**PRIVA - Anonymization Tool**

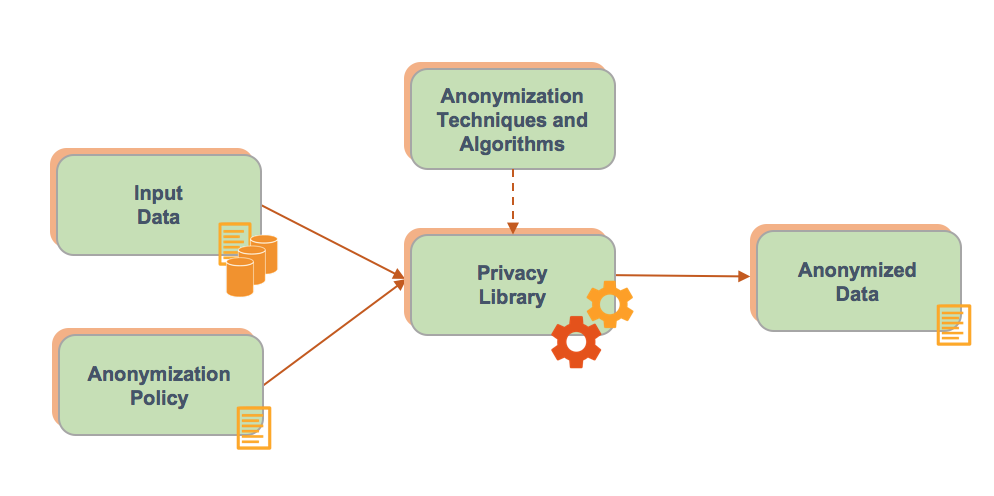
**Documentation Release (r1.0)**

**User Guide - A detailed end-user documentation**

1. **What is PRIVA?**

PRIVA is a set of libraries that allow controlling and reducing data leakage in the context of big data processing and, consequently, protecting sensible information that is processed by data analytics algorithms. PRIVA is based on anonymization techniques and anonymization policies.

Anonymization, roughly speaking, is the act of removing personal identifiers from data, for example, by converting personally identifiable information into aggregated data. Anonymized data can no longer be associated with an individual and *anonymization techniques* have been used to provide a balance between the beneficial use of data and the individual privacy. *Anonymization policies* can help in this direction. There are regulations and guidelines to standardize the use of anonymization techniques and algorithms and the implementation of such policies are considered an important progress in organizations that want to protect their customer personal data. Figure 1 overviews PRIVA. The components are ahead in this document.



**Figure 1. PRIVA overview**

**1.1 Privacy Library**

The *Privacy Library* is a set of programs that has as input the data set to be anonymized, the anonymization policy and the configuration files. The library applies the existing anonymization techniques and algorithms according to the policy. It uses the configuration files to auxiliate this techniques applications and provides, as output, the anonymized data set.

In this release (r1.0), the Privacy Library is available as a JAR (Java ARchive) file. It can be executed through *java -jar <fileName>.jar* .

**1.2. Anonymization techniques and algorithms**

The Anonymization Techniques and Algorithms refers to several anonymization techniques that can be applied on data in order to protect the privacy of individual. Release r1.0 implements the following techniques:

**Generalization**: attribute values are generalized to a range in order to reduce the granularity of representation. E.g., the date of birth may be generalized to a range such as year of birth, so as to reduce the risk of identification.

**Suppression**: the attributes (usually key attributes or quasi-identifiers) are removed completely to form the anonymized table, providing only summaries of the table data instead of individual data.

**Encryption**: it uses cryptographic schemes to replace key-attributes, quasi-identifiers and sensitive attributes for encrypted data.

**Perturbation (Masking)**: it consists of the replacement of the actual data values for dummy data. The idea is to randomly change the data to mask sensitive information.

**1.3 Input Data**

The *Input Data* represents the database whose data must be anonymized. They can be loaded from files (e.g., CSV file, JSON file) or by reading data from input streams, iterators, lists or arrays. Also, they can be imported from relational database systems. The fields are identified and must be anonymized according to the Anonymization Policy.

