*

By using the *PrivasL*, the user of the tool provides all the information (type of information present in the data repository and attack to prevent from) required and desired in a uniform way, making this configuration confined to a unique entry point.

3.3 Implementing Anonymization Techniques

The Anonymization techniques are contained in a component reserved for such (Figure 3). The user specifies through *PrivasL* descriptions the types of privacy attacks to protect the repositories from. And, an available and appropriate model and its techniques is chosen and applied, transforming the repository in a new repository with data *anonymized*.

The tool implements four privacy models: the **k-anonomity model**, the **l-diversity model**, the **t-closeness model**, and the **e-differential privacy model.** With these models, and as seen in Table 1, in conjunction, *Privas* cover all of the attack models.

The ***k*-anonomity model** is implemented through the generalization and suppression techniques. ***k*-anonomity model** guarantees that for each QID present, there are at least *k* entities with the same value, making them indistinguishable. As seen in Table 1, this privacy model protects the repository against Record Linkage attacks. The algorithm chosen to implement the **k-anonomity model** was Mondrian. Mondrian is a Top-down greedy data anonymization algorithm for relational dataset, proposed by Kristen LeFevre in (LeFevre, DeWitt and

Ramakrishnan, 2006). The algorithm is based on the concept of data partitioning that is a clear connection with the sections that are so characteristic of Pietre Mondrian work arts.

Figure 6 instantiates the Figure 2 for the ***k*-anonymity** and shows this privacy model and the surrounding concepts that are tied to it.

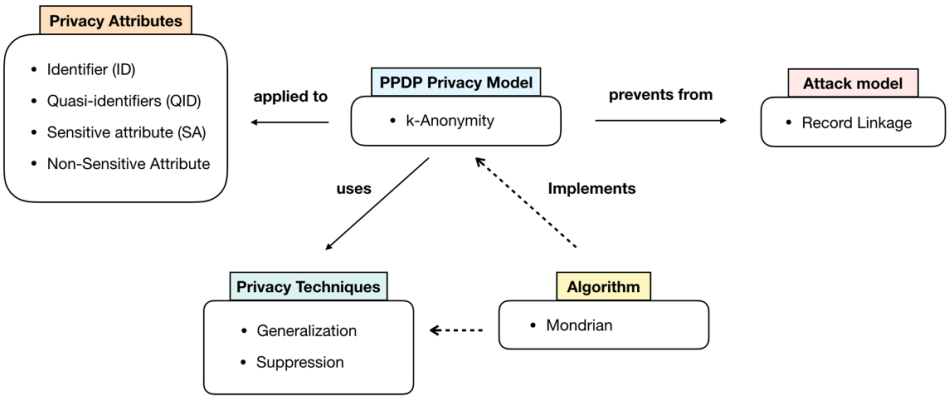

Figure 6. k-anonymity privacy domain context

The ***l*-diversity model** was conceived as an extension of the ***k*-anonymity model** and as such, it inherits all the properties of the latter model including the protection against Record Linkage. Besides these properties it also ensures that each anonymous group contains at least *l* different values of the sensitive attribute value – preventing against Attribute Linkage. Therefore, even if an adversary can identify the group of an individual it still would not be possible to find out the real value of that individual's sensitive attribute with certainty.

To implement **l-diversity** the algorithm followed was adapt the Mondrian base algorithm and adapt it to achieve l-diversity. The required changes to achieve ***l*-diversity** with Mondrian were 1) modify the function that validates the partitions to include the diversity and 2) modify the split function to produce partitions that are diverse (if possible).

Like ***l*-diversity**, ***t*-closeness model** extends the ***k*-anonymity model** but it offers different characteristics when applying the privacy preservation. The concept behind t-closeness is that the statistical distribution of the sensitive attributes' values in each k-anonymous group is "close" to the overall distribution of that attribute in the entire repository. This property prevents against the identification of the individual in the privacy resulting dataset and it prevents against Attribute Linkage as it ensures the closeness between different entities and against Probabilistic Attack as it not discloses disperse information.Typically, the closeness between two elements can be measured using different mathematic formulas, here it was opted for the Kolmogorov-Smirnov distance. To implement ***t*-closeness** the chosen algorithm was also the Mondrian adapted to this model. The changes to the algorithm to achieve this were 1) modify the function that validates the partitions to include the diversity against global data and 2) use the Kolmogorov-Smirnov distance to obtain the diversity validity.

