Western Graduate&PostdoctoralStudies

Western University

Scholarship@Western

Electronic Thesis and Dissertation Repository

12-14-2018 10:30 AM

# Secured Data Masking Framework and Technique for Preserving Privacy in a Business Intelligence Analytics Platform

Osama Ali, *The University of Western Ontario*

Supervisor: Dr. Abdelkader Ouda, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering

© Osama Ali 2018

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Business Analytics Commons, Business Intelligence Commons, Computational Engineering Commons, Computer and Systems Architecture Commons, Databases and Information Systems Commons, Data Storage Systems Commons, Information Security Commons, Management Information Systems Commons, and the Software Engineering Commons

## Recommended Citation

Ali, Osama, "Secured Data Masking Framework and Technique for Preserving Privacy in a Business Intelligence Analytics Platform" (2018). *Electronic Thesis and Dissertation Repository*. 5995.
https://ir.lib.uwo.ca/etd/5995

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

# Abstract

The main concept behind business intelligence (BI) is how to use integrated data across different business systems within an enterprise to make strategic decisions. It is difficult to map internal and external BI's users to subsets of the enterprise's data warehouse (DW), resulting that protecting the privacy of this data while maintaining its utility is a challenging task. Today, such DW systems constitute one of the most serious privacy breach threats that an enterprise might face when many internal users of different security levels have access to BI components. This thesis proposes a data masking framework (iMaskU: Identify, Map, Apply, Sign, Keep testing, Utilize) for a BI platform to protect the data at rest, preserve the data format, and maintain the data utility on-the-fly querying level. A new reversible data masking technique (COntent BAsed Data masking - COBAD) is developed as an implementation of iMaskU. The masking algorithm in COBAD is based on the statistical content of the extracted dataset, so that, the masked data cannot be linked with specific individuals or be re-identified by any means.

The strength of the re-identification risk factor for the COBAD technique has been computed using a supercomputer where, three security scheme/attacking methods are considered, a) the brute force attack, needs, on average, 55 years to crack the key of each record; b) the dictionary attack, needs 231 days to crack the same key for the entire extracted dataset (containing 50,000 records), c) a data linkage attack, the re-identification risk is very low when the common linked attributes are used. The performance validation of COBAD masking technique has been conducted. A database schema of 1GB is used in TPC-H decision support benchmark. The performance evaluation for the execution time of the selected TPC-H queries presented that the COBAD speed results are much better than AES128 and 3DES encryption. Theoretical and experimental results show that the proposed solution provides a reasonable trade-off between data security and the utility of re-identified data.

## Keywords

Business Intelligence; Data Masking; Software Framework; Data Warehouse; Anonymization; Data Security and Privacy; Health Data.

# Acknowledgments

First of all, I thank God for giving me the ability to achieve my goals. Secondly, I would like to thank my lovely mother for her daily supplication for me, wishing me all the best in my entire life. Thirdly, sincere thanks to my lovely family (my wife and my two sons and daughter) for their patience, support and encouragement in my life.

I would like to express my deep appreciation to my supervisor Dr. Abdelkader Ouda for his insightful guidance, invaluable advice, and constructive criticism during my Ph.D. program. His academic expertise helped me to improve my research skills. His support and motivation gave me the confidence and the strength to accomplish my goals.

I would like to thank my advisory committee, Prof. Roy Eagleson and Dr. Abderhaman Quazi for their encouragement and valuable suggestions on tracking my research progress. Thanks to Electrical and Computer Engineering department at Western Engineering faculty for their kind supports. Furthermore, I am extending my gratitude to the examiners' time and efforts to be part of this journey and make this thesis more reliable and authentic.

I acknowledge my workplace supervisor Mr. Pete Crvenkovski at the Erie St. Clair LHIN and my colleague at the Decision Support Department for their support and advice during my Ph.D. program.

# Table of Contents

# List of Tables

# List of Figures

# List of Appendices

# List of Abbreviations

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **API** | Application Program Interface |
| **BI** | Business Intelligence |
| **CIHI** | Canadian Institute for Health Information |
| **COBAD** | COntent BAsed Data masking |
| **CCI** | Classification Code of health-related Interventions |
| **CCRS** | Continuing Care Reporting System |
| **CSD** | Census Sub Division |
| **DM** | Data Masking |
| **DW** | Data Warehouse |
| **DES** | Data Encryption Standard |
| **DAD** | Discharge Abstract Database |
| **DBMS** | Data Base Management System |
| **DDM** | Dynamic data masking |
| **ICD-10** | International statistical Classification of Diseases - 10th revision |
| **EDW** | Enterprise Data Warehouse |
| **ETL** | Extract, Transform, and Load |
| **iMaskU** | Identify, Map, Apply, Sign, Keep-test, Utilize |
| **HN** | Health Card Number |
| **$K_{M}$** | Masking signature Key |
| **$K_{SH}$** | Shuffle Key |
| **FPE** | Format-preserving Encryption |
| **SIN** | Social Insurance Number |
| **LHIN** | Local Health Integration Network |
| **NACRS** | National Ambulatory Care Reporting System |
| **NRS** | National Rehabilitation Reporting System |
| **MIPS** | Million Instructions Per Second |
| **SSAS** | SQL Server Analysis Services |

| | |
|---|---|
| **SSIS** | SQL Server Integration Services |
| **UML** | Unified Modeling Language |
| **HIPAA** | Health Insurance Portability and Accountability Act |
| **OLAP** | On-Line Analytical Processing |
| **PCI DSS** | Payment Card Industry Data Security Standard |
| **PHI** | Protected Health Information |
| **PII** | Personally Identifiable Information |
| **PIPEDA** | Personal Information Protection and Electronic Documents Act |
| **PHIPA** | Personal Health Information Protection Act |
| **MOBAT** | MOdula-BAsed masking Technique |
| **SaaS** | Software as a Service |
| **SDM** | Static data masking |
| **SQL** | Structured Query Language |

# Chapter 1

## 1 Introduction

Nowadays, organizations are undergoing a tremendous change in analytical computing. The vehicle for this change is the implementation of a Business Intelligence (BI) platform, which includes On-Line Analytical Processing (OLAP) Cubes, data visualization tools, and integrated data warehouse (DW).

Extracting sensitive data (Personally Identifiable Information-PII, Protected Health Information-PHI, Payment card information, Intellectual property) from operational databases, then saving them into a consolidated central repository to facilitate efficient analysis is the best solution. However, this huge DW constitutes one of the most serious privacy breach threats that any organization might face when many internal users of different security levels have access to BI components. Hence, data security and reliability are important factors in the analytics platform of BI.

Data masking, also called de-identification, obfuscation, or anonymization, these techniques are used to minimize the inadvertent disclosure risk of sensitive data as well as to preserve the proper quality of data analytics (data utility) within test or BI environment. Using reversible masking is a quite challenge compared to traditional techniques.

The primary purpose for gathering data within BI platform and storing it in integrated data warehouse is to obtain information/knowledge from the data to improve business processes using statistical and data mining tools. The sensitive numerical data are the most important attributes in the analysis process; they pose the greatest threat as they offer great benefit to understand business performance and trends. Moreover, taking into account the importance of some textual data, such as diagnosis, treatment codes in addition to geographical locations.

This chapter discusses the motivation and objectives of this research. It additionally explains the research methodology and the main contributions. Finally, this chapter describes the structure of the thesis.

## 1.1 Motivation and Problem Statement

Business, industries, and healthcare environments are expanding to include not only the traditional information systems, but also a business intelligence and big data analytic platforms. For executive leaders, consultants, and analysts, there is no longer a need to spend hours in designing and developing typical reports or dashboard; the entire solution can be completed through using Business Intelligence (BI) software.

In many organizations, it is often important to create copies of production databases for non-production use, such as, application development and testing, personal training, research, and business analytics modeling as shown in Figure 1.1 below. These multiple copies of sensitive data increase the potential risk of attacks and unnecessary expose to employee who are unauthorized to see the data.

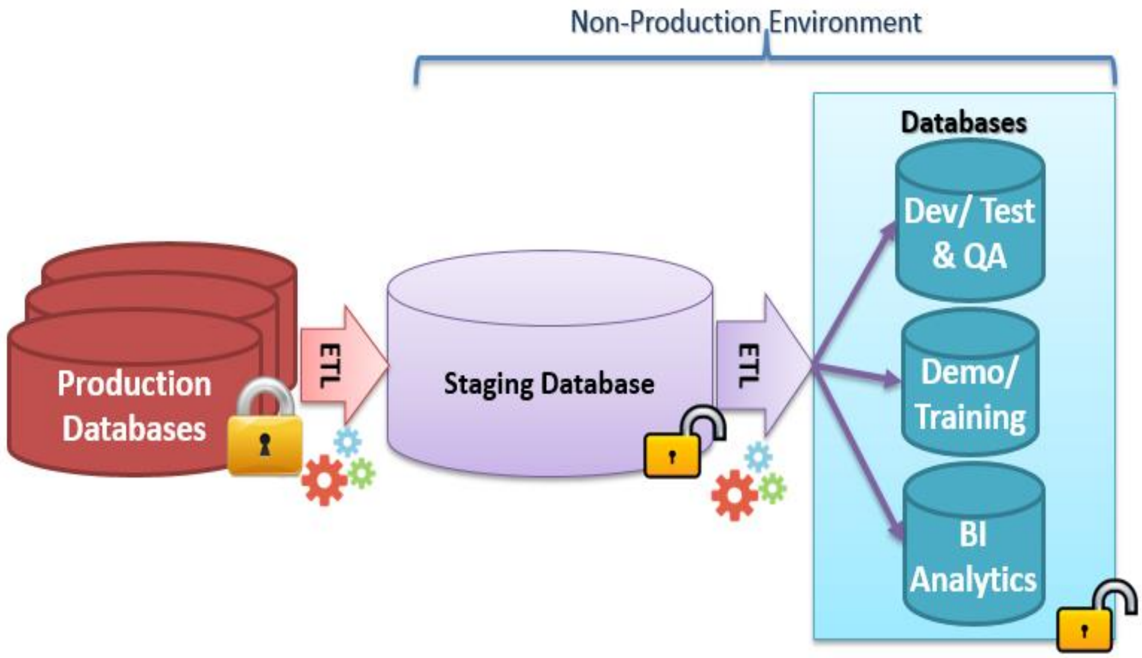
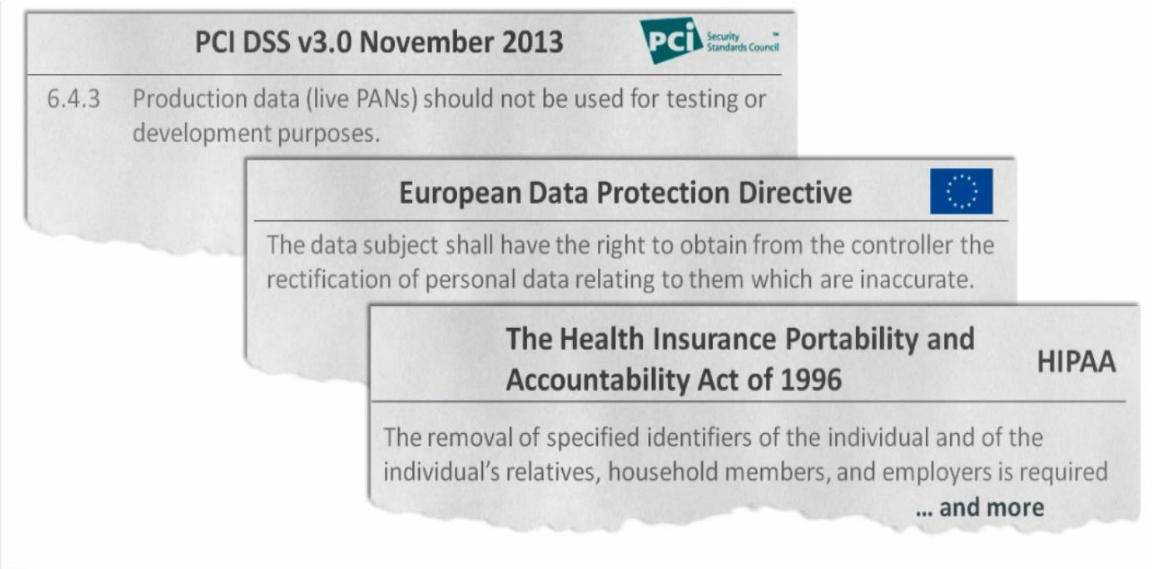


**Figure 1.1: Cloning the production databases to non-production environment for different purposes, using Extract, Transform and Load (ETL) tools**

Another primary reason to protect the sensitive data in non-production environment is to comply with the data privacy rules and regulations that focus on safeguarding the most sensitive data and limit the access to them based on a need-to-know process, see Figure 1.2. For example,

- Personal Information Protection and Electronic Documents Act (PIPEDA), it is sets the ground rules for how private-sector organizations collect, use, and disclose personal information in the course of for-profit, commercial activities across Canada. [44]
- Personal Health Information Protection Act (PHIPA), establish rules for the collection, use and disclosure of personal health information about individuals that protect the confidentiality of that information and the privacy of individuals in Ontario province. [45]
- EU General Data Protection Regulation (GDPR) introduces data minimization and pseudonymization as key data protection principles organizations must follow.
- The Payment Card Industry Data Security Standard (PCI DSS) Requirement 6.4.3, specifically prohibit the use of production data for test and development. [46][58]
- The Health Insurance Portability and Accountability Act (HIPAA), it is US regulation to remove specific identifiers that lead to recover the individuals' information



PCI DSS v3.0 November 2013

6.4.3 Production data (live PANs) should not be used for testing or development purposes.

European Data Protection Directive

The data subject shall have the right to obtain from the controller the rectification of personal data relating to them which are inaccurate.

The Health Insurance Portability and Accountability Act of 1996 HIPAA

The removal of specified identifiers of the individual and of the individual's relatives, household members, and employers is required

... and more

**Figure 1.2: Some compliance drivers for using data masking**

Addressing data privacy and its utility for DW and analysis service in BI platforms is a major challenge that includes the difficulty of masking the sensitive data from internal and external users as the integrated DW originates from different operational databases and contains the disclosed sensitive data.

The traditional data masking method is irreversible in that it masks (de-identify) one-way and fails to maintain the utility of data for analysis, reporting, and research purposes. Several vendors and researchers have proposed and implemented data masking solutions for the test environment, however, most of them are third party standalone applications, whereas others are built-in data masking frameworks based on traditional irreversible masking algorithms.

Neither the traditional data masking techniques nor the third-party solutions can fulfill the need of integrated BI analytics environment to deliver the accurate information to clients while preserving the privacy of the sensitive data within DW and the entire platform. Thus, it is improbable that the BI platform will benefit from having and external solution in which needs to be tweaked and customized to fit the integrated platform.

Hence, it is important to protect textual and numerical data from disclosure while also making it available for analysis purposes. Using the classical data masking techniques are useful to de-identify sensitive data to maintain the confidentiality and privacy, but the serious problem is losing the quality of the data analysis.

Also, conventionally, data masking techniques for protecting such sensitive data are developed manually and implemented independently in an ad-hoc and subjective manner for each application out of BI platform. Such an ad-hoc data masking approach requires time-consuming iterative trial of developing and implementing inconsistent algorithms/techniques on multiple interfacing applications. So to overcome this issue, it's more effective to establish a built-in data masking framework within BI platform encompass the most efficient techniques to be applied directly to the selected sensitive data that need to be masked in staging area.

### 1.1.1 Case Study and Analysis

In this section we will show that the quality measure is absent in most traditional data masking, and specifically with such techniques that deal with numeric data. Hence, we are in need to have other type of data masking.

#### A. Empirical Assessment Using Simulated Health Data

To demonstrate the concept of the traditional masking problem, we used simulated data to illustrate the summary level of Emergency Department's (ED) daily visits (with no patient names and health card numbers). The proposed ED data set consists of 11 attributes with observations reflecting last 30 days summarized records (approx. for 3000 patients' visit), as shown in Table 1.1 below: (see Appendix A for patient flow within at Emergency Department).

