Privacy and security for analytics on healthcare data

Editors Albana Gaba, Yeb Havinga
Date November 7, 2014

Contributors Albana Gaba, Yeb Havinga, Henk-Jan Meijer, Evert Jan Hoijtink (Portavita)

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr. 318633.

Contents

List of Figures

List of Tables

Acronyms

CBS
Centraal Bureau Statistiek – Statistics Netherlands. 21

CHH
Community Health and Hospitals – a fictional example hospital. 31

COPD
Chronic obstructive pulmonary disease. 31

EMR
Electronic medical record. 7, 9, 23

EU
European Union. 11, 24

HbA1c
Glycated hemoglobin. 19

HCS
Healthcare Privacy and Security Classification System. 12, 15

HDL
High-density lipoprotein. 19–21

HIPAA
Health Insurance Portability and Accountability Act. 24

HIV
Human immunodeficiency virus. 13, 14, 20

HL7
Health Level 7. 6, 11, 12, 17, 32

ML
Machine learning. 23

OLA
Optimal Lattice Anonymization. 25, 28, 38

RIM
Reference Information Model. 12, 31

RIVM
Rijksinstituut voor Volksgezondheid en Milieu – Dutch intstitute for national health and environment. 21

RLS
Rowlevel Security. 22, 28, 35

TTP
Trusted third party. 20

VIP
Very Important Person. 10, 20, 22

XACML
Extensible Access Control Markup Language. 15

1. Summary

We investigated standards for the exchange of security policies and other security and privacy related restrictions to accessing healthcare data. We describe use cases that gather data during the treatment of patients, as well as use cases that need access to data for analytical purposes. These use cases were expressed using the Healthcare Classification System and Patient Consent Directive of the HL7 standards. We describe how data collected from disparate medical data sources can be accessed for analytical purposes, governed by security policies imposed by the source systems. We implemented prototypes for de-identification and access control.

2. Introduction

In a healthcare facility, such as a hospital, data is collected while treating patients. As patient data is considered sensitive by law, the custodian organisations limit access to such data to only the users that are necessary for the purpose of use. To this end, healthcare organisations author access policies based on patient consent and regulations, to establish who may access what data under what circumstances.

In the context of a European-scale data analytics scenario, patient data, originally collected in various hospitals, laboratories, and so on, is transferred to a data storage for research. As health data is originally protected by various policies in the healthcare organisation it comes from, the same protection policies must still hold in a collaborative analytics framework. The interoperability

between healthcare organisations and a collaborative analytics framework breaks the boundaries of a single department or organisation and imposes new challenges. Thus, it becomes crucial to adhere to standard ways to model the data, express various concepts and security policies, and communicate this with the organisations involved.

Our goal is to examine the security and privacy aspects of an architecture for collaborative data analytics. Ultimately, we aim at providing an architecture for safe access to patient data for data mining purposes. To this end, we look at the means currently available for protecting privacy in a standard healthcare organisation (e.g., hospital), and further, we examine mechanisms to preserve the data protection in a collaborative analytics framework.

2.1. Primary vs. secondary use of health data

The distinction between primary and secondary use of healthcare data is an initial step in the analysis of why healthcare data must be protected, and to what extent the data can be protected. Figure 2.1 presents a schematic overview.

Primary use of data is the collection and use of data for the treatment of patients. Healthcare providers that are involved in the treatment of patients need to access the EMR to assess the current status, access new results from lab tests, and update the EMR to reflect updates in the treatment plan. The data collected during treatment is used for claims and reimbursement, quality reporting, as well as research to improve future treatments, such as data analytics for clinical decision support. This use of data is called *secondary use* of healthcare data.

Compared to primary use, secondary use requires different access control and privacy protection measures. The reasons are the following:

No direct relationship with the patient In contrast with primary use, where the data users are the providers that treat the patient, with secondary use the data users have no direct treating relationship with the patient. There is no need to access the data with the purpose of improving health of the patient at hand.

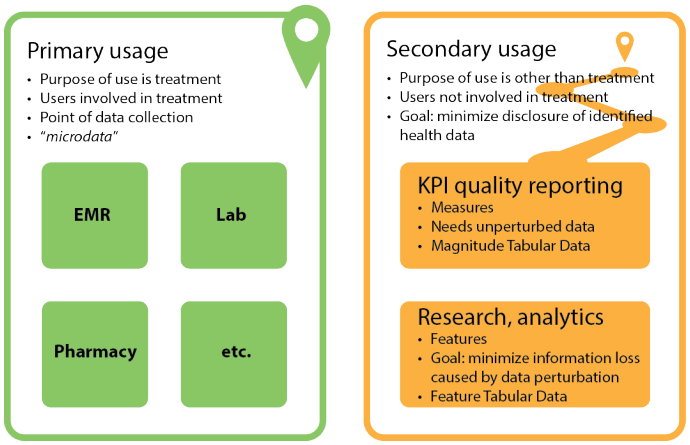

Figure 2.1.: Primary vs. secondary use of healthcare data.

Comprehensive knowledge Usually information systems for secondary use collect data from many specialized information systems that are used for the active treatment in specialized departments. This creates a comprehensive picture of all summarized treatments. While this comprehensive picture is necessary for research purposes, it also creates a higher risk of exposure when information is disclosed.

Long term data storage When time passes and the active treatment of a patient is over, there is less need to access the treatment data, and patients are less inclined to disclose their health data. Traditionally, departmental information systems would archive old data. However, advancements in economic storage allow systems for secondary use to keep keep data much longer. There is a tension between long term data storage and 'the right to be forgotten'.

To summarize, the further away from the point of care in time, and the people involved, the more protection data needs.

2.2. Data Lake for Secondary Use

To perform data analytics on a European-scale healthcare database, it is necessary to collect data from an array of medical data sources (MDS), e.g., EMR's from hospitals, into a single, integrated data storage. To this end, we consider what in the big data jargon is known as a data lake. The main difference between a data warehouse and a data lake is that in a data warehouse the data is pre-categorized at the point of entry, which can dictate how it is going to be analyzed. In contrast, in a data lake, like in the scenarios we foresee, the data can be used for a number of secondary use purposes–not necessarily known a priori, upon data collection–including analytics, reporting, accountability and so on. Hence, the data should be stored in its 'raw' state.

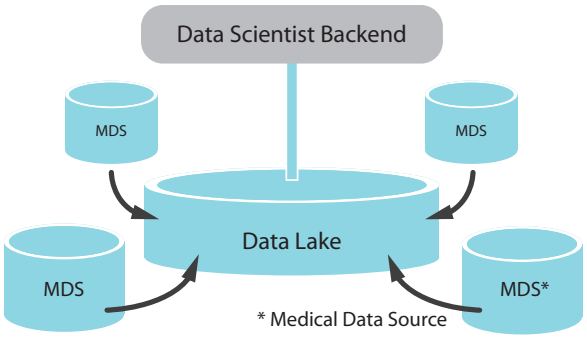

Figure 2.2.: Illustration of a framework for collaborative analytics.