The ***e*-differential privacy model** works differently from the previous privacy models implemented. It limits the knowledge gain between repositories that differ in one individual. This property offers this model prevention against Table Linkage and Probabilistic Attack. The way this limit is done is by replacing the individual's values for some value (mean of values for example) that dilutes the presence of different individuals. For example, if the value being

anonymized is the salary, one could replace the salary values by the mean of all salaries. This imply that adding or removing a new salary would not impact the mean value. Because of its nature and how ***e*-differential privacy** works, its utilization is best suited to statistical databases. The *e* parameter is sometimes referred as the privacy-loss budget, meaning that greater the *e* value less privacy will exist, and more information will be on the repository. In (Domingo-Ferrer and Soria-Comas, 2015), the authors reached a conclusion that the privacy model t-closeness when $t = exp^{\frac{e}{2}}$, yields *e*-privacy. And for these reasons, to implement this technique, the *t*-closeness was used with the tweak in the *t* parameter.

The *Privas* tool encloses all these privacy models and automatically chooses and applies them when different type of attacks is chosen. To allow an advanced configuration about the models being applied, the tool also offers the possibility of specifying values of the parameters (*k*, *l*, *t* and *e*, depending on the privacy model being applied).

3.4 Measuring the Information Loss Percentage

Knowing the impact, the privacy transformation caused on the data is crucial to take better decisions on what type of attacks to prevent from, and if the information chosen to be sensitive can be more relaxed. This metric can also give the Data Publisher a strong indicator if it makes sense to publish its data.

Privas also produces this output after transforming the data. For each entity transformed (a table in case of Relational Databases), a % of the information loss is displayed. This information loss percentage is calculated by knowing how much the privacy has affected the entity.

Each entity has N attributes (columns in case of a table) and L lines. To calculate the information loss percentage is required to calculate and sum the information loss percentage for each attribute. For each attribute, the information loss percentage of its elements (specific line in an attribute), must be added. The information loss percentage of each element depends on if it suffered suppression (all the information loss) or if it suffered #g generalizations, meaning it lost $\frac{\#g}{1+\#g}$, between 0 and 1 of information. The following formula calculates the information loss percentage for each entity:

$$\%informationLoss_{entity} = \frac{100}{N * L} * \sum_{i=1}^{N} \sum_{j=1}^{L} \begin{cases} 1, & element_{i,j}\ suffered\ suppression \\ \frac{\#g}{1+\#g}, & element_{i,j}\ suffered\ generalization \end{cases}$$

3.5 Web Platform

To reach multiple and different types of users to use *Privas* to transform their repositories, a Web Platform was designed and implemented. This web platform was built as a Single Page Application, using Interaction Design Patterns (to help having a more natural user experience).

The platform contains all the configuration information present in the *PrivasL* and described in Section 3.3. It works by helping build the information about the repository incrementally and in the end generate a description in *PrivasL* language. Figure 7 depicts the Web Platform main page where all the description options can be seen.