**Table 1.1: Simulated aggregated data**

| Data Attribute ($A_1..A_{11}$) | Description | Data Type | Value or range |
|---|---|---|---|
| $A_1$. VisitDate | Visit Date at ED | Date | 2015-05-30 |
| $A_2$. ED_Visits | Number of patients' visits per day | Integer | 125 patients |
| $A_3$. ED_CTAS1 | Number of patients with Triage Level 1 (RESUSCITATION – very bad condition) | Integer | 3 patients |
| $A_4$. ED_CTAS2 | Number of patients with Triage Level 2 (EMERGENCY) | Integer | 28 patients |

| | | | |
|---|---|---|---|
| $A_5$. ED_CTAS3 | Number of patients with Triage Level 3 (EMERGENCY) | Integer | 48 patients |
| $A_6$. ED_CTAS4 | Number of patients with Triage Level 4 (SEMI-URGENT) | Integer | 39 patients |
| $A_7$. ED_CTAS5 | Number of patients with Triage Level 5 (NON-URGENT) | Integer | 2 patients |
| $A_8$. IP_Admits | Number of patients being admitted to Inpatient units (IP) | Integer | 16 patients |
| $A_9$. ED_ALOS_AllDisp | Average Length Of Stay-LOS (Wait Time) to All patients regardless there discharge status | Real | 4.178 hours |
| $A_{10}$. EDALOS_NonAdmit | Average Length Of Stay-LOS (Wait Time) to All patients who non admitted to Inpatient's units | Real | 3.851 hours |
| $A_{11}$. EDALOS_Admits | Average Length Of Stay-LOS (Wait Time) to All patients who admitted to Inpatient's units | Real | 6.096 hours |

The general masking techniques used was "Date and Numeric variance" that does not requires any parameter specifications. We applied random number generation function within SQL Server on numeric data attributes taking in the considerations the ± range of each data element. Please see the SQL statement in Appendix B.

### B. Visual Result of the statistical analysis of the original and masked data

From the sample data set, we selected the first 9 sensitive numeric attributes and applied the "Numeric Variance" algorithm as follows, see Figure 1.3:

1. Consider a database D consist of two tuples T (tables): $D=\{T_1, T_2\}$.
2. Each tuple T consist of set of attributes:

   $T1 = \{A_1, A_2, A_3, \ldots\ldots, A_{11}\}$ , Original Table

   $T2 = \{A_1', A_2', A_3', \ldots, A_9', A_{10}, A_{11}\}$ , Masked Table
3. Identify the sensitive numeric attributes, in our case study $\{A_1, A_2, A_3,\ldots\ldots,A_9\}$
4. Apply the random number generation function: (for more details see Appendix B) (± CAST(NEWID() AS binary(6)) % N) to Original $T_1\{A_1..A_9\}$ as follows:

   $A_1$'= VisitDate ± 10

   $A_2$'= ED_Visits ± 10

   $A_3$'= ED_CTAS1 ± 3

$A_4$'= ED_CTAS2 ± 5

$A_5$'= ED_CTAS3 ± 8

$A_6$'= ED_CTAS4 ± 10

$A_7$'= ED_CTAS5 ± 3

$A_8$'= IP_Admits ± 5

$A_9$'= ED_ALOS_AllDisp ± 5

5. Save new generated numeric values into Masked $T_2$ {$A_1$..$A_9$}.
6. Repeat the above three steps (3, 4, and 5) to all rows.
7. Compare the final results by using SUM and AVG function within Chart data sources

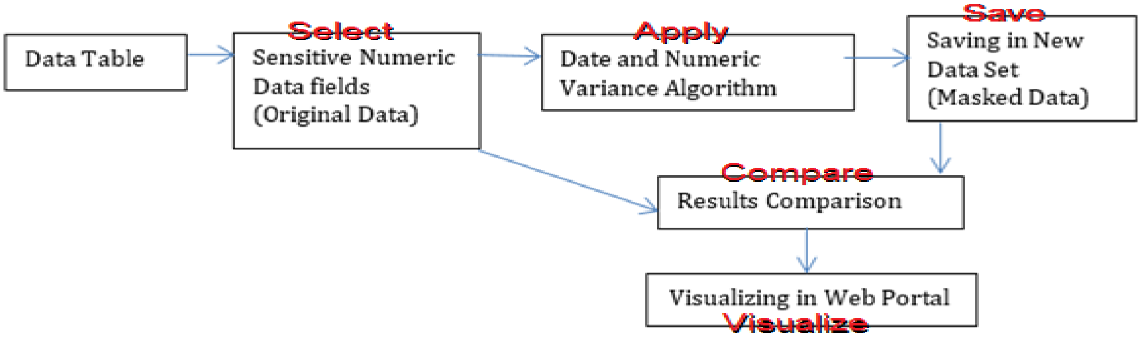

Data Table
Select
Sensitive Numeric Data fields (Original Data)
Apply
Date and Numeric Variance Algorithm
Save
Saving in New Data Set (Masked Data)
Compare
Results Comparison
Visualizing in Web Portal
Visualize

**Figure 1.3: The main steps of applying traditional masking technique.**

We have some potential results illustrating the statistical analysis of the original data versus masked data. We applied the SUM and Average functions at query level to get the final results for past 14 days or 7 days based on the requirements of the chart, as shown in Figure 1.4.

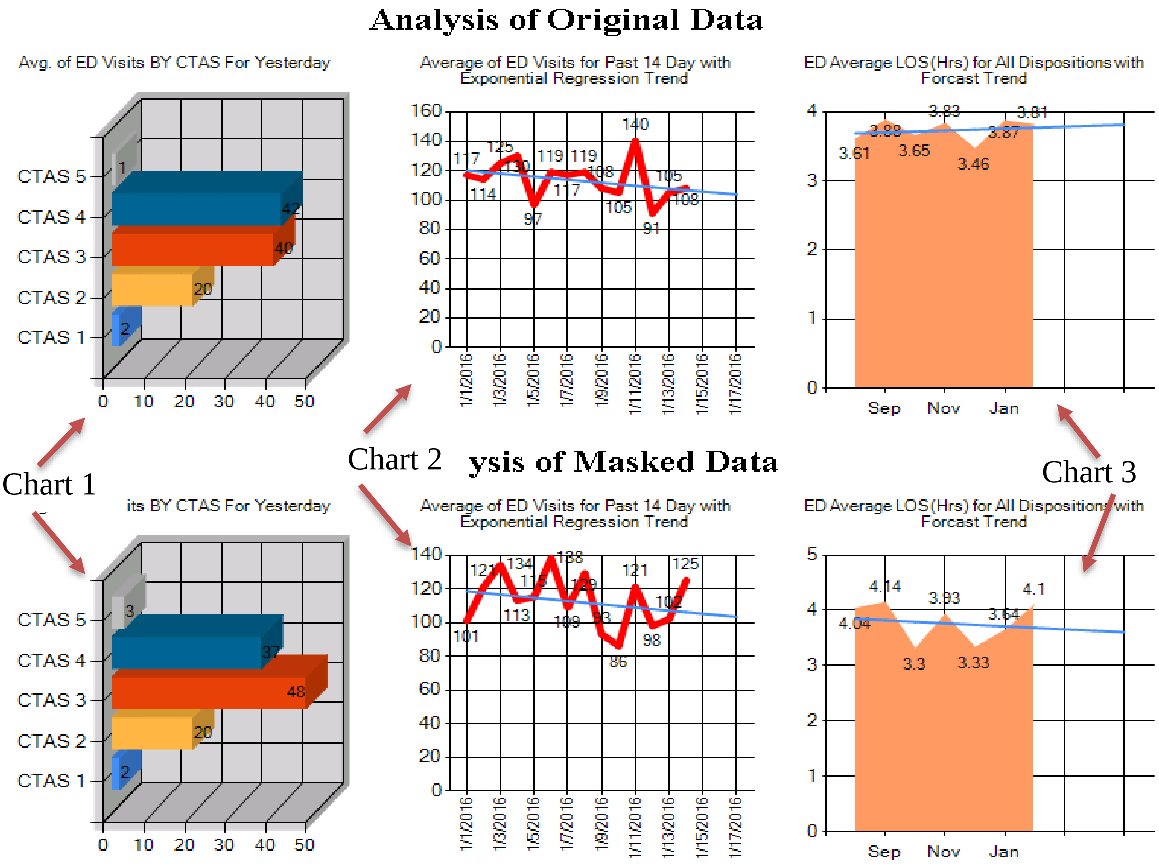


**Figure 1.4: Comparison of the Statistical Analysis between the Original Data and Masked Data**

The results show huge discrepancies between the original and masked data sets, which are not even similar and the masked data did not produce the same outcome as follows:

- Chart 1 depicts the average of Emergency room visits by Triage level (CTAS-urgency of the case) for one day before of the current date, the visit volume of masked data is totally changed and not even close to the actual values.
- Chart 2 shows the discrepancies of the average of Emergency room visits for past 14 days with including the exponential regression trend for next 4 days, the mismatch of numbers will make it unreliable for accurate statistical presentation
- Chart 3 represents the outcome of the average length of stay (LOS) of patients who left Emergency room to anywhere for the past 7 days. Based on the development of the diagram and the established hypothesis the original data can predict the positive trend of LOS. The difference is obvious in which the

forecasting trend declined and get affected negatively by applying the making formula, which is completely inverse direction of the trend line and it is unacceptable in BI analytics platform.

The results of the traditional masking techniques is might acceptable in terms of application testing, improving development functionality, and building a research model, however, it is totally unacceptable from business intelligence data analytics perspective. This means the traditional data masking algorithms whether using random number variance or shuffle the values have a significant negative impact on the data accuracy and quality.

### 1.1.2 Research Questions

The goal of this thesis is to propose a novel technique to build a secure utility-enabled data masking framework for an integrated BI platform. To achieve this goal, the following research questions have to be addressed:

1. **What data masking techniques and approaches can be utilized to build data masking framework in BI Platform?**

   The data masking framework should be usable and scalable in order to fulfill the masking requirements in which can be applied on different types of sensitive data fields. It might utilize one or both categories of masking techniques as follows: (In this framework, we are avoiding to use any encryption techniques due to complex calculation, heavily increase storage space and introduce very large overheads in query response time.)

   - Irreversible traditional masking techniques: such as, substitution, shuffling, masking out, nulling, date aging, numeric alternation.
   - Reversible masking techniques: such as, Pseudonymization (Random Shuffling and Set-Lookup Methods), One way function using Knapsack, Format Preserving De-identification, etc.

2. **What data masking metrics are effective in protecting sensitive data in BI platform while maintaining its utility and privacy?**

   Metrics specifications: Since many factors collected were actually related to the risk, speed, quality metrics, however, in general data to be fit for masking use, data must possess the following attributes:

   - Fitting/Appropriateness: refers to the selection of the proper masking technique for the sensitive data element
   - Quality/Validity: refers to the usefulness of the data or information, accuracy and reliability and is presented in an accurate and clear manner.
   - Safety/Privacy: refers to strength of security or protection of data from unauthorized access and preventing the release of identity of origin data.
   - Performance: refers to the execution speed of masking/de-masking algorithm comparing with no masking task whether in ETL stage or querying layer.
   - Storage Growth: refers to the change of the size of new masked data table in compare with the original one

3. **How can a data masking model be evaluated and used for BI projects?**
   - Defining Assessment methodology
   - Conducting Case Studies for different type of attacks
   - Comparing and interpreting assessment of data in terms of security and quality metrics

## 1.2 Research Objectives

The goal of this thesis is to propose a novel technique to build a secure utility-enabled data masking framework for an integrated BI platform. The following research objectives will help achieve this goal:

| | |
|---|---|
| Objective 1 | Investigate the major components of BI and Data Warehouse, as well as identify the advantage of applying BI platform in healthcare environment. |

| | |
|---|---|
| Tasks: | • Took research-related courses, such as information security, BI & DW, and Data Mining to develop background on the area of the research<br>• Explore the advantage of BI Platform<br>• Examine the existing types of Data Warehouses in traditional BI systems<br>• Examine the existing online analytical processing OLAP in BI platform |
| Deliverable: | Conference paper had been published to demonstrate that the investigation and analysis have been completed |

| | |
|---|---|
| Objective 2 | Highlight the effective data masking metrics in managing sensitive data analysis in BI platform |
| Tasks: | • Conducted a literature review to find out the most significant metrics that being used in the area of data encryption and data masking, such as, Fitting/Appropriateness, Quality/Usability, Safety/Privacy, Performance, and Storage Growth.<br>• Focused on the trade-off between the two significant metrics, data safety and utility, which affect the analysis quality of masked sensitive data. |
| Deliverable: | • Addressed in a conference paper |

| | |
|---|---|
| Objective 3 | Investigate data masking techniques and approaches that can be utilized to build the data masking framework within a BI Platform |
| Tasks: | • Investigated several common traditional data masking algorithms (Irreversible masking) that being used by researchers and industries, such as, fictitious data (random substitution), date aging, numeric alternation, data shuffling, masking out or nulling out.<br>• Examine the existing types of reversible data masking algorithms, such as, Pseudonymization (Random Shuffling and Set-Lookup Methods), One way function using Knapsack algorithm, Format-preserving encryption (FPE) |

| | |
|---|---|
| | • Identify the problems statement and the research questions have been addressed<br>• Determine the stages in BI platform should accommodate the data masking framework, we proposed to be placed in integration services phase before data staging area and within analysis services |
| Deliverable: | • Submitted as a presentation for the 3 Minute Thesis Competitions at UWO, and participated on Annual Grad Symposium 2015.<br>• Addressed in a conference paper to support the idea |
| Objective 4 | Analysis and design of data classification module (Identify & Map) for data masking Framework (iMaskU) |
| Tasks: | • A built-in data masking framework (iMaskU- Identify, Map, Apply, Sign, Keep testing, and Utilize) is proposed<br>• first and second components (Identify + Map) of the proposed secured data masking framework to protect the sensitive data is presented<br>• The hypothetical sample data is based on Canadian healthcare data sources is simulated<br>• develop the data classification module to automate the discovering of the sensitive data, and map the best-fit masking solution to meet the compliance requirements<br>• Define the masking library based on the domain of the business rules |
| Deliverable: | Conference paper has been published to fulfill the Objectives 1, 2, 3, & 4 to ensure that the proposed works have been reviewed by experts in this field and get accepted. |

| | |
|---|---|
| Objective 5 | Design, develop, and validate the new data masking technique COBAD (COntent BAsed Data masking) that uses apply and sign components within iMaskU Framework |
| Tasks: | • Third, fourth, and fifth components (Apply+Sign+ Keep) of the proposed secured data masking framework to protect the sensitive |

| | |
|---|---|
| | data is presented<br>- A new COBAD masking algorithm is well defined and presented<br>- The format of masking signature key alongside with shuffle key are well structured to be used part of the masking process<br>- The masking library domain of the business rules and the user masking definition object are used in masking process<br>- Analyze and validate the security of the proposed BI-based COBAD masking algorithm via three attack methods |
| Deliverable: | A journal paper has been submitted and it is under peer review to ensure the accuracy of the proposed technique |

| | |
|---|---|
| Objective 6 | Design and develop the re-identification process (Utilize component) for the new data masking COBAD technique within iMaskU Framework |
| Tasks: | - The algorithm of the sixth component (Utilize) of the proposed secured data masking framework is presented<br>- The accuracy of the re-identification process is measured to ensure the data utility is within the acceptable range |
| Deliverable: | The built-in re-identification algorithm is validated and the accuracy has been measured |

We can summarize the research challenges and the objectives to propose the best data masking solution for BI platform as shown in Figure 1.5 below.