As depicted in Figure 2.2, the data lake interfaces with medical source organisations that are part of the collaborative analytics framework, and with data scientists, who may employ various libraries and tools to perform analytics on the collected data. In this context, in order to provide data scientists access to health data, we identify the following challenges:

Data variety Various medical organisations may use different ways to model data and to express medical concepts. In a data lake scenario as described above, this would tremendously increase the complexity of using the data in its entirety. Adopting standard data models and vocabulary for expressing medical concepts at the medical source organisations is

crucial for facilitating data processing for secondary use in a collaborative data lake environment.

Privacy and security Every medical organisation has its own rules for determining access to patient data, which are typically based on regulations, practical use and patient will. The way data is used for primary and secondary use is quite different, and as a consequence, so are the security access policies. Since we assume that in a data lake environment the data is used only for secondary use purposes, selected policies corresponding to the allowed use should be taken into account. Again, in order to automatically compute and enforce access policies deriving from different medical data sources, standards play an important role.

Many projects In a data lake environment, it is often necessary that a multitude of projects operate on the collected data, all in different ways. For example, a project may be required to perform analytics on particularly sensitive data. In this case, it may be necessary to restrict access to such data to a very limited number of users and to increase protection barriers that aim at de-identifying patients. It is thus necessary to automate data governance, privacy protection and management for each individual project.

2.3. Restrictions for handling patient data

Medical data are sensitive data under Article 8 (1) of the Data Protection Directive [9], and as such, they are subject to a stricter data-processing regime than non-sensitive data. For one, explicit consent is required and processing is permissible only where performed by a healthcare professional subject to an obligation of professional secrecy [9]. In addition, a number of circumstances require special protection as individuals may be exposed to potential risks of financial, reputational or personal harm if their health data is made available to unauthorized individuals or entities. Think of genetic information, mental health information, the health information of children and adolescents and other sensitive information such as health records of VIPs. In this case, additional protection may be required.

In a collaborative analytics environment, which demands interoperability between different medical data sources, such security concepts should be recognized and properly interpreted by all the parties involved in data processing. This way, security policies that impose access restriction to certain health records can be correctly enforced at a data lake.

2.4. Problem Statement

The main question from a security and privacy perspective is how we can make data available for research in a data lake environment in such a way that:

1. National and EU law regarding protecting healthcare data is not violated.
2. Patient consent is not violated.
3. Organisational policies regarding access to healthcare data of the source operational data sources is not violated.

Based on the challenges described above, our contribution is focused in two directions. First, we examine the way data is modelled at individual Medical Data Sources and the privacy protection mechanisms. To this end, we refer to the standards proposed by the non-profit organisation for interoperability in healthcare, Health Level 7 (HL7). Second, we look at data protection mechanisms at a data lake environment, and transformations required on tabular data before the data is handed out to data analysts.

3. Standards for security in healthcare

In the last few years the Security Working Group of Health Level 7 has been developing Healthcare Classification System (HCS), an architecture that is suitable for "...automated privacy and security labeling and segmentation of protected health information (PHI) for privacy policy enforcement through security access control services..." [14]. HCS is centered around security labels, which are used to tag resources (e.g., clinical facts). Security access policies refer to such labels in order to determine access on labelled resources. In addition, Privacy Consent Directive Implementation Guide [15] describes a data model to record a client's health information privacy consent(s) and to exchange those privacy consent directives with other entities (i.e., custodians of the client's health records).

In the rest of this chapter we discuss in detail the main security components as introduced by HL7 and show how they fit in a data analytics environment.

3.1. Security Labels

Security Labels are meta-data that convey constraints on the use of a labelled resource. Labels are used in access control policies as attributes to express access rules when such labelled resources are requested. Security labels are grouped in the following categories:

Confidentiality is probably the most important label. It is used to indicate the degree of classification of a clinical fact. The values range from 'not restricted' to 'very restricted'. It is also included in the Hl7 version 3 RIM.

Category indicates the law that protects a clinical fact.

Integrity is used to express the reliability of the inserted data (e.g., inserted by patient, nurse, doctor, device)

Control is also known as Handling Instructions and it indicates handling caveats when clinical documents are exchanged between two or more systems. For example, the source organization may indicate that the purpose of use is `treatment` before sending out patient data to another organization. In addition, it can indicate transformation obligations to the clinical facts upon reception (e.g. de-identify). The receiving organization has to comply with the handling caveats.

For each of the categories there is a set of labels with various possible values. Table 3.1 illustrates some examples of labels for each of the categories. An important label in the scenario we are looking at is Control, Handling instructions. The medical data sources may choose to tag the health records with the following labels before transferring them to the data lake: `PurposeOfUse = HRESCH (health research), ObligationPolicy = DEID (de-identify)`. This labeling prevents the recipient organization to use the records for purposes other than health research or in a way that patients can be identifiable. A complete list of security labels can be found in the HL7 Security Label Vocabulary [13].

Category	Examples of label names	Examples of label values
Confidentiality	- confidentialityCode	- normal, restrictive, very restrictive
Category	- InformationSensitivityPolicy	- HIV, ETH,PSY
Integrity	- IntegrityConfidence	- highly reliable, uncertain reliability
	- Provenance	- clinician asserted, patient asserted, device asserted
Control	- PurposeOfUse	- treatment (TREAT), research (HRESCH), clinical research
	- ObligationPolicy	- de-identify (DEID), mask

Table 3.1.: Labelling of clinical facts

An important component of the HCS is the Security Labeling Service, which automatically labels clinical facts upon data entry or data retrieval based on Security Labelling Rules. Figure 3.1 illustrates how clinical facts are labelled. A

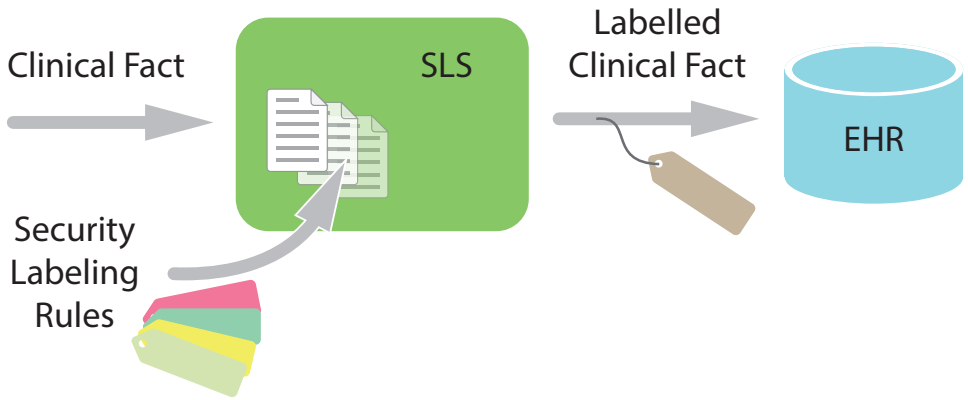

Figure 3.1.: Security Labelling Service

Security Labelling Rule typically consists of a condition and one or more labels associated with it. For example, to tag HIV-related clinical facts, a Security Labelling Rule could be expressed as follows:

```
if diagnosis=111880001 (HIV) and medication=11413 (Zidovudine)
    then Security Label Tags are:
      confidentialityCode=R (restricted) and
      InformationSensitivityPolicy=HIV
```

In addition, clinical facts can be labeled also manually, at data entry, based on patient's request or professional judgment, irrespective of the Security Labelling Rules. When a patient decides that certain clinical facts are sensitive then the following Category label may apply:

`InformationSensitivityPolicy=PRS (patient requested policy).`

This is the case when a patient feels that a treatment is highly sensitive and hence requires high access restriction.