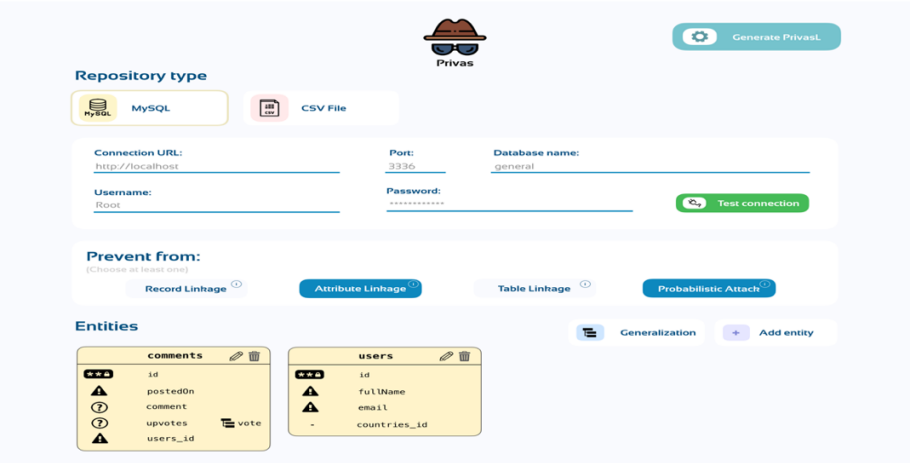

Figure 7. Web platform (SPA) that generates *PrivasL* descriptions

4. SAKILA CASE STUDY

Relational databases are widely used and well-suited in knowledge extraction processes due to their strong structuring of data. Because of its widely adoption and usage in the real world we chose a relational database as a case study of the *Privas* tool. The database chosen was *Sakila* database - a MySQL relational database (https://dev.mysql.com/doc/sakila/en/). The *Sakila* sample database was initially developed by a former member of the MySQL documentation team and aims to provide a standard schema easily available to all. This database is a nicely normalized schema modelling a DVD rental store, featuring things like films, actors, film-actor relationships, and a central inventory table that connects films, stores, and rentals.

Because of type of information it contains, it has several tables with interesting attributes to be analyzed at the privacy level (such as customer information, staff, payments, etc).

To apply *Privas* tool to the *Sakila* database we simulate the use of *Privas* by a regular user:

1. We analyzed all the tables individually to identify if the table needed some kind of privacy transformation - this was done by classifying each information with either ID, QID, SA or NSA;
2. After visiting every table and classifying its information, we collected the tables' names and its attributes with the information classification and described in *PrivasL*;
3. We added the rest of information needed to *PrivasL* description (database connection and type of attacks we are preventing the repository from).

The list of tables in the repository is: *actor*, *address*, *category*, *city*, *country*, *customer*, *film*, *film_actor*, *film_category*, *film_text*, *inventory*, *language*, *payment*, *rental*, *staff*, and *store*.

From the analysis of all tables and from the information classification we concluded that tables like *actor*, *category*, *city*, *country*, *film*, *film_actor*, *film_category*, *film_text*, *inventory*, *language*, and *store*, do not have privacy transformation needs given that all the information present, besides database domain value like primary keys or foreign keys, is classified as *Non-Sensitive Attribute*.

4.1 *PrivasL* Applied

On Listing 2 one can see the *PrivasL* description for the *Sakila* database. We can conclude that from 16 tables, 5 have privacy needs, and from those a minor part of its attributes has privacy concerns.

```
Relational
    Connection String: "jdbc:mysql://localhost:3306"

sakila
    address [ &address, &district, &postal_code, @phone ]

    customer [ &first_name, &last_name, @email ]

    payment [ &payment_date ]

    rental [ &rental_date, &return_date ]

    staff [ &first_name, &last_name, @email, @username ]


Prevent From: [ recordLinkage ]
```

Listing 3. *PrivasL* description to apply Privas to Sakila database

From the analysis we concluded that no attribute has sensitive information that should be taken care. As an example, for this type of attribute would be the staff table having salary information of each staff member. This information almost certainly would not be a Quasi-Identifier but would be the type of information that is sensitive and should be classified as such.

4.2 Results Obtained

After running the *Privas* Tool with the specified *PrivasL description* (Listing 2), we obtained a new privacy transformed database. With *Record Linkage* to prevent from, *Privas* applied the *k*-anonymity. Figure 4 represents three examples of *Sakila* tables and its transformation.

On the upper part is the original data and, on the right-bottom part the table with privacy transformation. It is clear from the data transformed for each entity that all the information classified as Identifier or Quasi-Identifier has been processed. The algorithm applied guaranties that at least 2 lines ($k = 2$) have the same values of Quasi-Identifiers in each table, preventing the attacker of knowing what entity holds that information.

The staff table, for example, even though *first_name* and *last_name* were chosen as Quasi-Identifiers, its generalization (due to the need of $k = 2$) behaved as the columns were suppressed.