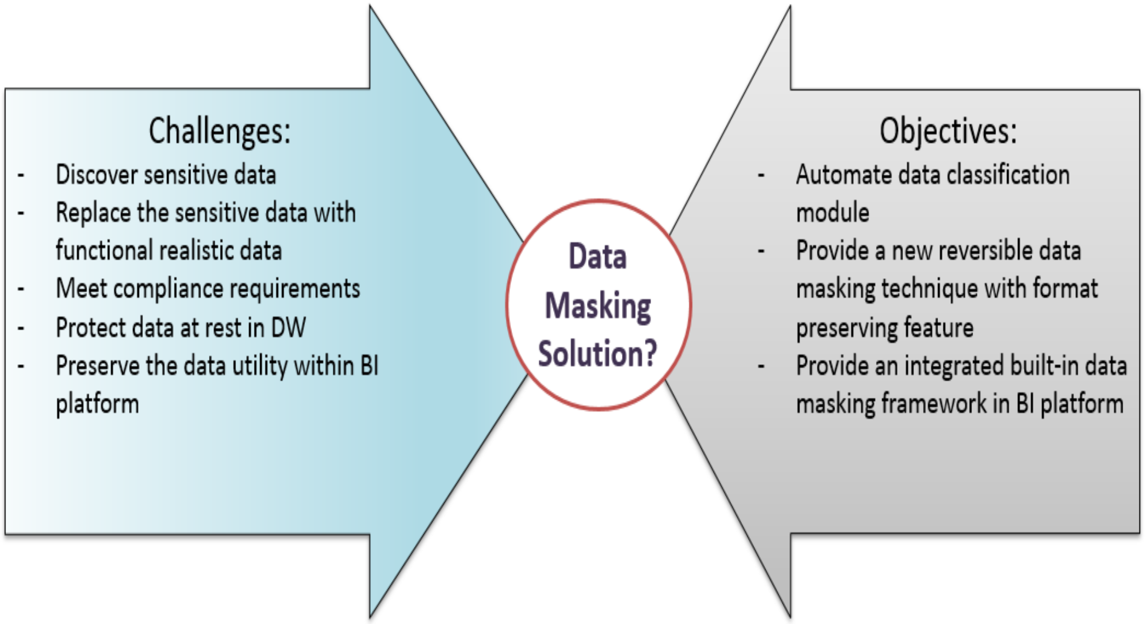

Challenges:
- Discover sensitive data
- Replace the sensitive data with functional realistic data
- Meet compliance requirements
- Protect data at rest in DW
- Preserve the data utility within BI platform
Data Masking Solution?
Objectives:
- Automate data classification module
- Provide a new reversible data masking technique with format preserving feature
- Provide an integrated built-in data masking framework in BI platform

**Figure 1.5: Finding a Data Masking Solution based on challenges and objectives.**

## 1.3 Research Methodology

This section describes the methodologies that are applied in this research to design and develop the built-in data masking framework and technique to secure the privacy of sensitive data at rest in data warehouse against internal and external attacks. The proposed BI-based data masking framework and technique can not only protect data at rest, but it can also recover the original data for utility purposes. In addition, the designed and developed framework satisfies the following requirements:

- Read the heterogeneous data from different data sources.
- Ability to identify the sensitive data based on pre-defined business rules.
- Mask the sensitive data in a reversible way that can be recovered for research and analysis purposes.
- Validation of the security of the masking technique through different attacking approaches.
- Use formal validation to measure re-identification accuracy.
- Validate the performance of the new masking algorithm versus standards encryption techniques

**Notes: A)** All diagrams notation follow the UML standards to be a common language for the reader and practitioners. **B)** The used technical platforms to accomplish all the practical results were MS-SQL Server, MS-Excel, and ASP.Net with C#.Net

### 1.3.1 Data Classifier Module Design

It is a process of identifying the sensitive data from data source and classify the privacy levels of provided data that would make it easier for user to determine how best to keep data safe in the DW in which contain direct identifiers, quasi-identifiers, and other sensitive attributes. Our approach is based on analysis of the entire attributes of the sample data through exploring the enterprise reporting and data definition documents, as well as expert determination method that relies on extensive questionnaire (cover these topics: data appearance and usability, static vs dynamic masking, reversible vs irreversible, uniqueness-consistency, security strength, and privacy policies). This study led to describe their occurrences in health reporting systems in terms of frequency and sensitivity categories (PII/PHI/None). The classifier module is totally relies on the business rules in which define all the data attributes that needs to be masked or de-identify in order to comply with the privacy regulations.

After a clear analysis of masking methods for sensitive data and the privacy requirement, a set of business rules established based on intensive investigation of different BI tools in healthcare to map the recognized sensitive data to the appropriate masking method in which fit the usage nature of the data attribute. The data being initially masked is profiled to detect invalid data and an extra information attribute will be added

The classification module of the masking framework focuses on the first two components, which are “Identify” and “Map.” The detailed processes of these two components have been set up and organized in Chapter 3.

### 1.3.2 COBAD Module Design (Apply, Sign, and Keep)

It is the second module (Apply, Sign, and Keep) stands for (COntent BAsed Data masking - COBAD) of the BI-based data masking framework is proposed to protect the privacy of data and prevent internal and external attackers from disclosing the original information at rest. In addition, it securely de-identifies the sensitive data using a well-defined light mathematical formula relied on the statistical content of the extracted data.

The new COBAD data-masking algorithm (reversible de-identification) is based on the derived statistical variables of the extracted data content and how to code the shuffle of the sequence of these variables, then followed by pack the sequence codes in a binary format to generate the Shuffle Key $K_{SH}$ with application of AES256 encryption before saving it in a protected table. Furthermore, the values of the statistical variables have been packed in a binary Masked Signature Key $K_M$. The $K_M$ is publicized, kept with each row and saved into the destination data table.

COBAD emphasizes the trade-off between data privacy and its utility by preserving the data format and type to keep it looking realistic for developer, tester, and data scientist within the non-production environment (e.g., DW) by using a new reversible data masking technique in the context of maintaining the utility of data analytics

### 1.3.3 Re-identification Module

The re-identification process is the last component in our iMaskU framework, which is called “Utilize”. The utilize component is a reverse engineering process that adds a built in functionality in which acts as a de-masking tool to enable the authorized end-user to retrieve data and apply the required analysis process to get the right results to be utilized by analytical tools

The algorithm of the de-masking process relies on reading the public masking signature key ($K_M$) for each record and then interprets its contains by using the associated shuffle key ($K_{SH}$) to get the right order and size of the statistical variable that had created from the extracted data content. In addition, from the size of $K_M$, the re-identification function determines the complexity of the masking formula in which used to de-identify the sensitive data attributes.

### 1.3.4 Utility Validation: Accuracy Measure

The accuracy is the degree to which the result of an analysis conforms to the correct values. To calculate the accuracy of the re-identification process, we are going to calculate the ratio of an error of de-masked value versus the original one. We compare the re-identified value with the original to determine the accuracy ratio using below formula:

%Accuracy = de-mask / original × 100%

The accuracy results show, by applying our COBAD's simple masking formula (Min, Max, $K_{S1}$, $K_{S2}$) on LOS will get 100% accuracy result. On the other hand, if we apply Min-Max normalization formula (Min, Max, newMin, newMax) to the Age attribute, the re-identification accuracy reaches to ~98%, which is acceptable range from statistical analysis perspective [76].

### 1.3.5 Security Validation: Re-identification Risk Measure

To validate the security requirements of COBAD data masking technique, it is important to analyze the strength of the proposed algorithm against the most common attack methods. Therefore, the proposed COBAD masking algorithm is formally analyzed via these attacking methods to validate its security strength (Section 5.3).

The following attacking methods are considered, a) using a brute force attack, needs, on average, 55 years to crack the key of each record; b) the dictionary-based attack, needs 6.6 minutes to crack the key for a single record, and 231 days for the entire extracted dataset (contains 50,000 records), c) a data linkage attack, the re-identification risk is very low when the common linked attributes are used (e.g., postal code, age group, and sex). Only two records have been identified out of 51,000 (0.004%).

### 1.3.6 Performance Validation

We explain and analyze the performance comparison between COBAD (simple, medium, complex) technique and AES128 3DES encryption algorithms using the well known TPC-H benchmark. TPC-H is a decision support benchmark in which consist of datasets and ad-hoc queries that examine large volume of data [72].

## 1.4 Main Thesis Contributions

This thesis focuses on designing and developing a novel data masking technique to secure the sensitive data at rest against the external and internal risks of attacks and to maintain the utility of data for the analytics within BI platform. Research contribution can be mainly summarized as follows:

- A taxonomy of existing data masking techniques within the database management system (DBMS) or as a third part solution.
- An investigation and design of a data masking framework iMaskU starting with a classifier module in order to automate the process of identification the sensitive data and then mapping them with the proper masking technique.
- An investigation and design of a COBAD module to apply the proper proposed algorithm based on the type of the sensitive data and then construct the shuffle key and masking signature keys
- Design of the required re-identification functionality and embed it into the analysis services of the BI platform
- A security validation of the proposed COBAD technique against internal and external attacks.

The research contributions of this thesis have been published in conference proceedings and a journal paper in the areas of information systems and security. Therefore, these contributions have been peer-reviewed by external researchers who are experts in the field.

## 1.5 Thesis Structure

The thesis structure is outlined as follows:

- Chapter 2 provides an overview of the BI platform in general and the related data protection processes static or dynamic. It presents an in-depth available data masking technique that protects the sensitive data in the test environment and DBMS. Also, it presents a literature review of the existing data masking research from industry or and academic perspectives.

- Chapter 3 presents and discusses the data classification module for a built-in data masking framework (iMaskU- Identify, Map, Apply, Sign, Keep, and Utilize), and the first and second components (Identify + Map) of the proposed BI- based secure data masking framework to protect the sensitive data are introduced.

- Chapter 4 presents and discusses the second module (COBAD technique, Apply and Sign) of the BI-based data-masking framework to protect the privacy of data and prevent internal and external attackers from disclosing the original information at rest. In addition, it securely de-identifies the sensitive data using a well-defined mathematical formula.

- Chapter 5 describes the re-identification formulas and the final results have been compared with the outcomes of the analysis of the original data to measure the accuracy rate. Furthermore, the achievement of the security requirements of the COBAD masking algorithm by using simple formula is verified. In addition, the performance of the algorithm is assessed.

- Chapter 6 summarizes the contributions of the thesis and outlines the future work.

# Chapter 2

# 2 Background and Literature Review

This chapter overviews the BI platform in general and the related data protection processes whether in static or dynamic fashion. It presents an in-depth available data masking technique that protects the sensitive data in the test environment and DBMS. Finally, it presents a literature review of the existing data masking research from industry and academic perspectives.

## 2.1 Business Intelligence (BI) Platform

BI is one of the hottest buzzwords in the last several years in the business administration and information management fields. There are many definitions of BI:

"BI is a strategic initiative by which organizations measure and drive the effectiveness of their competitive strategy" [1]. To achieve this grand goal, analysis, software, resources, technical leadership, process specialists, executive leaders, and much more are needed.

"BI is a broad category of applications and technologies for gathering, storing, analyzing, sharing, and providing access to data to help enterprise users make better business decisions", [2]. Gartner also defines a BI platform as a software platform that delivers the 14 capabilities listed below within three main categories of functionality:

1. Integration: BI infrastructure, Development tools, Metadata management, Collaboration.
2. Information Delivery: Reporting, Ad-hoc query, Dashboards, Data integration, Search-based BI, Mobile BI.
3. Analysis: Online analytical processing (OLAP), Interactive visualization, Data mining and Predictive modelling, Scorecards (Key Performance Indicators -KPIs, Performance Management Methodology) [2].

BI goals are often to:

1. Contribute to the button line by measuring specific operations.

2. Enhance competitive advantage.
3. Achieve secure and reliable access to data by developers and decision makers to do their jobs effectively, (this is the most critical point of this work).
4. Use flexible tools to browse the information.

Figure 2.1 represents the essence of BI architecture with the proper workflow of the interdependent components [7]:

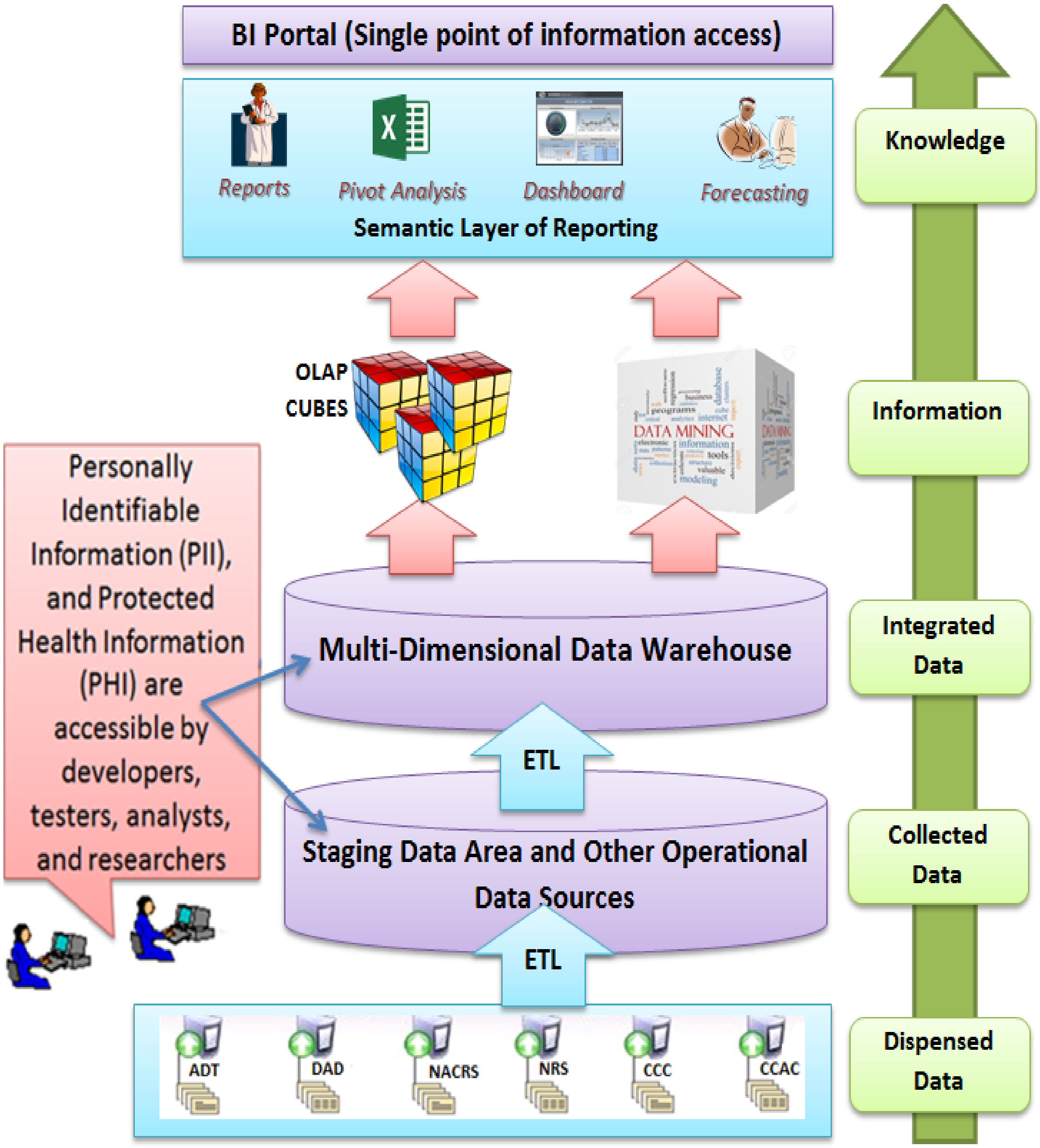

BI Portal (Single point of information access)
Reports
Pivot Analysis
Dashboard
Forecasting
Semantic Layer of Reporting
OLAP
CUBES
DATA MINING
Personally Identifiable Information (PII), and Protected Health Information (PHI) are accessible by developers, testers, analysts, and researchers
Multi-Dimensional Data Warehouse
ETL
Staging Data Area and Other Operational Data Sources
ETL
ADT
DAD
NACRS
NRS
CCC
CCAC
Knowledge
Information
Integrated Data
Collected Data
Dispensed Data

**Figure 2.1: The main components of BI architecture and the location of PII & PHI sensitive data.**

- **A Data Warehouse (DW)** is the core of any solid BI solution. A Data Warehouse can be defined as a "repository for keeping data in a subject oriented, integrated, time-variant and non-volatile manner that facilitates decision support" [3][4]. Basically, it is a big database containing all the data needed for performance management, decision making, and prediction. Multi-dimensional modeling techniques use facts and dimensions within relational or multi-dimensional databases and are typically used for the design of corporate data warehouses and departmental data marts. Such a model can adopt a star schema, snowflake schema, or fact constellation schema [6][52].

- **External source systems** are not really considered as a part of the BI environment, but they feed the BI solution, so they are at the base of the whole architecture and should be totally understood by developers. One of the important things in the set-up of a BI environment is to consider all the types of data that may need to be included in the analysis process.

- **ETL: Extract, Transform and Load (Integration Services):** After building up a multi-dimensional data warehouse, data from all different sources need to be extracted and brought to the BI environment; sensitive data disclosure is a possibility here. After the extraction task, the data need to be transformed. The transformation process can mean a lot of things, including all activities to make the data fit the multi-dimensional model of the data warehouse. The proposed data masking framework will be applied in this stage.

  The transformation process may become quite complex, especially when it includes extra work to clean up, harmonize, and secure the data coming from different systems, which is why most BI professionals describe ETL work as 70% of the IT side of a BI project.

  While working on the development of the ETL component, a separate database in the data warehouse is reserved as a storage space for intermediate results of the required transformations. This area is called a staging area or work area. Once the transformation work is done, the prepared data can be loaded into the multi-dimensional model; this step is not as complex as the transformation, but attention needs to be given.