3.2. Access Control Policies

The goal of HCS is to provide a "..standard, computable, and semantically interoperable means to apply sufficiently descriptive metadata about healthcare information so that rights of access can be established, and appropriate access control decisions can be made at each layer of security services." In the previous section we introduced security labels as a means to describe health data. Now we look at how security labels can be used to establish appropriate access control decision.

Automated access control mechanisms, such as XACML [18] and drools [5], allow to establish permissions using attributes to describe users and resources. They provide an expressive and manageable way to control access to data in face of very large databases with many roles and various resources. In simple words, a system uses policies to describe, through attributes, who can access what resources under what conditions. Each request specifies the requestor (subject), the requested resources, the action to be performed on the resources and other attributes necessary for deciding whether access can be established. When the system receives a request, it looks up for a policies that match the attributes of the request and apply the decision accordingly.

HCS integrates an access control mechanism in its design to determine access to health data. Figure 3.2 shows a simplified version of HCS, which assumes that data has been labelled at the time of entry. A request includes a number of attributes. The proxy that intercepts the request looks up for policies that have rules that match the attributes of the request. To do this, labels of the requested resources are taken into account as part of the attributes necessary for deciding data access. If authorization is granted, then the proxy will retrieve the resources and return them.

Below, we illustrate a simple XACML policy, which determines that nurses of the infectious department are allowed to insert and select sensitive information. A rule is the most elementary unit of a policy and it consists of a Target, an Effect (permit or deny) and a Condition. The Target defines the requests to which the rule applies (e.g., subjects are nurses of the infectious disease department and resources are sensitive information). A Condition may further refine the applicability of the Target (e.g., requests received during the working hours). Effect specifies whether access is granted. To enforce policies the

attributes therein are translated into queries that are then run against the database.

```
<policy>
  <rule effect=permit>
    <target>
      <subjects>
        <subject>
          <id>position</id><value>nurse</value>
          <id>department</id><value>infectious diseases</value>
        <subject>
      </subjects>
      <resources>
        <resource>sensitive information</resource>
      <resources>
      <actions>select, insert</actions>
    </target>
  </rule>
<policy>
```

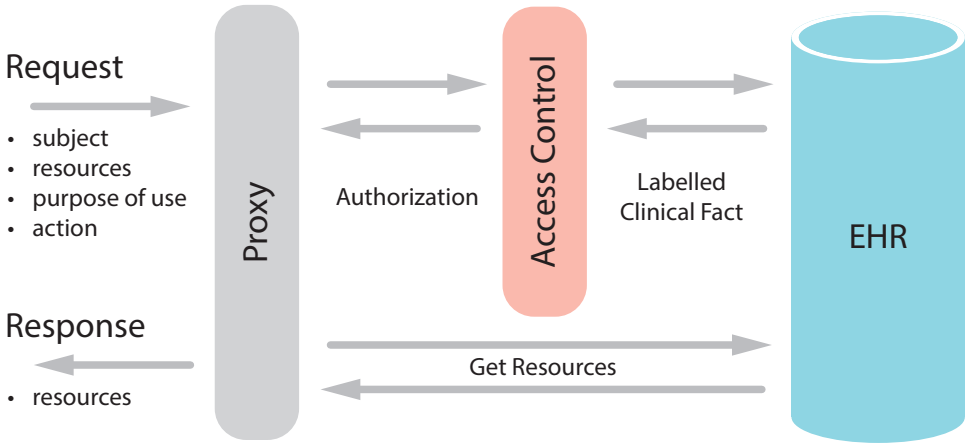

Figure 3.2.: HCS proposed design for accessing patient data within a medical data source.

3.3. Patient Consent

While the Data Protection Directive establishes that patient consent is required for processing health data, when such data is anonymised it falls outside the scope of the Directive. More specifically, in the Dutch implementation of this directive [17], patient data can be used without consent in the following two instances:

- It is not reasonably possible to ask for consent and the privacy of the patient is not unnecessarily jeopardised.
- Given the nature of the research, asking for consent is not feasible and the data arrive at the researcher in such a way that re-identification is sufficiently prevented.

The first instance is, among other things, applied to situations when the patient is deceased or, also after a reminder, does not respond to a written request asking for permission to use his or her data for a specific research. The second instance applies to situations where many patients would retroactively have to be asked for consent.

In both instances though the following conditions have to be met:

- The research serves a general interest
- The research cannot be carried out without those data
- The patient has not objected to such use of his or her data for research.

Basically, in order to use health data for research purposes, we have to make sure that i) patient data is de-identified, and ii) any patient objection to using the data for research should be taken into account.

In the next Chapter we discuss various techniques and issues related to the de-identification of health data, whereas in Appendix A we show through an example how patient consent is modelled according to the HL7 Privacy Consent Directive Implementation Guide [15].

4. Protecting Tabular Data

4.1. Introduction

Once data has been loaded into the data lake, it must be transformed into a tabular format suitable for statistical analysis. This chapter describes techniques that can be used to protect tabular data. We use the following predictive analytics use case to illustrate the protection methods.

Predict diabetic related micro-angiopathy

"Angiopathy is the generic term for a disease of the blood vessels (arteries, veins, and capillaries). The best-known and most prevalent angiopathy is diabetic angiopathy, a common complication of chronic diabetes. Chronic dysregulated blood glucose in diabetes is toxic to cells of the vascular endothelium which passively assimilate glucose. Ultimately this leads to diabetic nephropathy - where protein leakage caused by late-stage angiopathy results in diagnostic proteinuria and eventually renal failure. In diabetic retinopathy the end-result is often blindness due to irreversible retinal damage." [adapted from Wikipedia]

Given a database with records of diabetic chronic disease management, predict if a (hypothetical or real) patient will develop a form of micro-angiopathy.

To re-iterate this use case in terms of the original problem statement, the question becomes: how can data be supplied to the data analyst, in such a way that a statistical model to predict angiopathy can be build, while at the same time the restrictions, obligations and handling caveats by the source systems are applied.

4.1.1. Tabular data

The basic information structure that analytic tools such as Orange, Matlab, R and SAS operate on is tabular data. It is the task of the database system to provide access to tabular data.

Each table (data sample / data set) contains $i = 1, \ldots, n$ rows. Each row (data instance / example) has a number of input variables and one output variable. The rows usually correspond to patients, also known as subjects, individuals or record targets.

What kind of variables to prepare in the tabular data depends on the analytical use case and the machine-learning algorithm chosen. Creating these tables and removing or adding variables, is a repetitive process called *feature selection*. For instance, the initial tabular data for the micro-angiopathy use case contains variables related to:

- age
- gender
- smoking
- blood pressure
- total / HDL cholesterol
- HbA1c
- peripheral vascular disease (output variable)

These initial analytical tables form the boundary where secondary use of data starts, and this is the point where filtering and de-identification techniques must be applied.