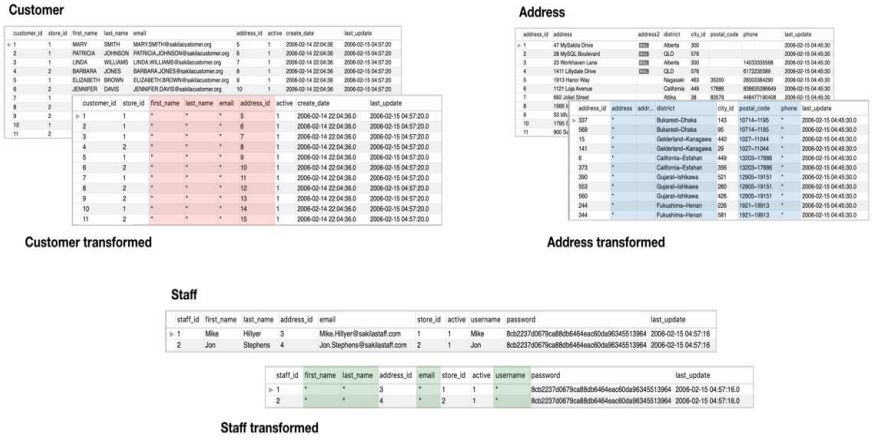

Figure 8. Three examples of *Sakila* tables before and after *Privas* process

4.3 Discussion

Some of the conclusions that are directly extracted is that the type of information present is not so rich and definitely its transformation had an impact on the information present in the table.

The information lost percentage was calculated with the algorithm presented before (Section 3.4). For each *table* of the *Sakila's* database the values can be consulted in Table 2.

Table 2. Information loss percentage for each table of *Sakila* relational database

Table	Information Loss
address	34.41%
customer	33.33%
payment	7.44%
rental	14.76%
staff	40.00%

The quantity of information loss is not far from what was expected: the tables with more Identifiers and Quasi-Identifiers by total number of columns, have a higher *Information Loss* percentage value. One factor that also weights to the information loss is the natural distribution of the data. If data already contains little values that make the entities identification impossible the need to *anonymize* it would be lower and the information loss would have a low value.

The table *staff*, that has the higher number of *Identifiers* identified (values that are suppressed) is also the table with the higher value. Table *payment*, on the other hand, has the lower value, which can also be explained because it only has one Quasi-Identifier identified.

With this information loss values calculated, if there were different techniques to apply that guaranteed the user with the same prevention for the requested attack, the algorithm that selects and applies the techniques could select the best suited technique for this use case. This is

something that *Privas* was built to support from its foundations since it allows the existence multiple models to solve the same attack privacy threads.

Even with the information loss percentage calculated, the data publisher should always look into the output produced. For example, in Figure 4, one can see that the information remaining in the table *staff* is not that useful, due to the privacy transformed table only contains suppressed attributes and references to other tables.

5. CONCLUSION

Due to the technological evolution that has been observed in the last decades, the data and the information of individuals have gained more and more value. Data is a high value asset for all types of organization and is exploited for various purposes. No matter the purpose, all these exploration processes share some steps and actors; they have a common flow. This work focusses in the data publishing phase, where the most relevant user is the Data Publisher.

Though the information and knowledge discovered by the exploration and use of data can be very valuable in many applications, people are increasingly concerned about the other side of the coin: the privacy threats that these processes bring. With the methodology based on the role of the user, it is considered that the Data Publisher should assume the main responsibility of protecting confidential data. Therefore, it must follow techniques that anonymize the original data so that it is not possible to identify the owner of the data. These techniques are categorized under the privacy-preserving on data publishing techniques (*PPDP*). The current problem is that the Data Publisher's role is performed by the Data Collector or Explorer, that already has other concerns. Ideally, Data Explorer should receive the data with no sensitive information in it, that was treated by the Data Publisher.

Although there is already a lot of information and many techniques in the bibliography, the Data Publisher in order to implement them has to do it as an ad-hoc process: analyze tables, attributes, data types, choosing technique, algorithm, apply rule by rule and repeat. Till the moment, it was not found any automatic approach to do so. So, *Privas* can be considered a valuable contribution in this field.

Privas, using language processing techniques, allows to apply *PPDP* techniques to a repository. The tool produces a new repository with the privacy assured to some level. The architecture of this tool has been developed in a way to be divided into several components, which allows to easily add new techniques and new types of repositories in order to evolve the tool. One of the main components is *PrivasL* that allows to specify all the information that the user will have to provide: type of repository, entities to apply the privacy, the attributes and their types in *PPDP* format. The *DSL* is the only point of entry of inputs which makes the process simpler, because the user just has to write a textual description or fill the fields in a web interface.