- **Online Analytical Processing (OLAP) cube** refers to analysis techniques (Analytical Services stage) including a variety of functionalities such as aggregation, summarization, and consolidation as well as the ability to view information from different angles [7]. OLAP offers high performance in analysis and loading of the data. OLAP cubes have had very high success rates for business environments where the BI solution is used for what-if analysis, financial simulations, budgeting and target setting, etc.

- **BI portal:** When the number of different reports begins to grow, the best solution is to create a single point of access to information within the organization. Usually, the effort of creating a single point of access results in building an intuitive portal solution that contains different reports with clear descriptions of the scope of each report, as well as an indication of who is the business owner of the report.

As shown in Figure 2.1, the staging storage area and multi-dimensional data warehouse (DW) are considered the core components of the BI platform and are an integrated repository derived from multiple data sources (operational and legacy) in the production environment. Saving sensitive data into a central repository is a serious privacy disclosure threat when many internal users of different security levels have access to the BI services. Thus, data privacy and the reliability/utility are considered important issues that can be compromised in the BI platform while using data masking techniques [7].

## 2.2 Cryptography, Reversible and Irreversible Data Masking

Cryptography is the science and art of secret writing or “Secret codes”, it does so using cryptosystems, or simply Ciphers. Cipher is used to encrypt or “encode” a message or information called “plaintext”, the result of this encryption is called ciphertext. The aim of the encryption is to protect a message so that only authorized parties, who possess a secret key, can access it by decrypting the received ciphertext. There are two main types of cryptosystems; a symmetric key cryptosystem uses the same key to encrypt as to decrypt messages, and a public key cryptosystem uses a public key to encrypt a message and a private key to decrypt (sign) later. The most common and secure symmetric key algorithms, e.g., AES and 3DES, depend on a mix between linear and nonlinear substitution, permutation and shifting operations to be executed on many number of rounds depends of the key size.  Public key algorithms are usually based on number theory rather than substitution or permutation operations, e.g., RSA, where a trap door, one-way function is used so that the it is easy to compute in one direction and it is very hard to compute in other direction. These types of function usually have a “Trap door” to create keys to be used for encryption and decryption. The more “computationally” complex of these functions, the more cryptography secure, and the longer key size used, the more complex these algorithms will be. In general, cryptography needs computational capabilities in addition to extra storage to achieve reasonable level of security.

Data masking is the process of replacing a sensitive data values such as credit card numbers with a fake yet realistic looking credit card number. Typically, the main objective of data masking is like cryptography in the sense that it makes sensitive information not available to unauthorized users, but without the need for heavy computations and/or many number repeated operations, see Figure 2.2.  Substitutions and replacement are the most common operations that are used in data masking techniques, the detail description of these techniques is given in the next Section. There are two types of data masking; traditional data masking also called irreversible data masking, and key-based reversible data masking. The most common data masking techniques are not reversible, where the masking process is done in such a way that there should not be any way to retrieve original data from masked data. However, in key-based reversible data masking the original data or near original data can be retrieved if a proper secret key can be generated during the masking process. The generation of such keys does not require heavy computation as well.

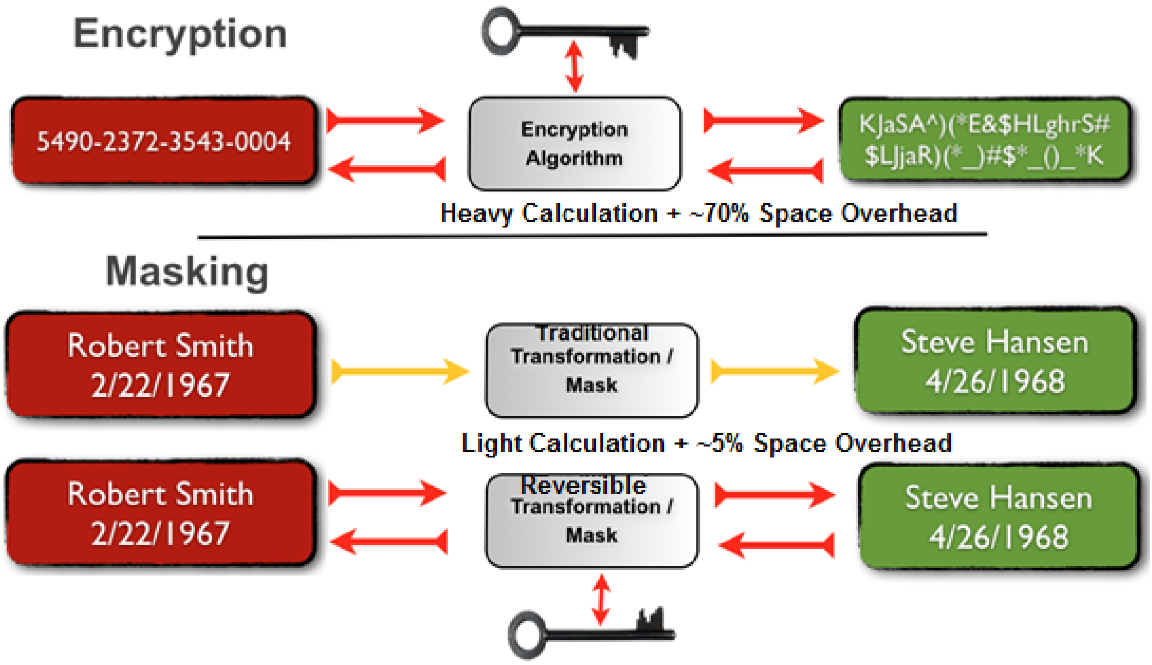

Encryption
5490-2372-3543-0004
Encryption Algorithm
KJaSA^)(*E&$HLghrS# $LJjaR)(*_)#$*_()_*K
Heavy Calculation + ~70% Space Overhead
Masking
Robert Smith 2/22/1967
Traditional Transformation / Mask
Steve Hansen 4/26/1968
Light Calculation + ~5% Space Overhead
Robert Smith 2/22/1967
Reversible Transformation / Mask
Steve Hansen 4/26/1968

**Figure 2.2: Encryption vs Data Masking**

## 2.3 Data Masking Techniques

Traditionally, data masking has been viewed as a technique for solving data privacy problems in non-production environments such as cloning databases for testers and developers to add new features to the application. However, the Gartner Magic Quadrant Report in December 2013 extends the scope of data masking technology to more broadly include data de-identification in production, non-production, and analytic use cases. [5]

Masking and de-identification deal with different attributes in a data set, so some attributes will be masked (irreversible) and some fields will be de-identified (reversible) to retrieve the original value later. Masking involves protecting direct identifiers such as patient name, SIN number, telephone number, email, and health card number (HN). The de-identification process involves protecting attributes by covering things like demographic and socio-economic information such as age (date of birth), postal codes, diagnosis code, income, number of children, and race, or other PHI, which are called indirect identifiers (Quasi-identifier QID) [8][50].

Data Masking is important for all enterprises for the following reasons:

- Protecting sensitive data in a non-production environment (i.e., Integrated Data Warehouse)
- Helps meet compliance requirements. The field-level de-identification methods can help to comply with data privacy regulations while leaving your non-sensitive data for further processing.
  Note: HIPAA (Health Insurance Portability and Accountability Act) compliance focuses on protecting patient health information to standardize communication between health care providers and health insurers and to protect the privacy and security of protected health information (PHI).
- Minimizing information risk when outsourcing or off-shoring

Masking and de-identification deal with different fields in a data set so that some fields will be masked (non-reversible), and some fields will be de-identified (reversible). Masking involves protecting things like names, SIN numbers, and health card numbers HN, which are called direct identifiers. De-identification involves protecting fields by covering things like demographic and individuals’ socio-economic information like age, postal codes, income, number of children, and race, or even protected health information, which are called indirect identifiers.

### 2.3.1 Traditional Data Masking Algorithms (Non-Reversible):

Several common traditional data masking algorithms can be used such as [9] [10] (See Table 2.1)

1. **Random Substitute (Fictitious data):** This technique substitutes data with similar random values from a pre-prepared dataset, making the data look real when it is in fact bogus. This technique does not typically affect the application or testing requirements because it retains all of the data properties. For example, the patient name "John Brown" could be substituted with the name "Jim Carlos" (could be enhanced by maintaining the gender).

2. **Date aging:** In this technique, and based on pre-defined policies and business rules, the date attribute is either increased or decreased. However, date ranges must be defined within acceptable boundaries so that the data utility is least affected. One example is moving the registration date back by 20 days within the same month minus 1 year, which would change the date "25-Jan-2015" to "05-Jan-2014."
3. **Numeric alternation:** In this approach, based on an acceptable percentage or range, the numeric attribute will be increased or decreased accordingly. For example, a wait time value could be increased by 10% of its original value. This approach disguises the real value of the data, but if someone knows even one real value, they could decode the entire pattern. This is an easy technique to mask, and it can also be easily de-masked.
4. **Shuffling data:** In this technique, a data attribute will be used as its own substitution dataset; it moves the values among rows in such a way that the no values remain in their original rows with a broken sequence. The drawback is with respect to the huge data and the pattern of shuffling: there would be a repetition of the shuffling patterns. For example, a patient name is moved to another random row.
5. **Masking out or Nulling out:** The Masking Out technique sanitizes the data by replacing certain specified characters with masked characters (i.e., a health card number might be hashed out as 552 88# ####). This effectively disguises the data content while preserving what was omitted on front-end screens and reports [11]. The Nulling Out technique simply removes the specified sensitive data by replacing it with NULL values; it is a simple way of ensuring that it is invisible.

**Table 2.1: Common traditional techniques of data masking**

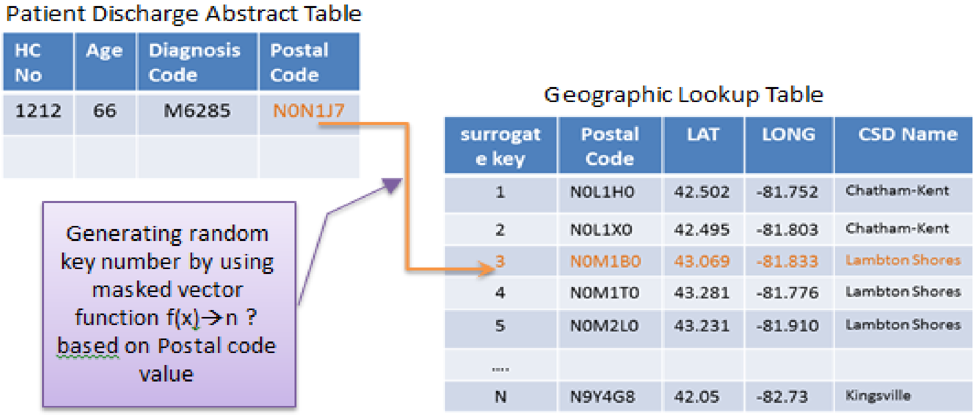
| **Algorithm** | **Original data** | **Masked Data** | **Explanation** |
|---|---|---|---|
| **Masking out** | 552-888-3291 | 552-88#-#### | Health Record Number’s Last four characters hashed out |

| | | | |
|---|---|---|---|
| **Random Data Substitute** | John Brown | Jim Arthur | Random data substitution from pre-prepared dataset |
| **Date Aging** | 2015/05/25 | 2013/05/15 | Date of admission decreased by 2 years and 10 days |
| **Numeric Alteration** | 10201 | 10401 | Patient's Length of Stay value increment by 200 |
| **Data Shuffling** | N6G7K8 | N5V1A1 | Postal Code was shuffled based on the same data set taking geo-location into an account. |
| | M6285 | M4433 | Diagnosis Code ICD-10 was shuffled within the same block. (See Appendix C) |

### 2.3.2 Reversible Data Masking Algorithm:

The de-identified attributes can look more realistic and still be recovered by using the following techniques:

1. **Pseudonymization (Random Shuffling and Set-Lookup Methods):** A method of shuffling data attributes to preserve data confidentiality that comprises masking particular attributes of a dataset which are to be preserved in confidentiality, followed by a shuffling step, which comprises sorting the transformed dataset and a transformed confidential attribute in accordance with the same rank order criteria. Figure 2.3 shows the general idea behind this method [12][53].



Patient Discharge Abstract Table

| HC No | Age | Diagnosis Code | Postal Code |
|---|---|---|---|
| 1212 | 66 | M6285 | N0N1J7 |
| | | | |

Geographic Lookup Table

| surrogate key | Postal Code | LAT | LONG | CSD Name |
|---|---|---|---|---|
| 1 | N0L1H0 | 42.502 | -81.752 | Chatham-Kent |
| 2 | N0L1X0 | 42.495 | -81.803 | Chatham-Kent |
| 3 | N0M1B0 | 43.069 | -81.833 | Lambton Shores |
| 4 | N0M1T0 | 43.281 | -81.776 | Lambton Shores |
| 5 | N0M2L0 | 43.231 | -81.910 | Lambton Shores |
| .... | | | | |
| N | N9Y4G8 | 42.05 | -82.73 | Kingsville |

Generating random key number by using masked vector function f(x)→n ? based on Postal code value

**Figure 2.3: The general idea of the Pseudonymization method**

2. **Format-preserving Encryption (FPE):** This technique differs in its purpose from the most common encryption techniques (i.e., DES, AES) which use an enhanced algorithm similar to AES Encryption (with truncation of the output) to retain the original formatting of the plaintext data, so that it appears real, preserves the utility of data by applying simple re-identification calculation, and compromises the risk factor [14,15].

   For example, a diagnosis code "M6285" can be transformed to "Z2138" maintaining the same alpha-numeric sequence and format.

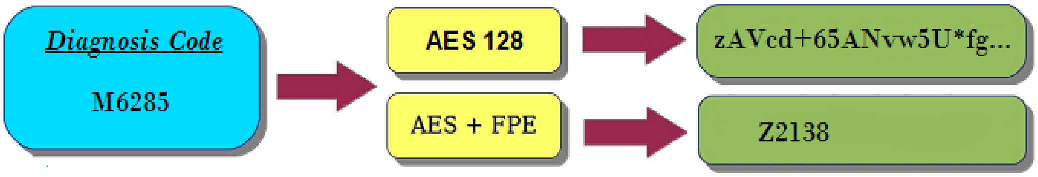


**Figure 2.4: Format Preserving Encryption (FPE)**

3. Modulus-based Masking algorithm: This technique applies a mathematical modulus operator on numerical attributes. It is called MOBAT (MOdulus-BAsed Technique) and is based on a formula that depends on three masking keys: two of them are encrypted private ($K_1$ and $K_2$, are not available to any users, including DBAs), and the third key is public ($K_3$). [18]

   Suppose a table T with a set of N numerical columns, $C_i = \{C_1, C_2,C_3, \ldots, C_N\}$ to be masked and a total set of M rows

   $R_j = \{R_1, R_2,R_3, \ldots, R_M\}$. Each value to be masked in the table will be identified as a pair $(R_j, C_i)$, where $R_j$ and $C_i$ respectively represent the row and column to which the value refers.

   Each new masked value $(R_j, C_i)'$ is obtained by applying the following formula for row j and column i of the table T

   $(R_j, C_i)' = (R_j, C_i) – ((K_{3,j} \text{ mod } K_1) \text{ mod } K_{2,i}) + K_{2, i}$

   The formula to retrieve the original data is:

   $(R_j, C_i) = (R_j, C_i)' + ((K_{3,j} \text{ mod } K_1) \text{ mod } K_{2,i}) - K_{2, i}$

## 2.4 Static versus Dynamic Data Masking

In a previous chapter, we mentioned that most organizations create copies of production databases for non-production use for many reasons, such as, application development and testing, personal training, building a prototype for research and analytics model. Data masking comes in two main fashions as follows [57][59]:

A. **Static Data Masking:** it is masking the original data at rest permanently in the physical storage layer. Almost all database vendors (e.g., Oracle, IBM) adopted this approach, see Section 2.5.1. This starts with reading data from production datasets and then applies a series of data transformational rules (reversible or non-reversible techniques) to produce realistic de-identified data and then save them into the destination data tables, as shown in Figure 2.5 below.