4.1.2. Categorizing variables

One of the first steps in gaining insight in the amount of protection tabular data needs, is to categorize variables according to sensitivity and re-identification risk. We use the following categories as described by [16].

Direct Identifiers - names and uniquely identifying numbers such as social security numbers. As direct identifiers are usually not correlated to the

output variable, there is no need for direct identifiers in data sets for analytics and must thus be removed. For situations where re-identification of subjects is desirable, i.e. when analysis reveals a healthcare risk for one of the de-identified subjects, a variable containing a *pseudonym* can be added to each row. By storing the link between direct identifiers and pseudonyms at a trusted third party (TTP), this can be done in such a way that the real identity of patients is protected from the data analysts, and the confidential data (variables) are not made known to the TTP.

Quasi-identifiers - a set of variables that, in combination, can be linked with external information to re-identify (some of) the patients in the data set. Unlike direct identifiers, quasi-identifiers cannot be removed. The reason is that any variable in the data set potentially belongs to a quasi-identifier, depending on the external data sources available to the user of the data set. Thus, all variables would need to be removed to make sure that the data set no longer contains quasi-identifiers.

Confidential variables - these variables contain sensitive information. All variables related to healthcare are considered confidential, but some variables are more sensitive than others. Goldstein et al. [11] describe the following categories of highly sensitive data: mental health, data regarding minors, intimate partner violence and sexual violence, genetic information and HIV related information. In addition, also data regarding VIPs is highly sensitive.

Non-confidential variables - these variables contain non sensitive data about the patients, such as town and country of residence. Note that these variables cannot be neglected when protecting a data set, because they can be part of a quasi-identifier. For instance, 'Job' and 'Town of residence' can be considered non-confidential outcome variables, but their combination can be a quasi-identifier, because everybody knows who is the doctor in a small village.

The use case to predict micro-angiopathy contains no direct identifiers. Age, smoking and gender are quasi-identifiers, as these values are likely to appear in other data sets as well, or can be otherwise linked to the patient through external available information. This is usually not the case for blood pressure and total / HDL cholesterol measurements. Typically, a series of recurring blood pressure or HDL cholesterol measurements of a certain period will be summarized with statistical aggregates, such as average, standard deviation

and trend, over a certain period specific for the specific research project at hand. Therefore blood pressure and total / HDL cholesterol are not classified as part of a quasi-identifier and are 'just' confidential variables. To summarize, the variables from the micro-angiopathy use case are classified as follows:

- *Quasi-identifier*: age, smoking and gender.
- *Confidential variables*: blood pressure and total / HDL cholesterol

4.2. Preparing Tabular Data

4.2.1. Disclosure Scenario

To decide which disclosure control method is appropriate for the micro-angiopathy data set, it is useful to consider disclosure scenarios. Suppose there exists an external archive, that contains one or more variables from the quasi-identifier *age, smoking and gender*, together with direct identifiers such as *name* and *SSN*, this archive can be used to link subjects in the protected set with identifiers. The question is, whether such an external archive is available about the same subjects from the protected data set. Publicly available datasets such as CBS Opendata [3], consist of magnitude tabular data, such as contingency tables that list the proportion of female smokers in the age group 40-50. Since these kinds of tables do not contain direct identifiers, the data cannot be linked to individuals, unless the magnitude in a cell is very small, e.g., if there is only one female smoker in the age group >100 in the state Utrecht. This is not the case for the CBS Opendata tables. The data collected for the 'Healthmonitor GGD'en, CBS and RIVM' does contain the key data from the micro-angiopathy quasi-identifier, but again no direct identifiers are publicly available. It is possible, that the data analysts that prepare the Healthmonitor datasets have access to age, smoking and gender, combined with direct identifiers. Therefore there is a disclosure risk of the confidential variables in the micro-angiopathy dataset, only if the dataset will be directly or indirectly accessible to the same organization. In this scenario, it is possible to re-identify a subject when the values in the quasi-identifier are 'rare' in the population. The definition of rare requires a threshold to be chosen for each combination of variables from the quasi-identifiers (key). A key is safe when it

occurs more often than the threshold. Hundepool et al. [16] describe methods to calculate the disclosure risk for categorical, continuous and combinations of categorical and continuous quasi-identifiers. See also section 4.2.4.

4.2.2. Cohort selection

The task of cohort selection is to select which patients we want to include in the data analysis. This phase in the analytics project overlaps with quality measure projects, where there is also a need to define which patients do, or do not count, for a certain measure. Examples of cohorts are:

```
patients with HbA1c >50 mmol/mol
    and have not had a yearly diabetic check up in the last year.

patients that are younger than 80
   and have a LDL measurement in the last year with a value > 2,5
   and didn't use lipid lowering medications in the last year
   and doesn't use statins currently
   is currently under treatment for CardioVascular Risk Management
```

Excluding restricted data segments

The kind of filtering in the cohort selection phase matches the kind of filtering needed for the following protection measures:

- Exclude specific segments of data from patients that have opted out for research, as described in Section3.3. For instance, if a patient has opted out for research of data related to mental health treatment, records pertaining to mental health must be excluded.
- Exclude data from VIPs. How clinical data is security labeled with VIP codes is described in the RLS prototype scenario 3 in appendix A.2.

These patients and their observations can be filtered from the resulting tabular data with Row Level Security. A prototype healthcare database to test RLS is described in appendix A.

4.2.3. Feature selection

Medical Data sources, such as EMR's and laboratory systems, register many different kinds of variables. For a specific healthcare analytic project, only a subset of the available variables will be relevant or useful. Also the format in which data is recorded is often not immediately useful for a specific ML algorithm. Feature selection/transformation is the task to select the relevant variables and process them to be appropriate input for the ML algorithm. Feature selection [12] is a repetitive task with the data analyst in the loop. An initial investigation wherein besides the data analyst, also a domain expert (medical doctor) is involved, is useful to perform an initial inclusion or exclusion of variables.

Example tasks of feature selection are:

- Convert physical quantity observations to canonical values, so they are comparable as numeric numbers without unit.
- Rank variables.
- Normalize variables if they are not commensurate.
- Select a subset of variables (forward/stepwise addition vs backward selection/pruning).
- Test statistical model performance of a subset of variables.
- Convert recurring event data into summaries.

4.2.4. Protecting quasi-identifiers

Each time in the feature selection phase, when a different set of variables are selected to build a statistical model for, the re-identification risk can be assessed and appropriate measures must be taken, before the tabular data can be made available.

1. Determine which variables form a quasi-identifier.
2. Determine if the data user has access to external data where data with the same key values are linked to identifiable data.
3. Take appropriate measures to protect the quasi-identifier. Safe harbor is safe enough for the micro-angiopathy use case, but this is not true

for the general case. See [16] for a thorough reference of techniques to assess risk and techniques to protect data.

There are a number of methods to protect quasi-identifiers, that can be categorized in non-perturbative and perturbative methods. Drawback of all protection methods is that there is loss of information. While methods exist to calculate information loss [16] and definitions are given to decide whether a protected data set is 'analytically valid', the calculations are performed on the columns of the data sets, rather than the space spanned by the vectors (rows).