In the current release (r1.0), the input is done through JSON files. Once the tool is started, the user must specify the path where the file is stored (e.g., C:\\Anonymization\\input\\data.json). This file must be a valid JSON.

**1.4. Anonymization Policy**

Anonymization policy represents the guidelines to the anonymization process. It specifies the fields to be anonymized and the anonymization technique that must be applied to each field.

In release r1.0, the anonymization policy is also a JSON file. After informing the path to input data, user must specify the path where the policy is stored (e.g., C:\\Anonymization\\input\\policy.json). This file must be a valid JSON.

The policy must have the attributes FIELD\_NAME, TYPE and DETAIL, where FIELD\_NAME is the name of the database field that must be anonymized; TYPE is the anonymization technique that must be applied; DETAIL is the complement of the anonymization technique. An example of the attributes is given below.

{

"FIELD\_NAME":"NAME",

"TYPE":"SUP",

"DETAIL":"\*"

}

The TYPE attribute allow the following parameters, which refers to the anonymization techniques implemented to the release r1.0 (see section 1.2).

* “SUP” for suppression
* “MAS” for masking
* “ENC” for encryption
* “GEN” for generalization

The DETAIL attribute is required for some anonymization techniques. Table 1 shows the rules to use DETAIL in the release r1.0. Two types of generalization (GEN) can be done in release r1.0: numerical and textual. They are also explained in Table 1.

**Table 1. TYPE and DETAIL attributes relationship**

|  |  |
| --- | --- |
| **TYPE attribute** | **DETAIL attribute** |
| SUP | Any character - The tool will replace the database field with the specified character. E.g., "\*" or "?". |
| MAS | FULL, LAST or FIRST. FULL will replace the database field (string) with a same-size string. LAST will replace the last set of the database field (string) and FIRST will replace the first set of database field (string) with new strings. All these new strings will be generated randomly. |
| ENC | No DETAIL is required |
| GEN (Numeric) | Generalizations of numeric types must initiate with brackets "[" and follow the pattern:  [<*initialValue1*>-<*finalValue1*>=<*rangeName1*>;<*initialValue2*>-<*finalValue2*>=<*rangeName2*>;<*initialValueN*>-<*finalValueN*>=<r*ange\_nameN*>]  Example:  [0-13=child;14-18=teenager;19-x=adult]  When there is no specific number for latest category, the "x" must be used. |
| GEN (textual) | Textual generalizations must initiate with keys "{" and follow the pattern:  {<*parameter11*>,<*parameter12*>,<*parameter1N*>=*rangeName1*;{<*parameter21*>,<*parameter22*>,<*parameter2N*>=*rangeName2*}; {<*parameter31*>,<*parameter32*>,<*parameter3N*>=*rangeName3*  Example:  {diagnosis1, diagnosis2, diagnoses3}=Normal;{diagnoses4,diagnoses5, diagnosis6}=Alert;{diagnosis7,diagnosis8,diagnosis9}=Urgency |

Example of Anonymization policy file:

{

"FIELD\_NAME":"NAME",

"TYPE":"SUP",

"DETAIL":"\*"

},

{

"FIELD\_NAME":"AGE",

"TYPE":"GEN",

"DETAIL":"[0-13=Child;14-18=Teenager;19-x=Adult]"

},

{

"FIELD\_NAME":"MEDICAL\_TYPE",

"TYPE":"GEN",

"DETAIL":"{diagnosis1,diagnosis2,diagnosis3}=Normal;{diagnosis4, diagnosis5, diagnosis6}=Alert;{diagnosis7,diagnosis8,diagnosis9}=Urgency"

},

{

"FIELD\_NAME":"DISEASE",

"TYPE":"ENC",

"DETAIL":""

}

**1.6 Anonymized Data**

Once the anonymization process is finished, a new and anonymized file (JSON file) is generated and stored at the same directory the input was set (in the example of this document, (C:\\Anonymization\\input\\data\_anonymized.json). It will be added the suffix *\_anonymized* in the name of the file.