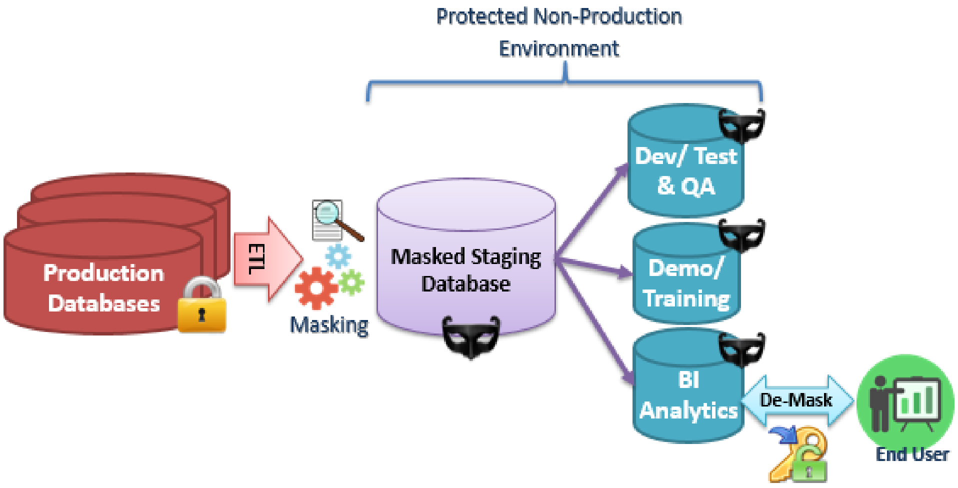


**Figure 2.5: High-Level Architecture of the Static Data Masking**

**Pros:**

- Sensitive data is permanently obfuscated due to application of the data masking techniques during the extract, transform, and load process.
- No need to any process when data is retrieved during the usage phase (if the accuracy of the result is not matter).
- Safe to share your data with internal and external stakeholders.

**Cons:**

- The masking process during ETL process may take minutes to complete depending on the size of extracted dataset.

- It cannot be easily used to back-up the production datasets as it needs to apply de-masking algorithm to the entire masked dataset and this may take time.

B. **Dynamic Data Masking:** it is anonymizing sensitive data on the fly at presentation layer during user request, leaving the original data at rest with no change. The primary use of dynamic data masking is to apply role-based security functionality to each end-user in a read-only context for reporting purposes. Most database vendors (e.g., Microsoft SQL Server) adopted this approach (see Section 2.4.1.) and used proxy layer to modify the SQL query and return a modified query result through applying different type of traditional masking techniques, as shown in Figure 2.6 below [59]:

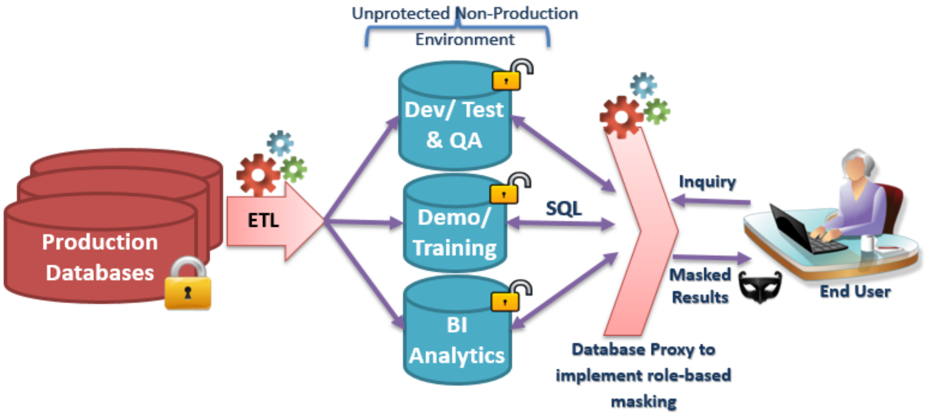


**Figure 2.6: High-Level Architecture of the Dynamic Data Masking**

**Pros:**

- Adds an additional privacy layer to protect the sensitive data.

- Controls the data protection at presentation layer in read-only approach.

- No need to apply masking techniques in advance to mask the extracted dataset.

**Cons:**

- Not proper to be sued in BI integrated DW environment to prevent the internal data breach.
- Performance overhead associated with inspecting all inquiry traffic.
- Needs detailed mapping of users, datasets, data fields, and masking rules as well as to maintain this matrix configuration.
- The database proxy is a single point of failure and might be bypassed by the end-user.

## 2.5 Literature Review

Research on data masking, in general, can be classified into two categories: static and dynamic models.

The Static model(data-at-rest) is built based on applying de-identification out of historical data and does masking based on most sensitive data fields identified by the user for non-production databases, so it applies physical data transformation without taking into consideration the outcome quality of data analysis; it's just for system testing purposes.

Meanwhile, the Dynamic model (data-in-motion) applies the transformation techniques on a query layer to hide the identified sensitive data, while keeping the data as its own and applying masking techniques based on users' security roles [58].

Also, we noticed that the quality factor/metric for data masking framework within BI platforms have not been covered in a systematic and experimental way.

### 2.5.1 Traditional Data Masking Researches and some Industrial Applications

- Ravikumar et al. (2011) conducted an analysis of traditional data masking techniques for testing purposes to design a uniform application architecture to automate processes that reduce the exposure of sensitive data without considering the usefulness of the analyzed data. The entire architecture is divided into two sub-divisions as Analysis and App&DB. In analysis, the sensitive data is identified and further processed for the implementation of Data Masking. In App&DB, they performed an empirical assessment of the two masking techniques using two data sets. The first masking technique used was data shuffling that does not require any parameter specifications. The second masking technique was the SBLM procedure with the requirement that Beta2 be a diagonal matrix. Their measurements highlighted the strengths and weaknesses of the assessment of disclosure risk. [9][11]

- Min Li et al. (2014) analyzed basic algorithms (such as numeric alteration, format preserving encryption, random substitution) and classic schemes of data masking, and provided a formal definition of data masking according to the work principle and process. At the same time, the paper proposed a generic data masking model (without considering the quality of data analysis), described and analyzed the algorithms' process of masking functions to implement secure and effective masking operation for data. The masked data generated from the generic model is similar to the original data sufficiently such that the model can satisfy the requirement of development and testing without the leakage of sensitive data [16].
- Based on Oracle Inc., Oracle Data Masking helps reduce this risk by **irreversibly** replacing the original sensitive data with fictitious data so that production data can be shared safely with non-production users [17]. Oracle has developed a comprehensive 4-step approach to implementing data masking called Find, Assess, Secure, and Test (FAST). These steps are:

  Find: This phase involves identifying and cataloging sensitive or regulated data across the entire enterprise. The goal of this exercise is to come up with a comprehensive list of sensitive data elements across enterprise databases that contain the sensitive data.

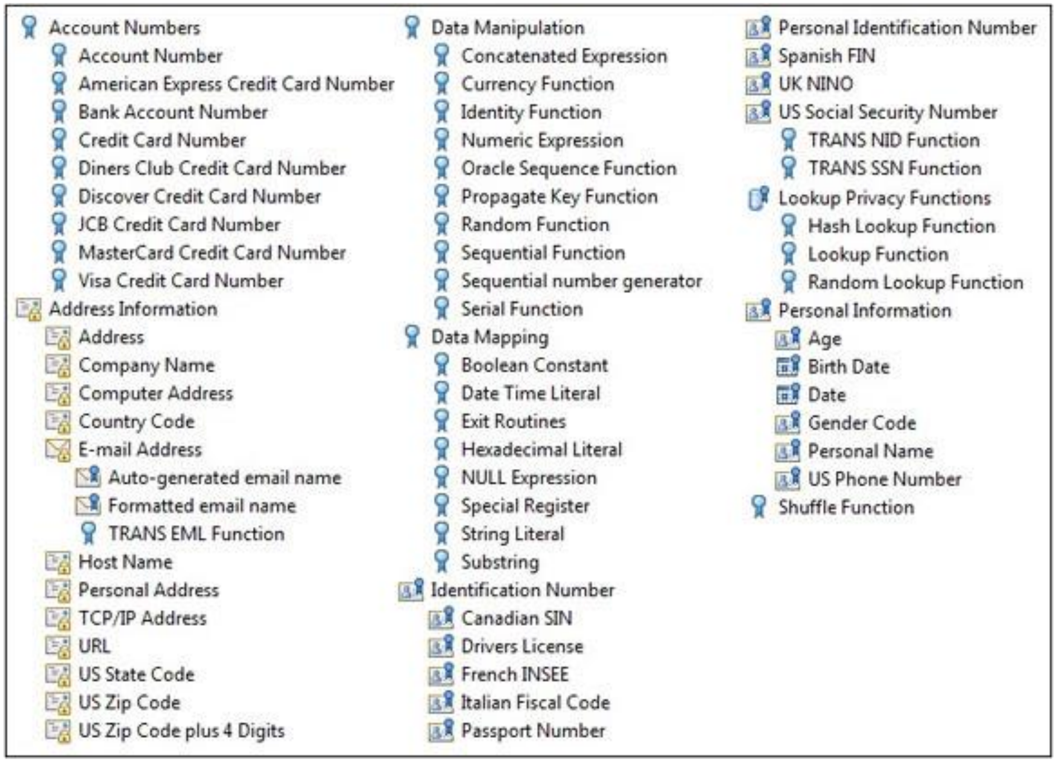
  Assess: In this phase, identify the masking algorithms that represent the optimal techniques to replace the original sensitive data. Developers can leverage the existing masking library or extend it with their own masking routines.

Secure: This and the next steps may be iterative. Execute the masking process to secure the sensitive data during masking trials and then verify the process.

Test: In the final step, execute application processes to test whether the resulting masked data can be turned over to the other non-production users. If the masking routines need to be tweaked further, the DBA restores the database to the pre-masked state, fixes the masking algorithms and re-executes the masking process [17][48].

- Based on SQL Server 2008/2012/2016 DBMS and BI tools, Microsoft added built-in dynamic data masking at the presentation layer with no change to the original data [19].
- Based on IBM InfoSphere Optim and DataStage, IBM has developed and integrated platform for defining, integrating, protecting, complying with privacy rules, and managing trusted information across the production and non-production databases, by using static (non-reversible) and dynamic data masking techniques as shown in Figure 2.7 below [68].

  Users can apply a variety of traditional data masking algorithms to replace sensitive real data with contextually accurate (not statistical accurate) and realistic fictitious data. IBM Optim and DataStage include substrings, arithmetic expressions, random or sequential number generation, date aging and concatenation techniques. Plus, the solution’s context-aware masking capabilities help ensure that masked data retains the look and feel of the original information, however, the masked data cannot be use in BI analytics platform in order to get the accurate final statistical and reporting results for end-users [69].



- Account Numbers
  - Account Number
  - American Express Credit Card Number
  - Bank Account Number
  - Credit Card Number
  - Diners Club Credit Card Number
  - Discover Credit Card Number
  - JCB Credit Card Number
  - MasterCard Credit Card Number
  - Visa Credit Card Number
- Address Information
  - Address
  - Company Name
  - Computer Address
  - Country Code
  - E-mail Address
    - Auto-generated email name
    - Formatted email name
    - TRANS EML Function
  - Host Name
  - Personal Address
  - TCP/IP Address
  - URL
  - US State Code
  - US Zip Code
  - US Zip Code plus 4 Digits
- Data Manipulation
  - Concatenated Expression
  - Currency Function
  - Identity Function
  - Numeric Expression
  - Oracle Sequence Function
  - Propagate Key Function
  - Random Function
  - Sequential Function
  - Sequential number generator
  - Serial Function
- Data Mapping
  - Boolean Constant
  - Date Time Literal
  - Exit Routines
  - Hexadecimal Literal
  - NULL Expression
  - Special Register
  - String Literal
  - Substring
- Identification Number
  - Canadian SIN
  - Drivers License
  - French INSEE
  - Italian Fiscal Code
  - Passport Number
- Personal Identification Number
- Spanish FIN
- UK NINO
- US Social Security Number
  - TRANS NID Function
  - TRANS SSN Function
- Lookup Privacy Functions
  - Hash Lookup Function
  - Lookup Function
  - Random Lookup Function
- Personal Information
  - Age
  - Birth Date
  - Date
  - Gender Code
  - Personal Name
  - US Phone Number
- Shuffle Function

**Figure 2.7: A sampling of the IBM Optim data masking algorithms [68]**

- Also, many personal sample projects attempted to build some ad-hoc masking procedures, functions, and simple components to be used by other developers; the result was hard to customize and modify the codes and took a long time to standardize the process [20][21][58].

### 2.5.2 Reversible Data Masking Techniques

- Ricardo and et al. (2011), investigated the best database encryption solutions to protect sensitive data. However, given the volume of data typically processed by DW queries, the existing encryption solutions heavily increase storage space and introduce very large overheads in query response time. They proposed a data masking solution for numerical values in DWs based on the mathematical modulus operator (MOBAD), which can be used with an extra software application layer (not embedded with a BI platform) [18].
- Sarada and et al. (2015), provided a few new approaches for data masking, such as, Min-Max Normalization which performs a linear mapping of the original data

into a new range with lower and upper limits matching the range of attribute under consideration. The main advantage of this approach is that it maintains accuracy between the original and the masked data. Also, using a Fuzzy Based Approach creates fuzzy sets based on gradual assessment of data using a S-Shaped membership function, which they claimed is efficient to preserve the privacy and maintain the relationship with the original data. Furthermore, they suggested a Rail-fence Method, which is mostly applied to categorical data where the original data is written row/column-wise and the transformed data fetched by traversing along column/row-wise respectively. The last proposed method was Map Range (Rosetta Code), which is used for numerical data and performs mapping of original data to a range (mostly for mapping large values to a small range) given by the user. This method is used for numerical data. However, the author kept all the works in theory without any practical software framework. [35]

- Krishnamurty et al. (2007), provided a method for data shuffling to preserve data confidentiality. The method comprises masking of particular attributes of a dataset that are to be preserved in confidentiality, followed by a shuffling step comprised of sorting the transformed dataset and a transformed confidential attribute in accordance with the same rank order criteria. For normally distributed datasets, transformation achieved by general additive data perturbation, followed by generating a normalized perturbed value of the confidential attribute using a conditional distribution of the confidential and non-confidential attribute. In another aspect, a software program for accomplishing the method had provided. They claimed that their method provides greater security and utility for the data, and increases user comfort by allowing the use of the actual data without identifying the origin. However, they used a complex multi-pass statistical process to mask and retrieve the original data [12].

### 2.5.3 Summary

The literature review on Data Masking techniques is summarized in Table 2.2.

**Table 2.2: Literature review summary**

| Proposed Masking Technique | Reversible/ Irreversible | Static / Dynamic | Has Framework | Built-in BI or DBMS (not third-party application |
|---|---|---|---|---|
| Shuffle & SBLM (Ravikumar et al. 2011) | Irreversible | Static | Yes | No |
| Numeric Alteration, Format Preserving Encryption, Random Substitution (Min Li and et al. 2014) | Both | Static | Yes | No |
| Oracle Corp., Data Masking Techniques | Irreversible | Static | Yes | Yes |
| Microsoft SQL, Data Masking Techniques | Irreversible | Dynamic | Yes | Yes |
| IBM , Data Masking Techniques | Irreversible | Static/ Dynamic | Yes | Yes |
| Other Ad-hoc data masking stand-alone projects | Irreversible | Static | No | No |
| Ad-hoc Masking Technique for Numeric data (Ricardo and et al. MOBAD technique. 2011) | Reversible | Static/ Dynamic | Yes | No |
| Shuffling (Krishnamurty et al. 2007) | Reversible | Static/ Dynamic | No | No |
| Min-Max Normalization, Fuzzy Based, Rail-fence, Map Range (G Sarada and et al. 2015) | Reversible | Static/ Dynamic | No | No |

Based on the implications of the above literature review summary, a new built-in data masking framework with a new reversible masking technique is herein proposed that works in the business intelligence platform. Moreover, it considers the security-utility trade-off requirements of the sensitive data with or without using a third-party application.

# Chapter 3

# 3 Data Masking Framework (iMaskU): A Classification Module (Identify & Map)

In this chapter, a practical data classification module for a built-in data masking framework (iMaskU- Identify, Map, Apply, Sign, Keep testing, and Utilize) is proposed, and the first and second components (Identify + Map → Data classifier module) of the proposed BI based secure data masking framework to protect the sensitive data are presented. The main objectives of developing the data classification module are to discover sensitive data, suggest the best-fit masking solution, discard unneeded sensitive data, meet compliance requirements, and provide an integrated solution within a BI platform. The masking solution is addressed in Chapter 4.