Safe Harbor De-identification

A well known de-identification technique is called Safe Harbor De-identification [4]. Safe Harbor de-identification is a fixed set of rules; it does not base the data transformation on the frequencies of key values in the quasi-identifiers. Consequently, Safe Harbor is only safe to use on data sets where the key values from the quasi-identifiers have high population frequencies, and a result, low re-identification risks. See [7] for a discussion on the limitations of using Safe Harbor.

The following rule from Safe Harbor applies to the data set of the micro-angiopathy use case:

- All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older.

The benefit of safe harbor for the micro-angiopathy data set is three fold:

1. Safe harbor is a de-identification technique approved by the US HIPAA legislation. As EU and Dutch law do not prescribe a specific de-identification algorithm, but rather requirements of any de-identification algorithm, the same statement cannot be made for EU and Dutch law in the general case.

2. It is easy to implement and validate. Complexity of the algorithm is $O(n)$ on the number of rows in the table.
3. There is no information loss on values other than the birth time, which is generalized to the birth year. Arguably this information loss does not impair analytical value to predict angiopathy, or in fact all medical conditions other than conditions that occur in infants or children.

Other de-identification algorithms

Hundepool et al. [16] describe a number of de-identification methods, algorithms and software, divided into non-perturbative (suppression, global recoding, sampling) and perturbative methods (adding noise, micro aggregation, data swapping, data shuffling and rounding). Of these methods, the generalization-based algorithms, global recoding for categorical data and micro aggregation for continuous data, cause the least loss of information for analytics on healthcare data, as they target outliers and perturb categories and values, so the individual outliers are replaced by representative values of a group of k values, to achieve k-anonymity. There exist several algorithms for recoding and micro aggregation, and commercial as well as open source toolkits exist that implement these [10].

In the context of the AXLE project, using a component separate from the database to perform de-identification would be infeasible, since the dataset would be too large to move to this component. Therefore, such an algorithm could only be implemented at database-level. No currently available open implementations were found that could easily be re-used for this purpose. To investigate the feasibility of an implementation at database-level, a prototype implementation of Optimal Lattice Anonymization (OLA) was developed as described in Appendix B. OLA achieves k-anonymity by applying a combination of generalization and suppression, a technique very commonly used in practice [6, 8]. Within the family of algorithms utilizing these techniques, OLA guarantees achieving k-anonymity with a minimal amount of information loss, hence preserving analytical value in the best way possible.

Drawback of this sequential implementation of OLA is high computational complexity, exponential in the number of key values in the quasi-identifier. As Hundepool et al. [16] indicate, the lowest computational complexity of useful

micro aggregation (multi-variate heuristic micro aggregation) is $O(n^2)$ in the number of rows. A full investigation into developing data parallel algorithms for de-identification of big data sets is a large project and outside of the scope of the AXLE project.

Estimating re-identification risk

There are a number of methods to calculate re-identification risks for quasi-identifiers that contain categorical, continuous data, or a mix of both. All methods are based on quantifying per-record rarity of the values in the quasi-identifier attributes [16]. Though most re-identification techniques are intended to calculate risk before SDC methods are applied, assessing re-identification risk is also useful to estimate whether a table de-identified with Safe Harbor is safe to release. Methods to measure the re-identification risk also differ in whether the risk of re-identification in the sample (data set) is estimated (prosecutor risk), or re-identification in the whole population (journalist risk).

To estimate re-identification risk of the micro-angiopathy data set we use the following method from [6]. The probability that record i is correctly identified is denoted θ_i, where $i \in 1, \ldots, n$ and n is the number of records in the table. Let J be the set of equivalence classes in the data set containing only the quasi-identifier values. Since all of the records in the same equivalence class will have the same probability θ_i, we will refer to the probability θ_j for an equivalence class where $j \in J$. For the micro-angiopathy data set we will use the metric for θ_j associated with the prosecutor risk, which assumes that the attacker knows that the victim is in the data set. For the prosecutor risk, $\theta_j = 1/f_j$ where f_j is the size of the equivalence class j in the data set. Given a pre-defined threshold τ, we can now calculate the proportion R of records that have a re-identification probability higher than this threshold:

$$R = \frac{1}{n} \sum_{j \in J} f_j \times I(\theta_j > \tau)$$

where I is the indicator function returning 1 if the argument is true and false otherwise.

Equivalence classes of the quasi-identifier and the corresponding values for θ_j were calculated on the micro-angiopathy data set. The data set contains $n \approx 33000$ rows, and the most rare quasi-identifier has a frequency $f_k = 13$ with associated risk $\theta_j \approx 0.077$. The amount of records at risk depends on the value we choose for the threshold τ. [6] cites the value 0.05 as the low end of threshold risks seen in current practice. Using this value, 2.6% of the records in the micro-angiopathy are risk for the prosecutor.

Is this an acceptable risk? The answer depends on the dissemination level of the data set. For public dissemination, the risk might be considered too high. Conversely, if the data set is made available to the data analyst with additional mitigating controls, such as remote execution and accepting license requirements that prohibit downloading and re-identification, the risk can be considered low. Another cause for the seemingly high risk, is the assumption made by the prosecutor risk measure, that the attacker knows whether an individual is a member of the disclosed data set. The following scenario shows that this is not too far fetched. Suppose that the prosecutor knows that the data set contains data of only diabetic patients of a single healthcare organization. The specific healthcare organization could be inferred from the number of patients listed in the organization's annual report, if that number coincides with the size of the data set. Related to a healthcare organization is the region it serves, and consequently, if the prosecutor knows that somebody has diabetes and lives in the region of the healthcare organization, there is a high probability that the person is in the data set. In this scenario the prosecutor risk is an appropriate measurement to calculate re-identification risk.

This re-identification risk was calculated on a data set de-identified with Safe Harbor. For data sets where the risk after application of Safe Harbor is decided to be to high, an alternative de-identification technique must be used. See section 4.2.4 and the OLA prototype described in appendix B for more information.

5. Conclusions

We have described the environment in which healthcare organizations must find a balance between recording confidential healthcare data in databases, and providing access to that data for primary and secondary use, where both usability and privacy are maximised.

This report describes an architecture for a secure healthcare data lake, that uses the security policies authored by the medical data sources, including patient consent, to provide limited access for secondary use to data analysts. We have shown how these source policies can be enforced on analytical tabular data. Enforcement is implemented by adding additional filter clauses in the cohort selection phase, as well as by posing Safe Harbor restrictions on variables selected in the feature selection phase.

Prototypes of protecting healthcare data with RLS, and de-identification with OLA have been made.

Appendix

Appendix A.

Prototype of Row Level Security in a healthcare database

We illustrate with an example how labelled health data can be safely accessed based on security polices for analytics purposes. To this end, we adopt a standard data model for healthcare data and use PostgreSQL Row Level Security for expressing and enforcing access policies at the database level.

A.1. Representation of health data

Performing analytics on health data that derives from different organziations and countries requires a common way to model data and express domain-specific concepts. We use the HL7 version 3 RIM (HL7v3RIM), which has been developed by the organization for standards and interoperability in healthcare, Health Level 7, and is widely adopted among healthcare organizations. In HL7v3RIM there are six high-level concepts to describe all clinical data: entities, roles, acts, participations, role links, and act relationships, as shown in Figure A.1. There are several specializations of main classes.