Privas tool, and the *PrivasL*, along with the web platform were successfully tested with 3 case studies: 1) *Sakila* relational database presented in Section 4, 2) An U.S. census dataset, and 3) an Employees relational DB (a large database with more than 4 million records with sensitive information). The required time for the transformation in each repository was not significant.

Each technique has its advantages and limitations, so it is important that the tool behaves and offers several technical options to the user. No technique is ideal, and data privacy and utility are inversely proportional, so when gains occur in privacy it means utility has suffered some loss. Therefore, in the end, there is the need to provide the user with metrics that will allow

to analyze the trade-off between data utility and data privacy. As presented in this paper, *Privas* already automatically prevents against all types of attacks by having four privacy model implemented (*k-anonymity, l-diversity, t-closeness, and e-differential privacy*). The next steps could be to increase the number of repositories that *Privas* supports, increase the number of models and algorithms implemented, and enhance the decision engine to offer a better performance when deciding which technique to apply.

A completely automatic tool that is able to discover the sensitive attributes and to choose the most probably attack model that should be avoided would be a big challenge. Data mining applied to databases could be used to take this kind of decisions. Still, by focusing on the attack prevention and not on specific techniques, *Privas* eases and automates the anonymization process. The automatic choice of what technique to apply saves the user from having to have a deep knowledge in privacy domain while also sparing a lot of time when compared to the completely manual application of the privacy techniques.

ACKNOWLEDGEMENT

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

REFERENCES

Agir, B. *et al.* (2014) 'User-side adaptive protection of location privacy in participatory sensing', *GeoInformatica*. doi: 10.1007/s10707-013-0193-z.

Bamba, B. *et al.* (2008) 'Supporting anonymous location queries in mobile environments with privacygrid', in *Proceeding of the 17th international conference on World Wide Web - WWW '08*. doi: 10.1145/1367497.1367531.

Chakrabarti, S. *et al.* (2006) *Data Mining Curriculum: A Proposal (Version 1.0), ACM SIGKDD*. doi: 10.2307/2627828.

Corrigan, H. B., Craciun, G. and Powell, A. M. (2014) 'How Does Target Know So Much About Its Customers? Utilizing Customer Analytics to Make Marketing Decisions', *Marketing Education Review*. doi: 10.2753/MER1052-8008240206.

Dankar, F. K. and El Emam, K. (2013) 'Practicing differential privacy in health care: A review', *Transactions on Data Privacy*. doi: 10.1309/LMAGPAENJKNARI4Z.

Domingo-Ferrer, J. and Soria-Comas, J. (2015) 'From t-closeness to differential privacy and vice versa in data anonymization', *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2014.11.011.

Dwork, C. (2006) 'Differential privacy', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/11787006_1.

Elsalamouny, E. and Gambs, S. (2016) 'Differential privacy models for location-based services', *Transactions on Data Privacy*. doi: 10.1097/IOP.0b013e3181e2f96f.

Fung, B. C. M. *et al.* (2010) 'Privacy-Preserving Data Publishing: A Survey on Recent Developments', *Computing*. doi: 10.1145/1749603.1749605.

Gal, T. S., Chen, Z. and Gangopadhyay, A. (2008) 'A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes', *International Journal of Information Security and Privacy*. doi: 10.4018/jisp.2008070103.

Ghasemi Komishani, E., Abadi, M. and Deldar, F. (2016) 'PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression', *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2015.11.007.

Goldman, J. (2018) 'How Companies Like Amazon and Google Turn Data Into a Competitive Advantage — and How You Can Too'. Available at: https://www.inc.com/jeremy-goldman/how-companies-like-amazon-google-turn-data-into-a-competitive-advantage-how-you-can-too.html.

Groat, M. M., Hey, W. and Forrest, S. (2011) 'KIPDA: K-indistinguishable privacy-preserving data aggregation in wireless sensor networks', in *Proceedings - IEEE INFOCOM*. doi: 10.1109/INFCOM.2011.5935010.