In addition, in this chapter, our hypothetical data is based on Canadian healthcare data that uses many standard databases to collect data from hospitals, primary healthcare, specialized services, community care, and pharmaceutical care [23]. The BI system in healthcare stores a huge amount of Personally Identifiable Information (PII), Protected Health Information (PHI), and other organizational information. This information is extracted from different types of standard healthcare databases, which are briefly described in Table 3.1 below:

**Table 3.1: A list of the most standards healthcare databases in Canada**

| | **Database Name** | **Full Name** | **Description** |
|---|---|---|---|
| **1** | **DAD** | "Discharge Abstract Database" | Contains demographic, administrative and clinical data (diagnosis and treatments) for all acute care discharges. The data is reported for completed cases only (discharges). Hospitals do not report on cases that are still being treated [24]. |
| **2** | **NACRS** | "National Ambulatory Care Reporting System" | Captures ambulatory care visits activity. It is completed at the hospital using different sources such as, emergency department information systems (EDIS), Admission /Discharge Transfer (ADT) systems, patient records, physician notes and |

| | | | |
|---|---|---|---|
| | | | laboratory, MIS functional centre, diagnostic imaging, high cost outpatient clinics, and day Surgery [25]. |
| **3** | **CCRS (CCC)** | “Continuing Care Reporting System” | The information contained in the “Complex Continuing Care (CCC)” (also known as a chronic care) and Long Term Care tables developed by the Canadian Institute for Health information (CIHI) and used by the Ontario Ministry of Health and Long Term Care [26]. |
| **4** | **NRS** | “National Rehabilitati-on Reporting System” | The information contained in the Inpatient Rehabilitation tables was obtained from the “National Rehabilitation System (NRS)” developed by the Canadian Institute for Health information (CIHI) in consultation with the Ontario Ministry of Health and Long Term Care. [27] |

## 3.1 A Data Masking Framework for the BI Platform

A BI platform consists of many components. Fig. 2.1 shows the areas in which PII and PHI are accessible within a data warehouse and staging area [28][29], as they are considered the core components of the BI platform that are an integrated repository derived from multiple data sources (operational and legacy) in the production environment.

Therefore, the right masking or de-identification techniques to protect sensitive data and preserve the data utility must be chosen to automate the manual processes that lead to designing a built-in data masking framework within the BI platform. In this chapter, we propose a conceptual design of the framework, which is called iMaskU. This framework consists of six components (Identify, Map, Apply, Sign, Keep testing, and Utilize). The first two components are the most critical and important that we cover in this chapter. (Note: Native encryption techniques are avoided due to complex calculations, increased storage space, and large overheads in query response time).

With these challenges of compromising privacy and utility of health data in mind, this chapter focuses on identifying the best data masking techniques that fit the sensitive health data attributes to maintain data privacy and the quality of data analytics. In addition to designing the required built-in data masking framework/model to be available in the early integration service stage (Extract, Transform, and Load - ETL), it must function in the analytical layer to re-identify the sensitive data in an easy way. Fig. 3.1 depicts the general context diagram of the data masking framework within the BI platform.

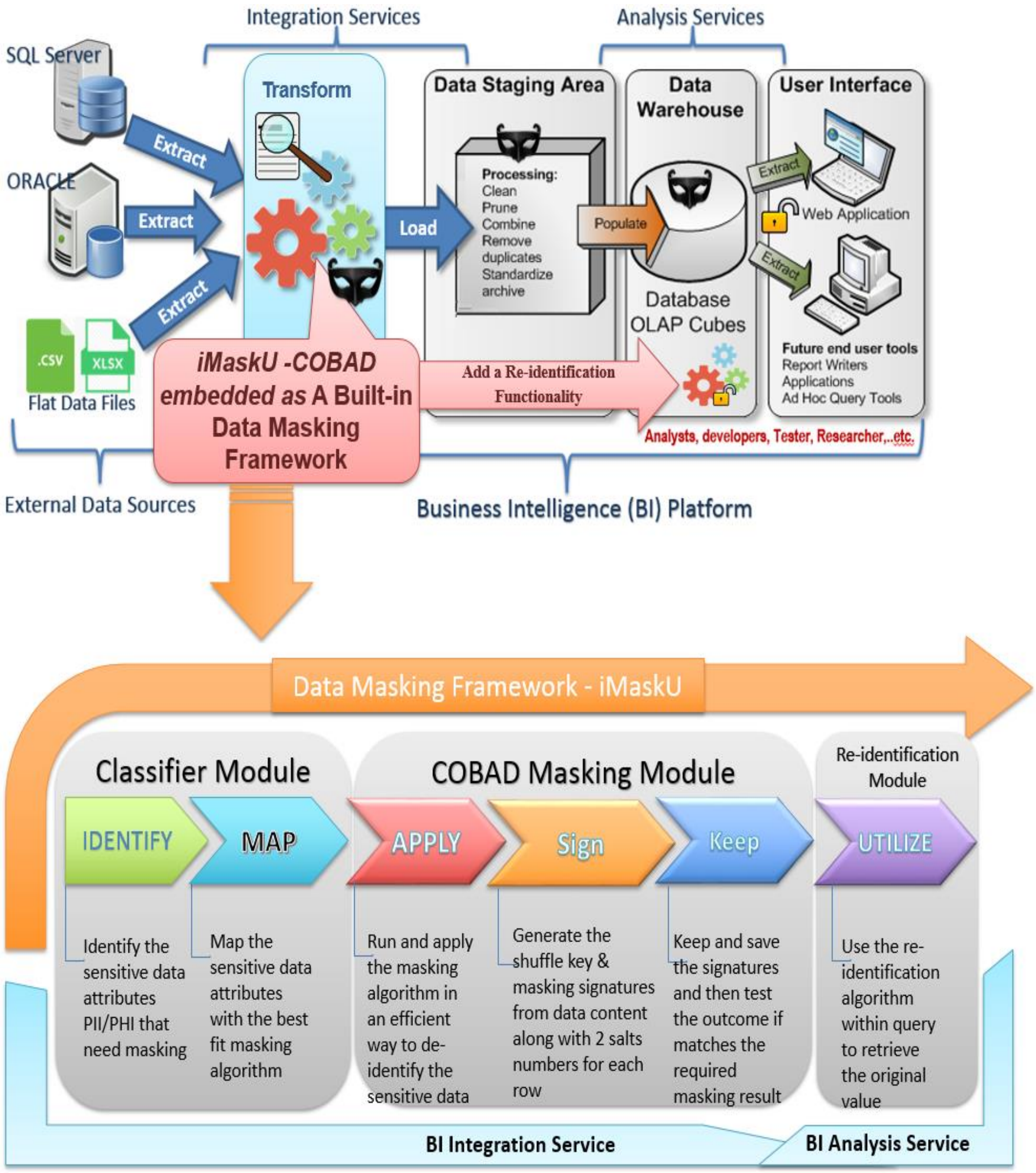

Integration Services
Analysis Services
SQL Server
ORACLE
.CSV
XLSX
Flat Data Files
Extract
Extract
Extract
Transform
Load
Data Staging Area
Processing:
Clean
Prune
Combine
Remove
duplicates
Standardize
archive
Populate
Data
Warehouse
Database
OLAP Cubes
User Interface
Extract
Web Application
Extract
Future end user tools
Report Writers
Applications
Ad Hoc Query Tools
iMaskU -COBAD
embedded as A Built-in
Data Masking
Framework
Add a Re-identification
Functionality
Analysts, developers, Tester, Researcher,..etc.
External Data Sources
Business Intelligence (BI) Platform
Data Masking Framework - iMaskU
Classifier Module
COBAD Masking Module
Re-identification
Module
IDENTIFY
MAP
APPLY
Sign
Keep
UTILIZE
Identify the sensitive data attributes PII/PHI that need masking
Map the sensitive data attributes with the best fit masking algorithm
Run and apply the masking algorithm in an efficient way to de-identify the sensitive data
Generate the shuffle key & masking signatures from data content along with 2 salts numbers for each row
Keep and save the signatures and then test the outcome if matches the required masking result
Use the re-identification algorithm within query to retrieve the original value
BI Integration Service
BI Analysis Service

**Figure 3.1: General context diagram of the proposed data masking framework iMaskU within the BI platform**

In this research study, the proposed masking framework will be shortened to the acronym iMaskU as shown in Fig. 3.1, consisting of six main components to automate the data masking/ de-identification processes. These components are:

- **Identify:** This component involves identifying sensitive data attributes that need masking based on the recognition of PHI/PII attributes by using the masking format library (pre-defined business rules).
- **Map:** This component involves mapping the selected PHI/PII data attributes with the right masking formula (algorithm) with allows an automated system. The way each attribute is masked or de-identified will depend on the type of data (e.g., the way a postal code is masked is totally different from the masking of date of birth or health card number).

  **Note:** The Identify and Map components are used to form the Classifier Module that automates the selection process of the masking algorithm and associates it to the sensitive data attribute.
- **Apply:** This component involves running and executing the masking algorithm in an efficient way. This technical task describes the automation of masking techniques and the preview list that is being generated before the execution.
- **Sign:** This component involves generating a shuffle key ($K_{SH}$) for each data extract batch and the masking signature key ($K_M$) for each record, then saving it into the destination dataset.

  **Note:** In this study, a technique based on a reversible function approach is proposed to realize and fulfil the requirements of the analytical model. Our COBAD Module (Content Based Masking Technique) is the core of the data masking framework and encompasses the Apply and Sign components which apply the best-fit algorithm to the specific data attribute.
- **Keep:** This component involves keeping and saving the signature as a trap-door and then testing the masking result if being applied successfully through comparing the preview list with the destination masked data set (health card no, postal code, diagnosis codes, etc.), as well as conducting re-identification risk assessment.

- **Utilize:** This module involves using re-identification methods in the analysis service stage for querying the de-identified data attributes to retrieve the original data to get the right result.

Once the first four automated components' tasks are completed, the physical anonymized dataset will be automatically created in the data staging area which contains the masked value of the sensitive data as a fully protected version. This chapter focuses on the first and second components of the framework to build the automated masking classification component as described in section 3.5.

## 3.2 Classification and Attributes of Sample Data (Health Data)

Classifying data in information security is the process of categorizing data assets based on nominal values according to its sensitivity (e.g., impact of applicable laws and regulations). For example, data might be classified as: public, internal, confidential (or highly confidential), restricted, regulatory, or top secret [77]

As we mentioned at the beginning of this chapter, we are going to use health data as our use case to classify the privacy levels of provided data that would make it easier for a user to determine how best to keep data safe in the DW as shown in Fig. 3.2 [30].

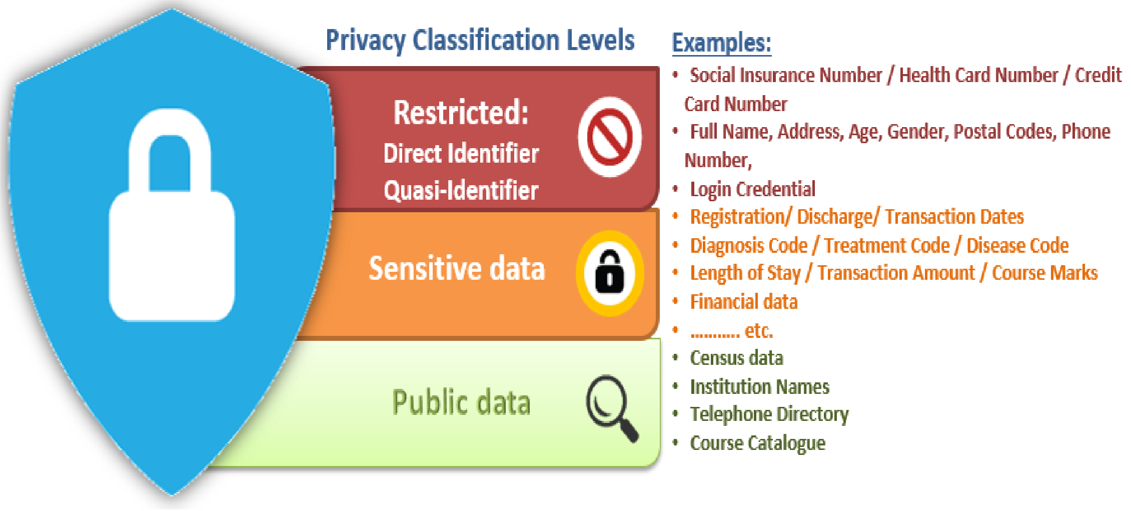

Privacy Classification Levels
Restricted:
Direct Identifier
Quasi-Identifier
Sensitive data
Public data
Examples:
• Social Insurance Number / Health Card Number / Credit Card Number
• Full Name, Address, Age, Gender, Postal Codes, Phone Number,
• Login Credential
• Registration/ Discharge/ Transaction Dates
• Diagnosis Code / Treatment Code / Disease Code
• Length of Stay / Transaction Amount / Course Marks
• Financial data
• .......... etc.
• Census data
• Institution Names
• Telephone Directory
• Course Catalogue

**Figure 3.2: Three levels of data classification system**

Public data can be disclosed to the public without affecting the corporate data security. Sensitive data needs to be kept confidential with authorized user access. Restricted data is highly confidential in nature and carries significant risk from unauthorized user access such as PHI and PII [55].

All masking techniques (irreversible and reversible de-identification) are important methods that can be used to minimize the privacy disclosure risk associated with saving, using, or even sharing data containing PII and PHI [31], which contain direct identifiers, quasi-identifiers, and other sensitive attributes as described in Table 3.2.

**Table 3.2: Hypothetical health dataset to illustrate a number of attributes.**

| Direct Identifiers | | | Quasi-Identifiers | | | PHI Sensitive Data Attributes | | | | Other Attributes | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Health Card No | Patient Name | Phone No. | Sex | Age | Postal Code | Admission Date | Diagnosis Code | Treatment Code | Length of Stay | Program Area | Transferred to Institution |
| 11111 | John Smith | 555-6667 | M | 52 | N6666 | 2017/05/05 | E761 | 1ZZ35HAB7 | 7 | INTERNAL MEDICINE | Hospital 1 |
| 12121 | Gill Brown | 555-4441 | M | 41 | N7666 | 2017/05/22 | J353 | 1FR89WJ | 4 | EMERGENCY MEDICINE | Hospital 2 |
| 87654 | Helen Arnold | 222-3332 | F | 38 | L4222 | 2017/05/12 | S72090 | 1YM27JA | 5 | CARDIOLOGY | - |
| 66442 | Kathy Plank | 555-6653 | F | 64 | M0011 | 2017/05/19 | J4500 | 1WA03JAF | 8 | ONCOLOGY | Hospital 3 |
| 23458 | Lisa Last | 554-1112 | F | 48 | N7766 | 2017/05/11 | S82300 | 2NM71BAB | 4 | INTERNAL MEDICINE | - |
| …….. | ……….. | …………. | … | … | …….. | ……………. | ……….. | …………….. | …. | ………………. | ………………….. |

Before navigating the details within the masking framework, we need to have a look at the health data attributes. The purpose of the "Personal Health Information Protection Act (PHIPA, Ontario 2004)" is to establish rules for the collection, use, and the disclosure of Protected Health Information (PHI) about individuals to protect the confidentiality of the information and the privacy of individuals, in order to facilitate the effective provision of healthcare [32].

Protected Health Information (PHI), means "identifying information about an individual in oral or recorded form that relates to the physical or mental health of the individual; relates to the providing of healthcare to the individual; relates to payments or eligibility for healthcare, in addition to, plan of service within the meaning of the Home Care and

Community Services Act" [32]. Note that, identifying information means, information that identifies an individual or for which it is reasonably foreseeable in the circumstances that it could be utilized, either alone or with other information to identify an individual [32].

Based on the definition of PHIPA, HIPAA ("Health Insurance Portability and Accountability Act"), or any other health privacy regulation, it is crucial to protect healthcare data whether from the patient perspective or the provider perspective. In this work, we use the most common database to understand the sensitive health data attributes that need to be obfuscated and protected from disclosure to internal and external users.

Our approach is to analyze inpatient discharge data (acute care) which is obtained from the "Discharge Abstract Database (DAD)" system, developed by the Ministry of Health of Ontario and the Canadian Institute for Health Information (CIHI) [33][47]. This database contains many data tables that include detailed patient-level "abstracts" for all type of hospital care services in one standardized source as a patient journey within hospitals' units.

The data collections contain demographic, clinical, and administrative data for all acute care discharges in Ontario [24][55].