Entities can be organizations or also persons. A role can be played by an entity and is scoped in another entity. For example, patient is a role that is played by a person (an entity) and is scoped by an organization (also an entity). Acts describe events. For example, observations and examinations are acts. Role participate in acts. For example, a patient role can participate as a subject in an observation and a practitioner role can participate as a performer in an observation.

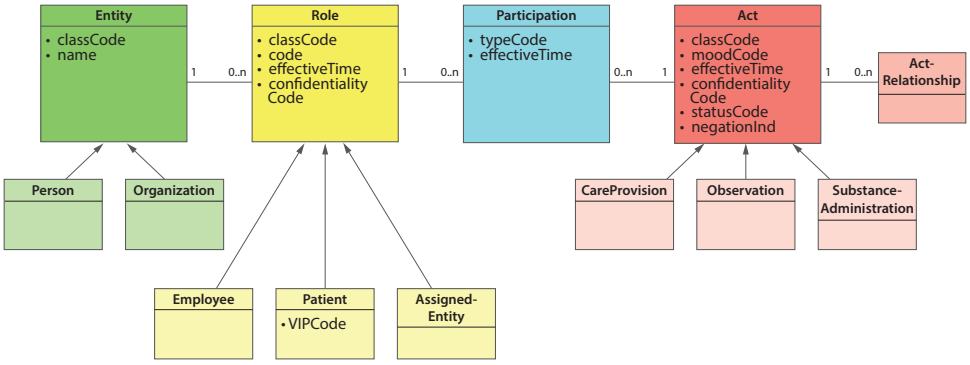

Figure A.1.: RIM

A.2. Use Cases

We create a small RIM database in order to represent a number of scenarios. Further, we show a couple of security policies and use Row Level Security to enforce them upon data access request.

Scenario 1 Mary White has been diagnosed with diabetes type II and is being treated at Community Health and Hospitals (CHH). Dr. Pete Zuckerman, employed at the CHH, is the treating physician for diabetes of Mary White (AssignedEntity). In the context of her diabetes treatment, Mary White has her blood pressure measured by Dr. Pete.

Scenario 2 Mary White is also being treated for Chronic Obstructive Pulmonary Disease (COPD) at CHH. Her principal physician for this treatment is Dr. Ronan Lang.

Scenario 3 Isabella Jones is a celebrity. She is being treated for diabetes at CHH by Dr. Pete Zuckerman. CHH has a labelling rule that each record attributed to vip patients is labelled with `confidentialityCode = v`.

Figure A.2 illustrates *Scenario 1* as a representation in a RIM database. Mary White is an Entity that plays the role of Patient and participates as a Receiver in the Observation that represents the act of measuring the Systolic Blood Pressure. On the other hand, Dr. Pete Lang participates in the same Observation as a Performer and has the Role of Assigned Entity within the organization

Community Health Hospital. The Act Relationship that connects this Observation with the CareProvision represents the fact that the Observation is done in the context of a Diabetes treatment.

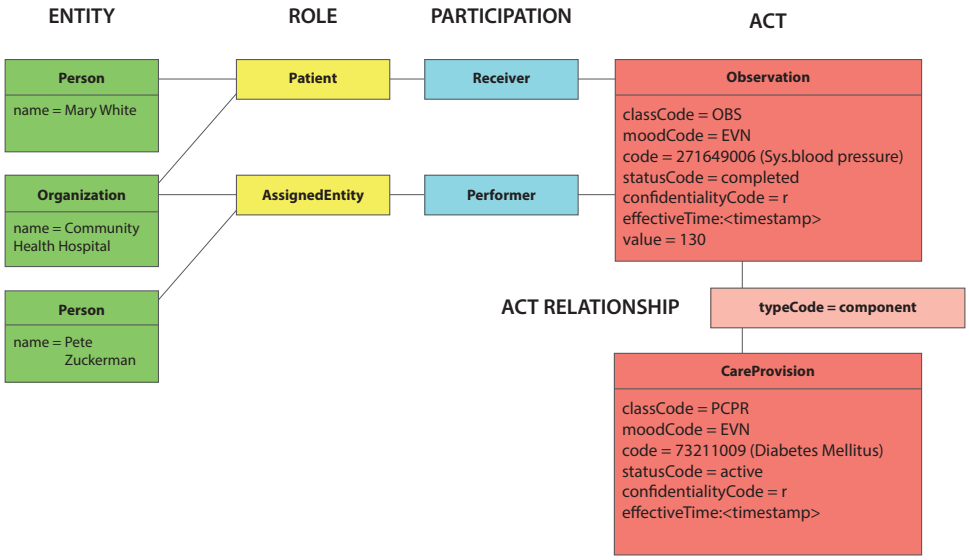

Figure A.2.: Representation of Scenario 1 in a RIM database.

While the CHH may have a vast number of security access control policies, we only consider security policies that are targeted for research purposes. In the standard HL7 there is a code to specify purpose of use.

Security Policy 1 Mary White issues a consent opt-out for research purposes regarding records collected throughout her diabetes treatment.

Security Policy 2 The Community Health Hospitals has a policy that restricts access to records labelled with `confidentialityCode = v` only to the treating physicians. The use for research purposes is also not allowed.

Figure A.3 shows the representation of a consent opt-out according to the HL7 Consent Directive Guide [15]. It is important to note the codes used and the HL7 classes involved, which are specific to model a consent.

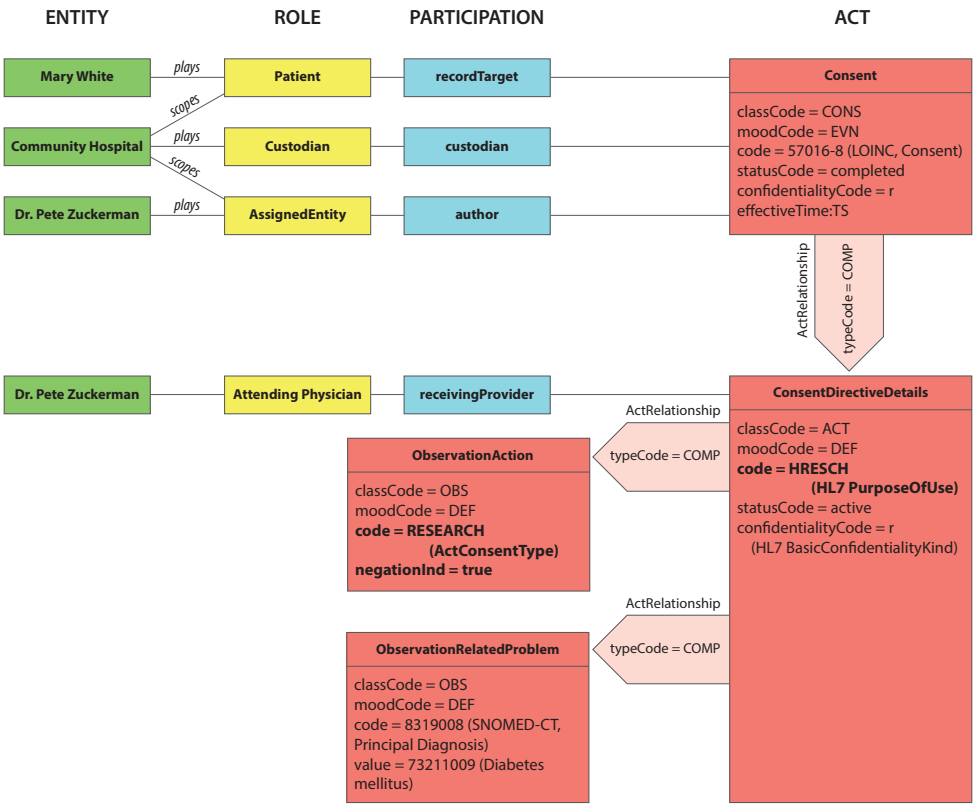

Figure A.3.: Representation of a patient consent opt out as described in the Security Policy 1.