He, X. M. *et al.* (2016) 'Semi-Homogenous Generalization: Improving Homogenous Generalization for Privacy Preservation in Cloud Computing', *Journal of Computer Science and Technology*. doi: 10.1007/s11390-016-1687-6.

Kim, S., Sung, M. K. and Chung, Y. D. (2014) 'A framework to preserve the privacy of electronic health data streams', *Journal of Biomedical Informatics*. doi: 10.1016/j.jbi.2014.03.015.

LeFevre, K., DeWitt, D. J. and Ramakrishnan, R. (2006) 'Mondrian multidimensional K-anonymity', in *Proceedings - International Conference on Data Engineering*. doi: 10.1109/ICDE.2006.101.

Lin, C. *et al.* (2016) 'Differential Privacy Preserving in Big Data Analytics for Connected Health', *Journal of Medical Systems*. doi: 10.1007/s10916-016-0446-0.

Liu, F., Hua, K. A. and Cai, Y. (2009) 'Query l-diversity in location-based services', in *Proceedings - IEEE International Conference on Mobile Data Management*. doi: 10.1109/MDM.2009.72.

Machanavajjhala, A. *et al.* (2006) 'ℓ-Diversity: Privacy beyond k-anonymity', in *Proceedings - International Conference on Data Engineering*. doi: 10.1109/ICDE.2006.1.

Marr, B. (2018) 'How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read'. Available at: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/.

Mendes, R. and Vilela, J. P. (2017) 'Privacy-Preserving Data Mining: Methods, Metrics, and Applications', *IEEE Access*. doi: 10.1109/ACCESS.2017.2706947.

Murphy, J. F. A. (2018) 'The General Data Protection Regulation (GDPR)', *Irish Medical Journal*. doi: 10.1007/978-3-319-57959-7.

Naik, D. P. and Ghule, A. N. (2013) 'An Advanced Data Transformation Algorithm for Categorical Data Protection', 4(6), pp. 899–902.

Ninghui, L., Tiancheng, L. and Venkatasubramanian, S. (2007) 't-Closeness: Privacy beyond k-anonymity and ℓ-diversity', in *Proceedings - International Conference on Data Engineering*. doi: 10.1109/ICDE.2007.367856.

Riboni, D. *et al.* (2009) 'Preserving anonymity of recurrent location-based queries', in *TIME 2009 - 16th International Symposium on Temporal Representation and Reasoning*. doi: 10.1109/TIME.2009.8.

Samarati, Pierangela and Sweeney, L. (1998) 'Generalizing data to provide anonymity when disclosing information (abstract)', in *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '98*. doi: 10.1145/275487.275508.

Samarati, P. and Sweeney, L. (1998) 'Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppresion.', in *Proc of the IEEE Symposium on Research in Security and Privacy*. doi: http://dx.doi.org/10.1145/1150402.1150499.

Sharma Amita *et al.* (2014) 'A Survey on Techniques for Privacy Preserving Data Publishing (PPDP)', *International Journal on Cybernetics & Informatics*, 3(1), pp. 1–8. doi: 10.5121/ijci.2014.3101.

Sweeney, L. (2000) 'Simple demographics often identify people uniquely', *Carnegie Mellon University, Data Privacy Working Paper 3. Pittsburgh 2000*. doi: 10.1016/S1003-6326(13)62577-7.

Wong, R. C.-W. and Fu, A. W.-C. (2010) 'Privacy-Preserving Data Publishing: An Overview', *Synthesis Lectures on Data Management*. doi: 10.2200/S00237ED1V01Y201003DTM002.

Xiao, X. and Tao, Y. (2006) 'Personalized privacy preservation', in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data - SIGMOD '06*. doi: 10.1145/1142473.1142500.

Xu, L. *et al*. (2014) 'Information security in big data: Privacy and data mining', *IEEE Access*. doi: 10.1109/ACCESS.2014.2362522.

Yuan, M., Chen, L. and Yu, P. S. (2010) 'Personalized privacy protection in social networks', *Proceedings of the VLDB Endowment*. doi: 10.14778/1921071.1921080.

Zhang, Z. *et al*. (2017) 'Cost-Friendly Differential Privacy for Smart Meters: Exploiting the Dual Roles of the Noise', *IEEE Transactions on Smart Grid*. doi: 10.1109/TSG.2016.2585963.