To build the classification module of the data masking framework, a thorough analysis has been conducted against the DAD dataset and its attributes working with the subject matter experts based on (A) the enterprise reporting perspective, (B) the data definition documentations, and (C) the expert determination method as follows:

A. **Enterprise Reporting Perspective:** It is a reporting process involves providing substantial timely information in effective way to different management level of end-users in the organization to help them monitor the daily/monthly/quarterly/annually operational performance and allow higher management staff to make informed business decision. As I'm working as a Business Intelligence and Decision Support Specialist in heathcare domain interacting with the IT experts and Privacy officers, I have access to and be able to comment on the enterprise reporting. For example, all organizations in healthcare have variety of enterprise and ad-hoc reports that includes different type of data attributes as follows:

- **Daily Utilization and Case Costing Reports and dashboards:** These reports provide informative metrics on the health services usage for the illness and conditions the patients diagnosed with, and the costs associated with those diagnoses. Multiple heterogeneous data sources from across clinical and financial systems have used to generate these reports/dashboards on timely basis in which start from high-level aggregation whether quarterly or monthly and can be drilled-down to daily, hourly, and patient levels. The used data attribute (Admission DateTiem, Discharge DateTime, Discharge Fiscal Year, Discharge Fiscal Quarter, Discharge Fiscal Month, Diagnosis Code, Gender, Age, PostalCode, Hospital, Geo-Location, Intervention Code, Number of Discharged, Re-Admit within 30-days, Cost, Chronic Condition, High Cost Patient, …. etc [60].
- **Clinical Performance for Hospital In-patient and Emergency department (ED):** Measuring and reporting on how the hospital acute care in-patient and emergency department are performing using different key indicators, such as, In-Patient Length Of Stay (LOS), ED Wait Time, Wait Time To Surgery, Patient Safety, Frequent ED Visit Patients. All these indicators need to use the most listed attributes in the Table 3.3 on different time interval down to daily basis [61][62].
- **Ad-hoc Patient Details Report:** It shows the patient profile and journey across the care services using patient health card number, chart number, Full name, Hospital names, unit names, age, gender, service date stamps, LOS, providers, diagnosis codes, etc. This ad-hoc report may runs on hourly, daily, monthly basis [63].

**B. Data Definition Documents:** Based on my daily involvement on BI project operation and documentation, I’m sharing in this work some of them to support our classification analysis [64]. Using data definitions document from different BI analytics projects implementation identify the sensitive attributes whether direct identifiers, quasi-identifier, or others [65] , these documents also known as the Metadata Repository that allows users to reference and get context on all of the artifacts and data accessible in the BI analytics platform. Whether casual report viewer or a decision support analyst, the data definitions repository allows to access descriptions of available BI reports right down to how indicators and measures are calculated in the BI to facilitate the understanding and analysis of the BI data and artifacts.

All the ad-hoc reporting and analysis tools are offered by the BI analytics platform; this repository will be a valuable internal reference for analysts who will want to create their own reports and identify the usage of the sensitive attributes. Subsequently, as the BI platform grows with new data sets and artifacts, the IDS Data Definitions repository will expand accordingly to accommodate new data definitions.

For example the data definition for the Patient Days Weekly Report includes: parameters (Hospital, Fiscal Year, Reporting Week]; Groupings (each row in the report represents the data for one hospital, further divided in sites, bed type, and ward, …etc); Fields (in-patient days for ALC patients, active patient); Data source (DAD Acute care In-patien) [63].

Depending on the analysis of the enterprise reporting indicators as described above, and my experience on data definitions and its related reporting style including the time interval, we summarized our results in a Table 3.3 as shown below. The sign (x) shows the usage of data attributes within specific time intervals of the required reports. Moreover, each data attribute has been categorized as to whether it holds sensitive data (PHI or PII), or non-sensitive data (-). Note: Attributes with (*) signs will be analyzed in more details.

**Table 3.3: Data attributes of Discharge Abstract Database (DAD).**

| | | | | Reporting Time Intervals | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Data Attribute** | **Data Type** | **Sample** | **PHI/ PII** | **Annually** | **Quarterly** | **Monthly** | **Weekly** | **Daily** | **Hourly** |
| Health Card Number * | Numeric | 1234 567 890 | PHI | x | x | x | x | x | |
| Visit Number/ Encounter Number | Numeric | 123456789 | PHI | x | x | x | x | x | |
| Credit Card Number | Numeric | 1234567890123 | PII | | | | | | |
| Full Name | Text | John Smith | PII | | | | | x | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Sex | Text | M | PII | x | x | x | x | x | x |
| Birth Date * | Date Time | 1975/06/01 | PII | x | x | x | x | | |
| Postal Code * | Text | N0N1J2 | PII | x | x | x | x | x | |
| Patient Municipality Code/ Resident Code * | Numeric | 0501 | PII | x | x | x | x | x | |
| Admit Date * | Date Time | 2015/12/01 | PHI | x | x | x | x | x | |
| Admit Time | Date Time | 9:30 PM | PHI | | | | | x | x |
| Admission Category | Text | LIFE THREATENING CONDITION/URGENT /IMMEDIATE ASSESSMENT | PHI | x | x | x | x | x | |
| Facility (Hospital) | Text | (0966) BLUEWATER HEALTH | - | x | x | x | x | x | |
| Discharge Date * | Date Time | 2015/12/10 | PHI | x | x | x | x | x | |
| Discharge Time | Date Time | 7:20 PM | PHI | | x | | | x | x |
| Discharge Status | Text | DISCHARGED TO HOME WITH NO SUPPORT SERVICES | PHI | x | x | x | x | x | x |
| ICD10 Block Diagnosis (Dx) * | Text | (N80-N98) NONINFLAMMATORY DISRD OF FEMALE GEN TRAC | - | x | x | x | x | x | |
| ICD10-CA Diagnosis Dx Code * | Text | N800 | PHI | x | x | x | x | x | x |
| CCI Intervention Tx Code * | Text | 3ZZ20WC | PHI | x | x | x | x | x | |
| CCI Intervention Tx Start Date * | Date Time | 2015/12/15 | PHI | | | | | | |
| CCI Intervention Tx Start Time | Date Time | 8:35 AM | PHI | | | | | | |
| Case CMG plus Code* | Text | 043 | PHI | x | x | x | x | x | |
| CMG plus Atypical Status | Text | TYPICAL - TYPICAL | - | x | x | x | x | x | |
| CMG plus ELOS | Numeric | 6.5 | - | x | x | x | x | x | |
| CMGplus RIW | Numeric | 0.5437 | - | x | x | x | x | x | |
| HIG Code | Text | (202) Arrhythmia wo Cor Angio | PHI | x | x | x | x | x | |
| HIG Weight | Numeric | 1.1297 | - | x | x | x | x | x | |
| HIG ELOS Days | Numeric | 4.4 | - | x | x | x | x | x | |
| Acute LOS * | Numeric | 6 | PHI | x | x | x | x | x | x |
| ALC LOS * | Numeric | 3 | PHI | x | x | x | x | x | x |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Total LOS * | Numeric | 9 | PHI | x | x | x | x | x | x |
| Transfer To Institution | Text | (4417) BLUEWATER HEALTH-SARNIA GENERAL SITE | - | x | x | x | x | x | |
| Transfer From Institution | Text | () UNKNOWN | - | x | x | x | X | x | |
| Emergency Wait Hrs * | Numeric | 1 | PHI | x | x | x | X | X | |
| Died During Intervention Flag | Boolean | FALSE | PHI | x | x | x | | | |
| Death in Special Care Flag | Boolean | FALSE | PHI | x | x | x | | | |
| Main Patient Service | Text | GENERAL MEDICINE | PHI | x | x | x | X | X | |
| Main Provider Service | Text | CRITICAL CARE MEDICINE | PHI | x | x | x | | | |
| Chronic Condition Diagnosis Dx | Text | *Not Applicable | PHI | x | x | x | X | | |

C. **Expert Determination Method:** This method helps to provide the proper answers from whom examines the data, understands the privacy policies, and determines an effective means for anonymization/ de-identification that minimize the risk of re-identification. By attending and holding regular and ad-hoc meetings with many IT technical and privacy domain experts as they have the appropriate knowledge of and experience with generally accepted statistical and scientific principles, the questions have been prepared to collect facts and best practices and cover many key factors that we use to build our iMaskU data masking framework.

The following questions have been asked to the domain experts and their responses are documented below. These question focusing on “Which data masking method should be used within BI Platform?”, the answers were taken seriously into our consideration:

1. **Data Appearance and Usability:** Should we keep the masked data looks realistic as well as retain the data format and size?

   **Summary answer:** In contrast of encryption, the non-production environment including BI platform, highly recommend to keep the masked data in the same format and size, this leads to retain the look and feel of the text, numeric, and alpha-numeric data fields. In addition, to determine the usability of the data alongside with the appearance without taking the statistical results accurate, we need to use the two most common ways, pseudonymization or format-preserving encryption. Alternatively, also, able to use substring masking (i.e partial field redaction, e.g., XXX-XX-1234, N6Gxxx) may be fit with SINs Number and postal code respectively. The recommendation is, if the requirement is realistic and functional masked data, do not go with the nulling, full redaction, hashing, encryption, and randomization, it is better try to use number and date ageing or sub-string manipulation within acceptable range to not affect negatively the result [66].

2. **Static versus Dynamic data masking approach:** Do we need masking data at rest or on the fly?

   **Summary Answer:** It is highly recommended to have data protection tools to mask the data at rest in the storage layer to prevent the internal inadvertently breaches. But this needs an effective trade-off between the security and utility of the data when it use for secondary purposes especially for prototyping research and analytics model, or for other purposes, such as application development, testing, training, and. On the other hand, the dynamic data masking can be best suited for read only case with keeping the original data unmasked, and easy way to apply role-based protection at the inquiry level.

3. **Reversibility (Re-Identification):** Need the original data restored at the inquiry level to get the correct results?

   **Summary Answer:** if the requirement is to permanently anonymize the sensitive data at the staging database and DW, keep looks realistic and usable, then we need to apply whether Format Preserving Encryption, reversible pseudonmization, custom formula, or tokenization techniques. In such cases, public and private keys can be employed. In this situation, the traditional non-reversible data masking techniques have been avoided.

4. **Uniqueness (Consistency):** Does the same original value always need to be replaced by the same value or not?

   **Summary answer:** If the data going to be joined on or grouped by the masked values, then the used masking algorithm must generate results which are unique and repeatable any time we apply it in order to maintain the referential integrity across all the data tables within staging data storage. This can be achieved by using the same encryption (FPE) or hashing algorithm with the same key (passphrase), i.e, patient health card number need to be consistent across all datasets in order to allow the analyst to track the patient journey across all services and hospitals. In this case other masking algorithm, such as substitution, shuffle, randomize, and pseudonmization techniques are not suitable to be used.

5. **Security Strength:** Does the masked data is secured enough to not be re-identified easily by using different attack scenarios?

   **Summary answer:** it needs to look inside the functionality of each algorithm and determine their crackability and asses that against the appearance and usability.

   For example, AES256 is stronger than AES128, SHA2 is stronger than SHA1, and all these stronger than Hexadecimal coding and decoding. In general, reversible techniques are relatively weaker than irreversible ones, but the down side with irreversible is lack of usability which is consider an integral part of the BI analytics platform. In the HIPPA safe harbor policy, the removal of key identifiers complies, however if we need to use the data for analysis, in this case we need a masking technique instead and a proof that this technique carry a low statistical likelihood of re-identification.

6. **Privacy Policies:** Which privacy policy and regulations that being used in workplace to comply with?

   **Summary answer:** Personal Health Information Protection Act (PHIPA) in Ontario, establish rules for the collection, use and disclosure of personal health information about individuals that protect the confidentiality of that information and the privacy of individuals in Ontario province [45].

   Furthermore, the Health Insurance Portability and Accountability Act (HIPAA), it is US regulation to remove/anonymize specific identifiers that lead to recover the individuals information.

   The Payment Card Industry Data Security Standard (PCI DSS), merchants and credit card processing companies need to comply with PCI-DSS that facilitates consistent measures for data security globally. *[Section 6.4.3, specifically prohibit the use of production data for test and development]* [46][58].

## 3.3 Identify the Sensitive Data Based on Business Rules and Regulation

The primary reason to protect the sensitive data in non-production environment is to comply with the data privacy rules and regulations, such as, Personal Health Information Protection Act (PHIPA), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), Personal Information Protection and Electronic Documents Act (PIPEDA), and others, that focus on safeguarding the most sensitive data and limit the access to them based on a need-to-know process. [45][46]

Once we have the prioritized list of data stores after taking into account all the risk factors defined by an organization, the right level of protection can be applied to the right dataset. The security team would be more effective if they can define this protection technique and specific tool to be used for each affected area. Such protection techniques could be automated or may require manual intervention by data owner/application owner.

Based on the collected information from and the knowledge that has been built on the domain of healthcare business rules, I am proposing a new framework on how to utilize reversible and irreversible data masking and de-identification techniques to protect sensitive health data that will not affect the data usability. This framework complies with the data privacy regulations that are being implemented in the most healthcare environments. Our built is relying on the following standards and best practices.

- Patient Names: based on HIPPA recommendation, this field needs to be removed totally, or based on Section 164.312 (ii) using encryption and decryption, these requirements are reliably accomplished with substitution and encryption components. [Hush Hush]
- Health Card Number (HN): It is a direct identifier and one of 18 types of HIPPA safe harbor in which recommend to be removed. However, in this case and based on expert determination, this field converts to a unique patient identifier (Masked HN) assigned to each patient that links his/her activity across all partner Health Service Providers (HSP). For this type of attribute, we will apply a one-way

masking (irreversible) method, such as one way hashing or PFE to preserving data format. [31]

- Chart Number and Visit Number: will be hashed to a numeric format, which is similar to the Health Card Number in the compliance to the HIPPA policy.
- Date of Birth (DOB: YYYY/MM/DD): It is consider a quasi-identifier, based on the expert determination, needs to be generalized by converting to Age Group (Text, i.e "40-44"), or by applying a custom transformational formula to get the Age Year (Numeric, i.e 42), and then using number alteration (i.e., +/- 3 year).
- Address: Consider as an quasi-identifier, HIPPA recommends, "All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP code, and their equivalent geocodes, except for the initial three digits of the ZIP code (Postal code)". Also based on the expert determination, many derived fileds will be generate, such as, Municipality code (4 digits Residence Code), County (first 2 digits of Municipality Code), Province, and Postal Code (first 3 characters). We can apply a de-identification algorithm such as FPE or custom formula based transformation. Postal Code can be rolled up to FSA level (first 3 characters) by applying the masked-out algorithm. Other geography information can be derived by using the Municipality and/or Postal Code, as shown as follows [24]:
    - From Municipality/ Residence Code will derive: Patient or Hospital Local Health Integration Network (LHIN), County (the first two digits), and Public Health Unit (PHU)
    - From Postal Code will derive: Forward Sortation Area (FSA - first three characters), Health Link, Census Division (CD), Census Sub-Division (CSD), Dissemination Area (DA), Statistical Area Classification (SAC), and Census Tract (CT).
- Activities' Date Stamps (i.e., Admission Date Time, Discharge Date Time, Surgery Date Time) are replaced with Fiscal Year, Calendar Year, Fiscal Quarter, Calendar Quarter, and Month. Beside this, date ranging masking can be applied to increase or decrease the date value within an acceptable time range (alteration within the same month or the same week). An additional de-identification

algorithm can be used, such as FPE or custom formula based transformations to maintain daily reporting values if needed.

- Numeric Valued Attributes, such as Length of Stay (LOS) in Acute Inpatient, or Emergency Wait Time in Hours, etc. Based on expert determination and HIPPA rules, these attributes might be masked using numeric alteration within a reasonable range, or using the reversible masking methods such as FPE or custom formula-based transformation.
- ICD 10 diagnoses' codes, based on expert determination and HIPPA privacy compliance rules, can be rolled up to the Category level (3 characters Non-PHI) by using masking out, and also can be grouped by chapter or block, in addition using de-identification algorithms for the entire code such as FPE, custom formula, or Pseudonymization transformations, as shown in Table 3.4 below [24].

**Table 3.4: The Privacy level of Diagnosis Code Attribute (Dx)**

| | | | |
|---|---|---|---|
| Chapter (23 chapters) | II or C00-D48 | Neoplasms | Non-PHI |
| Block (~250 blocks) | C00-C97 | Malignant Neoplasms | Non-PHI |
| Category: | C34 | Malignant Neoplasms of Bronchus & Lung | Non-PHI |
| Sub-Category | C340 | Malignant Neoplasms of Main Bronchus | PHI |
| ICD 10 Diagnosis (~15000 codes) | C3401 | Malignant Neoplasms of Main Left Bronchus | PHI |

- CCI intervention codes, based on HIPPA privacy compliance rules, can be rolled up to the Block level (first 3 characters, Non-PHI) by using masking out; they also can be grouped by the Chapter level, in addition to using de-identification algorithms for the entire code such as FPE or Pseudonymization transformations, as shown in Table 3.5 below [20].