The *Security Policy 2* may be expressed through XACML notation as shown below. The policy is composed by two rules, the first denying access to 'very restricted information' to all subjects, the second allows access to treating physicians, and finally a third one to explicitly deny access for research use. The rule-combining option `permit-overrides` at the policy level makes sure that permit is granted to subjects where at least one permit rule exists. Each of the attributes in the policy is resolved into a query by the XACML PIP component.

```
<policy ruleCombiningAlgId=permit-overrides>
  <rule effect=deny>
```

```
    <target>
      <resources>
        <resource>very restricted information</resource>
      <resources>
      <actions>select, insert</actions>
    </target>
  </rule>
  <rule effect=permit>
    <target>
      <subjects>
        <subject>
          <id>position</id><value>treating physician</value>
        <subject>
      </subjects>
      <resources>
        <resource>very restricted information</resource>
      <resources>
      <actions>select, insert</actions>
    </target>
  </rule>
  <rule effect=deny>
    <target>
      <resources>
        <resource>very restricted information</resource>
      <resources>
      <actions>select, insert</actions>
    </target>
    <condition><id>purposeOfUse</id><value>research</value></condition>
  </rule>
<policy>
```

A.3. Access control with RLS

To use health data for research purposes in a data lake environment it is necessary to comply with organization policies and patient's consent denials, which may require determined records to be left out. As data arrives from different organizations, we assume that medical data sources transfer all patient data to a data lake, along with security policies specific for secondary purposes, expressed in a standard representation, such as XACML. In order to safely access data in accordance with MDS policies, we adopt PostgreSQL

Row Level Security, which provide much faster authorization and data access compared to XACML.

Row Level Security is a security feature of PostgreSQL implemented in the context of AXLE WP 2.1. It allows one to define policies and enforce them upon request for data operation. RLS policies are applied to database tables in order to grant access to a subset of rows under certain conditions.

```
ALTER TABLE <name> ENABLE ROW LEVEL SECURITY;

CREATE POLICY <name> ON <table>
    [ FOR { ALL | SELECT | INSERT | UPDATE | DELETE } ]
    [ TO { PUBLIC | <role> [, <role> ] } ]
    USING (<condition>)
```

RLS policies apply to individual tables. When RLS is enabled on a table, records of that table can only be accessed when the RLS `condition` is true. `Condition`–the core of a RLS policy–is a query that returns a boolelan value.

The security policies we have considered determine opt-out rules based on security labels associated to clinical facts. Moreover, patient consent opt-out is based on the clinical treatment (Care Provision). To simplify the condition query we use auxiliary tables where opt-out policies are expressed in terms of attributes, such as security labels and care provision.

As depicted in Figure A.4, table *ExcludeClinicalFactsPolicies* has for each organization information about labels that should be left out. For the Patient table, only the value of `confidentialityCode` is used since no security labels apply at the Role classes. Finally, *OptOutConsent*, contains information about the care provision of patients who have objected to have their data used for research purposes.

For example, Figure A.5 shows the representation of the Security Policy 2, while patient consent opt-out scenario is represented in Figure A.6.

In a RIM database, the tables with confidential information are Role (Patient) and Act classes with their subclasses. We show below an example of the RLS policies for the two Security Policies introduced above.

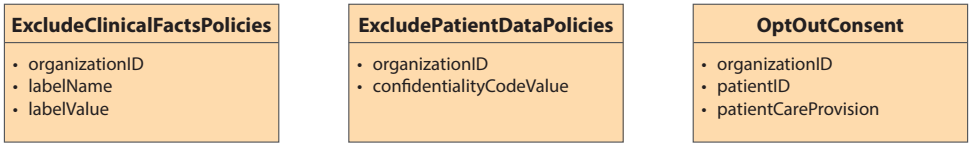

Figure A.4.: Exclusion policies

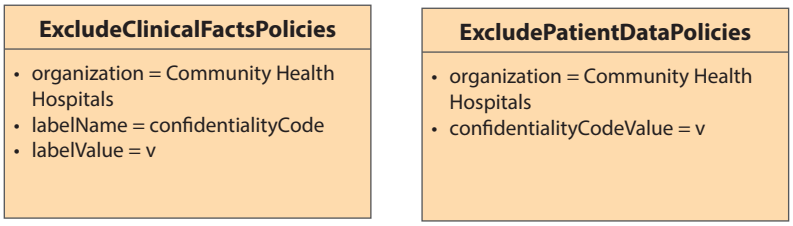

Figure A.5.: Scenarios of exclusion of labelled resources. Both for clinical facts and patient table.

```
ALTER TABLE Act ENABLE ROW LEVEL SECURITY;
ALTER TABLE Patient ENABLE ROW LEVEL SECURITY;

CREATE POLICY p1 ON Act FOR ALL
    USING (
        confidentialityCode not in
            (SELECT policy.labelValue FROM ExcludeClinicalFactsPolicies policy
                WHERE
                    policy.organizationID = _org_id
                AND policy.labelName = 'confidentialityCode')
        AND
            not(_care_provision && array(
                SELECT consent.patientCareProvision FROM OptOutConsent consent
                WHERE consent.patientID = _patient_id)));


CREATE POLICY p2 ON Patient FOR ALL
    USING (
        confidentialityCode not in
            (SELECT policy.confidentialityCodeValue
             FROM ExcludePatientDataPolicies policy
             WHERE policy.organizationID = _scoper));
```

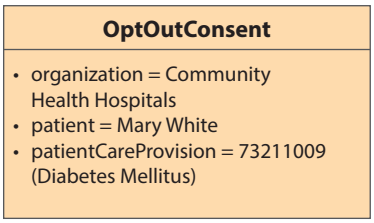

Figure A.6.: Scenarios of patient consent with opt out from research for records of a certain care provision.

The first policy applies on table Act and combines both Security Policies introduced above. In fact the first part prevents access to records labeled according to exclusion policies defined in table *ExcludeClinicalFactsPolicies*, while the second part prevents access to records with consent denials as specified in table *OptOutConsent*. We omit `<role>`, which implies that the policy applies to all roles (`<public>`). The `condition` has a boolean expression that is evaluated upon the request to access table Act through a select, insert, update or delete operation. In table Act we have included shortcuts to additional information needed by the RLS policies, such as Organization (`_org_id`), Patient (`_patient_id`), and CareProvision (`_care_provision`). Similarly, the second RLS policy applies on table Patient and prevents access to patients with a `confidentialityCode` as defined in table *ExcludePatientDataPolicies*.

A complete prototype of the RLS security policies considered in this Appendix can be found on the AXLE github page [1].

Appendix B.

De-identification using Optimal Lattice Anonymization

Optimal Lattice Anonymization (OLA, [8]) is an example of a generalization-based algorithm, a reference implementation of which was created in the context of the AXLE project. The algorithms combines generalization of quasi-identifiers and suppression of outlying data vectors to achieve k-anonymity while minimizing information loss within a data set.

B.1. Algorithm description

In this section we will describe the OLA algorithm. For this description, an important concept is the concept of *generalization trees*. Generalization is used to decrease the number of unique values for a quasi-identifier in the data set. After generalization, a specific value for this quasi-identifier taken from the data set is shared by multiple rows in the data set, hence lowering the risk of re-identification of a single row. The number of unique values after generalization will be lower for greater extents of generalization. We can order different levels of generalization in hierarchies and we can obtain a higher generalization level by taking the union of two values as a new value. For example, we can combine the two ranges $[0, 5)$ and $[5, 10)$ into the single (more general) range $[0, 10)$, which can be considered a parent in the generalization tree. The OLA algorithm requires us to define such generalization trees in advance for each quasi-identifier that we intend to generalize. The algorithm

aims to find an optimal balance in choosing generalization levels for each quasi-identifier.

All possible combinations of generalization levels are evaluated against a single qualifier: the amount of suppression needed in order for the data set to satisfy the k-anonymity constraint. The value of k needs to be chosen beforehand. A fixed threshold for suppression amount is also chosen up front and all generalization combinations yielding a suppression amount above this threshold are discarded as candidates. Of the remaining combinations, the combination requiring the lowest amount of generalization (i.e. the lowest generalization levels on all quasi-identifiers combined) is chosen as the best solution. This solution is assumed to maximally preserve statistically relevant characteristics (hence lowest information loss). Generalization is applied following that combination and a minimal number of rows is suppressed in order to achieve k-anonymity.

We can describe the steps of the algorithm in natural language as follows:

Before starting the algorithm:

- Choose desired k-anonymity level
- Choose maximum suppression threshold (d_{max})
- Build generalization trees for all quasi-identifiers

Next, the following steps are taken to evaluate all possible combinations of generalization levels:

1. Create a lattice using combinations of generalization levels as nodes and consider nodes connected when only one quasi-identifier differs between them by not more than a single generalization level
2. As long as non-evaluated nodes exist, pick one of them and apply the following steps:
    a) Generalize the data according to the generalization levels for this specific node.
    b) Determine the amount of rows that need to be suppressed to reach the desired level of anonymity (d).
    c) If the required suppression rate is smaller or equal than the maximum suppression rate ($d <= d_{max}$), this specific combination of generalization levels can be considered *acceptable*. If the required

suppression rate is higher than the maximum suppression rate ($d > d_{max}$), this combination of generalization levels is discarded.

d) By extension, all nodes describing generalization levels equal or higher in the tree than the evaluated node can be considered acceptable (and marked as such) in this context as well and thus need no separate evaluation. Similarly, if this node is discarded, all nodes describing generalization levels lower that the evaluated node can be discarded as well.

3. For all acceptable nodes calculate the information loss by comparing the de-identified data set with the original one according to a specific information loss metric. Our reference implementation uses the Samarati approach of selecting the node with the least amount of generalization [19].

Finally, the data set can be appropriately de-identified:

4. Apply to the data set the generalization levels described by the acceptable node with the least amount of information loss
5. Suppress the minimum number of needed to achieve k-anonymity

When the algorithm is finished, the data set satisfies the k-anonymity constraint and the maximum suppression constraint (d_{max}) with the least amount of generalization.

B.2. Implementation

A prototype implementation of the OLA algorithm was made and published on the AXLE github page [2].

B.3. Evaluation

Both increasing the number of generalization levels as well as increasing the number of quasi-identifiers will cause an exponential growth on the number of nodes (combinations) in the lattice. Even with the optimizations provided by the algorithm, search space for the optimal combination will be quite

large, which makes it difficult to evaluate in reasonable time with standard hardware. This was confirmed by the reference implementation: on a recent desktop system (Intel Xeon X3430 at 2.40GHz, 7GB of RAM), de-identifying a data set containing a reasonable number of 9 quasi-identifiers and 2 to 6 generalization levels for each of those, took more than a week to complete. However, considering the usefulness of the algorithm in finding an optimal solution, in some cases creating a faster implementation and using better hardware might be warranted.

Bibliography

[1] AXLE project GitHub page: Access Control. https://github.com/AXLEproject/axle-access-control, 2014.

[2] AXLE project GitHub page: OLA prototype. https://github.com/AXLEproject/axle-ola-prototype, 2014.

[3] CBS Open Data StatLine databank. http://www.cbs.nl/nl-NL/menu/cijfers/statline/open-data/default.htm, 2014.

[4] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/, 1996.

[5] Drools Business Rules Management System. http://drools.org, 2014.

[6] K. El Emam. *Guide to the De-Identification of Personal Health Information*. Taylor & Francis, 2013.

[7] K. El Emam. *Risky Business: Sharing Health Data While Protecting Privacy*. Trafford Publishing, 2013.

[8] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association*, 16(5), 2009.

[9] European Parliament and Council of the European Union. Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Online at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML, 1995.

[10] R. Fraser and D. Willison. Tools for de-identification of personal health information. Online at http://www.ehealthinformation.ca.php54-2.ord1-1.websitetestlink.com/wp-content/uploads/2014/08/2009-Tools-for-De-Identification-of-Personal-Health.pdf, 2009.

[11] M. Goldstein, A. Rein, P. Hughes, B. Williams, S. Weinstein, and M. Heesters. Data segmentation in electronic health information exchange: Policy considerations and analysis. Technical report, Prepared for the Office of the National Coordinator for Health IT, U.S. Department of Health and Human Services, 2011. Online at http://publichealth.gwu.edu/departments/healthpolicy/DHP_Publications/pub_uploads/dhpPublication_168F948B-5056-9D20-3D2C53BEED88834B.pdf.

[12] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[13] Health Level Seven International. HL7 Healthcare Privacy and Security Classification System Security Observation Vocabulary, 2013.

[14] Health Level Seven International. HL7 Healthcare Privacy and Security Classification System (HCS). Online at http://www.hl7.org/implement/standards/product_brief.cfm?product_id=345, 2014.

[15] Health Level Seven International. HL7 Implementation Guide for CDA® Release 2: Privacy Consent Directives, Release 1. Online at http://www.hl7.org/implement/standards/product_brief.cfm?product_id=280, 2014.

[16] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. S. Nordholt, K. Spicer, and P.-P. de Wolf. *Statistical Disclosure Control*. Wiley, 2012.

[17] Ministerie van Veiligheid en Justitie. Burgerlijk Wetboek Boek 7, Artikel 458. Online at http://wetten.overheid.nl/BWBR0005290/Boek7/Titel7/Afdeling5/Artikel458, 2014.

[18] OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0. Online at http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html, 2013.

[19] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, nov 2001.