**Table 3.5: The Privacy level of CCI Intervention Code Attribute (Tx).**

| Non-PHI | | PHI (Restricted) | | | |
|---|---|---|---|---|---|
| Section (broad type of | Group (anatomy- | Intervention (generic | Approach, technique, | Device, agent, method used | Tissue used (discrete |

| intervention, e.g., diagnostic, therapeutic) | driven) | procedure) | reason (discrete meaning) | (discrete meaning) | meaning) |
|---|---|---|---|---|---|
| # | AA | ## | AA | AA | A |
| 1 | TK | 93 | LA | XX | A |
| Amputation, Humerus | | | Using skin graft (for closure of stump) | | |

- Case Mix Group Plus (CMG+) Code: are the foundation upon which cases are classified into clinically relevant and statistically homogeneous groups to reflect patient's overall medical conditions and resource consumption. Based on the expert determination and HIPPA privacy compliance rules, these sensitive data attributes need to be de-identified by using FPE or Pseudonymization methods. The DAD Database contains CMG Plus information in a hierarchal pattern.

  In addition, other CMG Plus related data attributes could be considered as indirect identifier and need to be de-identified; these attributes can be listed as follows:
    - CMG Plus Factors: such as, CMG+ Comorbidity Level, Age Category, Flagged Intervention Tx Count, Intervention Tx Event, Out Of Hospital.
    - CMG Plus Atypical: Textual data
    - CMG Plus Resource Intensity Level (RIL)
    - CMG Plus Expected Length of Stay (ELOS)
    - CMG Plus Resource Intensity Weight (RIW)
- Other Textual data attributes: They are considered as indirect identifiers and should have the de-identification algorithms applied to the entire attribute such as FPE or Pseudonymization methods.

## 3.4 Define Masking Rules Based on the Regulation:

Broadly speaking, the security rule requires implementation of three types of safeguards [78]:

1. Administrative safeguard: It is administrative actions, policies and procedures, to manage the selection, development, implementation, and maintenance of security measures to protect PII/ PHI.

2. Physical safeguard: It is physical measures, policies, and procedures to protect a covered entity's electronic information systems and related buildings and equipment, from natural and environmental hazards, and unauthorized intrusion
3. Technical safeguard: It is the technology and the policy and procedures for its use that protect electronic PII/ PHI information and control access to it

Sensitive data must be secured in a manner consistent with PHIPA, HIPAA, PIPEDA, and PCI DSS policies as well as must not be able to be queried. Data that is queriable may be retrieved through use of different tools or by issuing a set of system instructions or tasks, as listed in the following hints and tips [79]:

- Classifier needs to ensure that an appropriate retention policy is implemented and maintained based on adherence to an information security policy through developing daily operational security procedures.
- DW needs to ensure that proper user authentication is implemented for staff, administrators, and external stakeholders.
- Needs to ensure that the direct identifiers are masked when accessed or displayed
- Ensuring there are no direct connections between DW and the Internet
- Ensuring that DW are maintained to secure configuration standards and are regularly tested for vulnerabilities.

The Personal Information Protection and Electronic Documents Act (PIPEDA) is the federal privacy law for private-sector organizations. Under PIPEDA, the following is protected as sensitive or Personally Identifiable Information (PII) [80]:

- Age, name, ID numbers, income, ethnic origin, or blood type
- Opinions, evaluations, comments, social status, or disciplinary actions
- Employee files, credit records, loan records, medical records, existence of a dispute between a consumer and a merchant, intentions (for example, to acquire goods or services, or change jobs)

## 3.5 Mapping the Sensitive Data

Given a clear analysis of masking methods for sensitive data in the previous section, the conceptual classification component of the masking framework as shown in Table 3.6 summarizes the potential required of the irreversible traditional masking and reversible masking techniques for DAD's data attributes. The sign (x) shows the usage of the appropriate data masking techniques for the selected attribute based on the analysis of the transformation method that fits the specific data attributes as categorized by whether it holds sensitive data (PHI or PII), or non-sensitive data (-).

**Table 3.6: Summary of applicable irreversible and reversible masking techniques for DAD's data attributes**

| | | | Irreversible Traditional Masking | | | | | Key-based Reversible Masking | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Attribute | Data Type | PHI/ PII | Date Breakdown (Year/ Qtr/ Month/ Age) | Alter/Grouping within range | Shuffle/ Substitute | Hashing (Numeric Format) | Masking Out/ Or NULL | format-preserving de-identification | Pseudonymization | Formula based Numeric Masking |
| Health Card Number | Numeric | PHI | | | | × | | × | | × |
| Visit/ Transaction Number | Numeric | PHI | | | | × | | × | | × |
| Credit Card Number | Numeric | PII | | | | × | × | × | | × |
| Full Name | Text | PII | | | × | | × | | | |
| Sex | Text | PII | | | | | | × | × | |
| Birth Date | DateTime | PII | × | × | | | | | | × |
| Email | Text | PII | | | | | × | × | | |
| Address | Text | PII | | × | | | × | × | | |
| PostalCode | Text | PII | | × | | | × | × | × | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Patient Municipality Code | Numeric | PII | | × | | | | × | | × |
| AdmitDate | DateTime | PHI | × | | | | | | | × |
| AdmitTime | DateTime | PHI | × | | | | | | | × |
| Admission Category | Text | PHI | | | | | | × | × | |
| Facility (Hospital) | Text | - | | | × | | | × | × | |
| Discharge Date | DateTime | PHI | × | | | | | | | × |
| Discharge Time | DateTime | PHI | × | | | | | | | × |
| Discharge Status | Text | PHI | | | × | | | × | × | |
| ICD10 Chapter Diagnosis Dx | Text | - | | | × | | | × | × | |
| ICD10 Block Diagnosis Dx | Text | - | | | × | | | × | × | |
| ICD10-CA Diagnosis Dx Code | Text | PHI | | × | | | × | × | × | |
| ICD10-CA Diagnosis Dx Desc | Text | PHI | | | × | | | x | x | |
| CCI Intervention Tx Code | Text | PHI | | × | | | × | × | × | |
| CCI Intervention Tx Description | Text | PHI | | | × | | × | × | × | |

## 3.6 Design of A Classification Module for Data Masking Framework

Given that clear modules for a data masking framework (iMaskU) in the previous section as well as the method for masking sensitive data while preserving data quality/utility has been selected as a primary problem for the classification component, the general practical approach of data masking framework can be outlined in brief as follows:

1. The in-scope data set is identified (DAD database).

2. The in-scope data set includes sensitive data which are determined to be masked to preserve their confidentiality.
3. Primary sensitive data attributes are identified.
4. The primary sensitive data attributes are classified according to sensitivity (PHI / PII/ - ).
5. The required data masking techniques and definitions are collected.
6. Appropriate masking techniques are selected from a pre-defined set to be applied to each data attributes based on the business rules exercised on the data.
7. The data being initially masked is profiled to detect invalid data and an extra information attribute will be added.

The classification module of the masking framework focuses on the first two components, which are "Identify" and "Map". The detailed processes of these two components have been setup and organized in the previous sections as shown in Fig. 3.3.

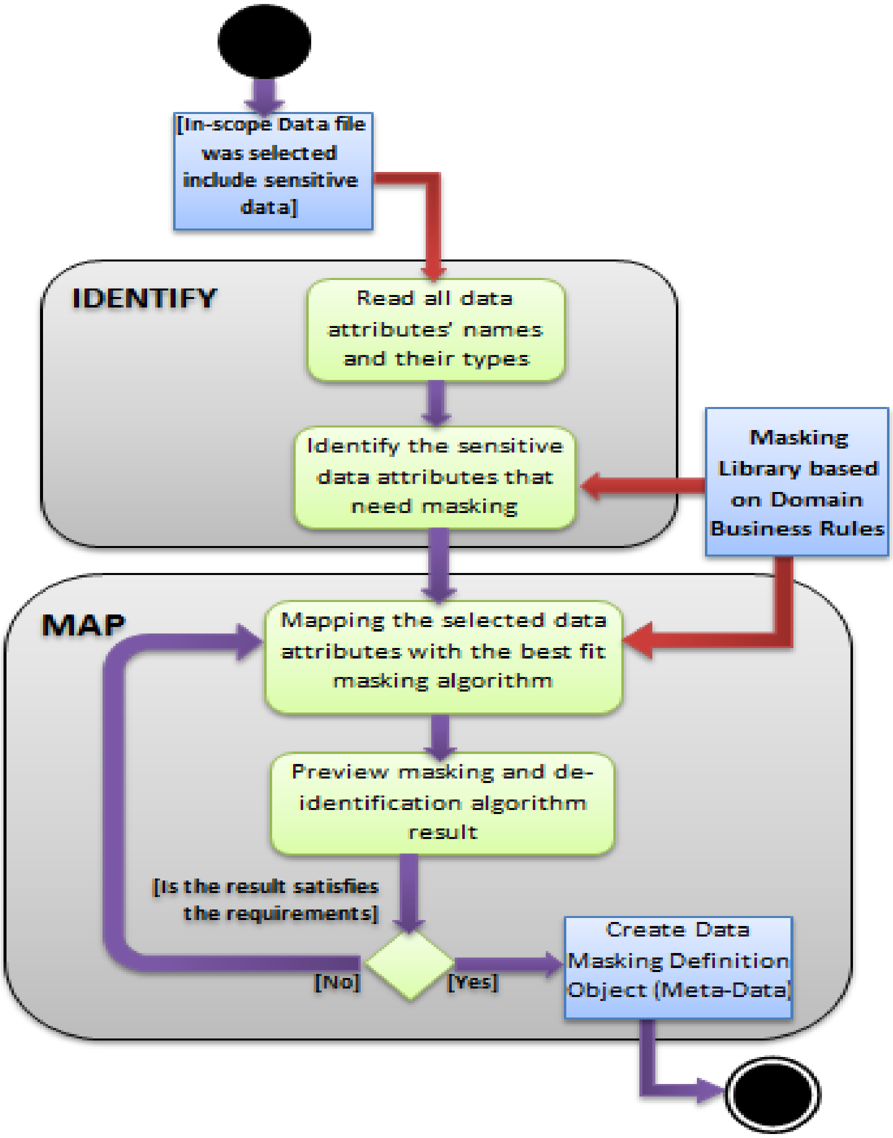

[In-scope Data file was selected include sensitive data]
IDENTIFY
Read all data attributes' names and their types
Identify the sensitive data attributes that need masking
Masking Library based on Domain Business Rules
MAP
Mapping the selected data attributes with the best fit masking algorithm
Preview masking and de-identification algorithm result
[Is the result satisfies the requirements]
[No]
[Yes]
Create Data Masking Definition Object (Meta-Data)

**Figure 3.3: UML Activity Diagram for the Classification component in Data Masking Framework ("Identify and Map")**

The classification module of iMaskU framework is the required way to:

- Automatically scan the column names of the data table, classify the PII and PHI sensitive attributes based on the business rules that relies on the privacy policies, and then assign the proper masking function to comply with the data privacy regulations.
- Determine and generalize quasi-identifiers to decrease the risk of re-identification

- Profile the data attributes within the extracted dataset and generate masking definition object (Meta-data).

## 3.7 Discussion

The classification module is proposed to identify the extracted sensitive data attributes from production datasets and provide the suggested data-masking algorithm that fit with the nature of the sensitive data based on the business rules. By taking into the consideration how to maintain the utility of the data for statistical analysis purposes while preserving the data privacy, a new reversible data masking technique (COBAD) has been integrated within the framework to provide a format preserving de-identification solution for different types of data formats. This will lead to protect data at rest within the enterprise data warehouse and meanwhile any internal or external attacks can be avoided.

No single universal solution addresses all privacy and identifiability issues. Rather, a combination of technical experiences and policy procedures are often applied to the de-identification task [67]. To accomplish the building of the classification module for the data masking framework, a thorough Q/A and analysis has been conducted to understand the sensitive attributes of the sample health dataset; the results derived from (A) the enterprise reporting perspective, (B) the data definition documentations, and (C) the expert determination method.

Using a sample of health data to build the classifier module, we rely on the data complexity to determine different levels of data privacy difficulties. Thus, the identify component is designed to minimize the manual data profiling time incurred by the user in addition to the computation cost incurred by an improper masking technique. The classification module adjusts the selection process of the masking technique based on the sensitivity of data using the business rules library.

In practical terms, the proposed classification module can be implemented in the data-masking framework of business intelligence platforms because its process simply relies on basic software requirements of both the ETL components and user interface.

However, the proposed classification module does not consider the full automation of the data profiling process, which leaves flexibility for the user to make the final decision to change the suggested technique as needed. Although the classification module can be implemented on different BI platforms, at this stage, it is suggested to be developed as a stand-alone prototype for implementation on a Microsoft SQL Server environment.

# Chapter 4

# 4 iMaskU COBAD Module (Apply, Sign, and Keep): COntent BAsed Data-masking Reversible Technique

Given a clear description of building the first module as a data classifier for iMaskU framework in the previous section as well as considering the method for masking sensitive data while preserving data quality/utility as a primary problem for the classification module. In this chapter, a second module (COBAD technique, Apply, Sign, and Keep) of the BI based data-masking framework is proposed to protect the privacy of data and prevent internal and external attackers from disclosing the original information at rest. In addition, it securely de-identifies the sensitive data using a well-defined mathematical formula.

In a data warehouse, it is difficult to map internal and external users to a subset of data. Consequently, the protection of data privacy while maintaining data utility is important, starting from the data warehouse to end-user reporting tools. Our objective is to design a data masking framework (iMaskU: Identify, Map, Apply, Sign, Keep, Utilize) for a BI platform to protect the data at rest, preserve the data format, and maintain the data utility at the querying level. A new reversible masking technique (COntent BAsed Data masking - COBAD) is developed and introduced to fit within the framework. The masking algorithm, which is based on the statistical content of the extracted dataset, is used to de-identify the sensitive attributes within the hypothetical data, so the masked data cannot be linked with specific individuals or be re-identified by any means.

The iMaskU framework (including COBAD masking module – Apply and Save signature) works to balance privacy protection and data utility in a corporate BI platform, especially within integration and analysis services. It is to be noted that it depends on the successful completion of a classification module; this module starts executing the algorithm to secure the sensitive information at the data warehouse.

## 4.1 COBAD Data Masking Solution

Current Data Masking (DM) is based on third party solutions used outside of BI platforms which are costly and require installation and extra labour time to administer the masking functionality to get the required result. The iMaskU framework introduces the COBAD as an evolved de-identification algorithm which is encapsulated in a component and embedded within the data integration development environment (ETL package – Extract, Transform, Load). This will mask data at rest to prevent data breaches from the beginning. COBAD is a new Data Masking technique that is not only stronger and more secure but also requires no programming skills for ETL developers and uses an algorithm to reduce development time on any given data masking project by ~80%.

### 4.1.1 What COBAD Does

A COBAD technique protects sensitive PII and PHI data from internal users who have access to the data staging area and DW within the BI platform. A COBAD also protects sensitive data from external hackers even if they get an off-line copy of the original dataset.

COBAD emphasizes the trade-off between data privacy and its utility by preserving the data format and type to keep it looking realistic for end users within the non-production environment (e.g., DW) by using a new reversible data masking technique in the context of maintaining the utility of data analytics as shown in Fig. 4.1. This trade-off has been demonstrated in our empirical assessment using simulated health data [34][58]. Also, COBAD helps to minimize the risk of an internal data breach by using a strong and secure masking algorithm that complies with privacy regulations.

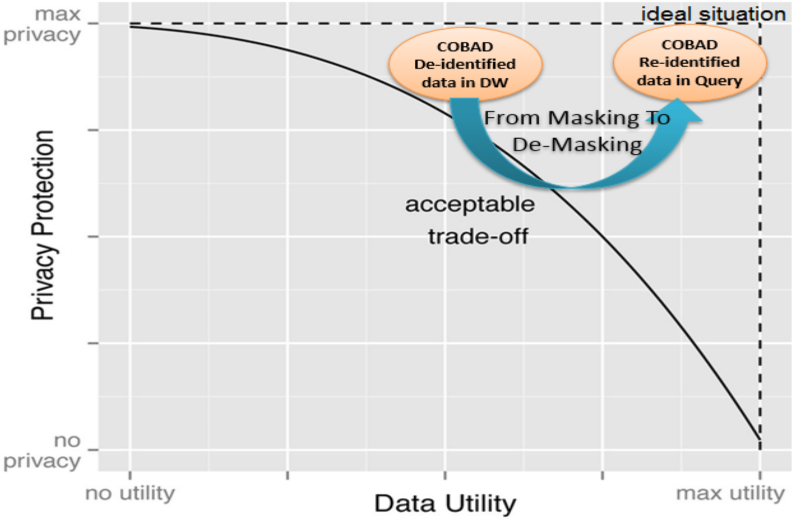
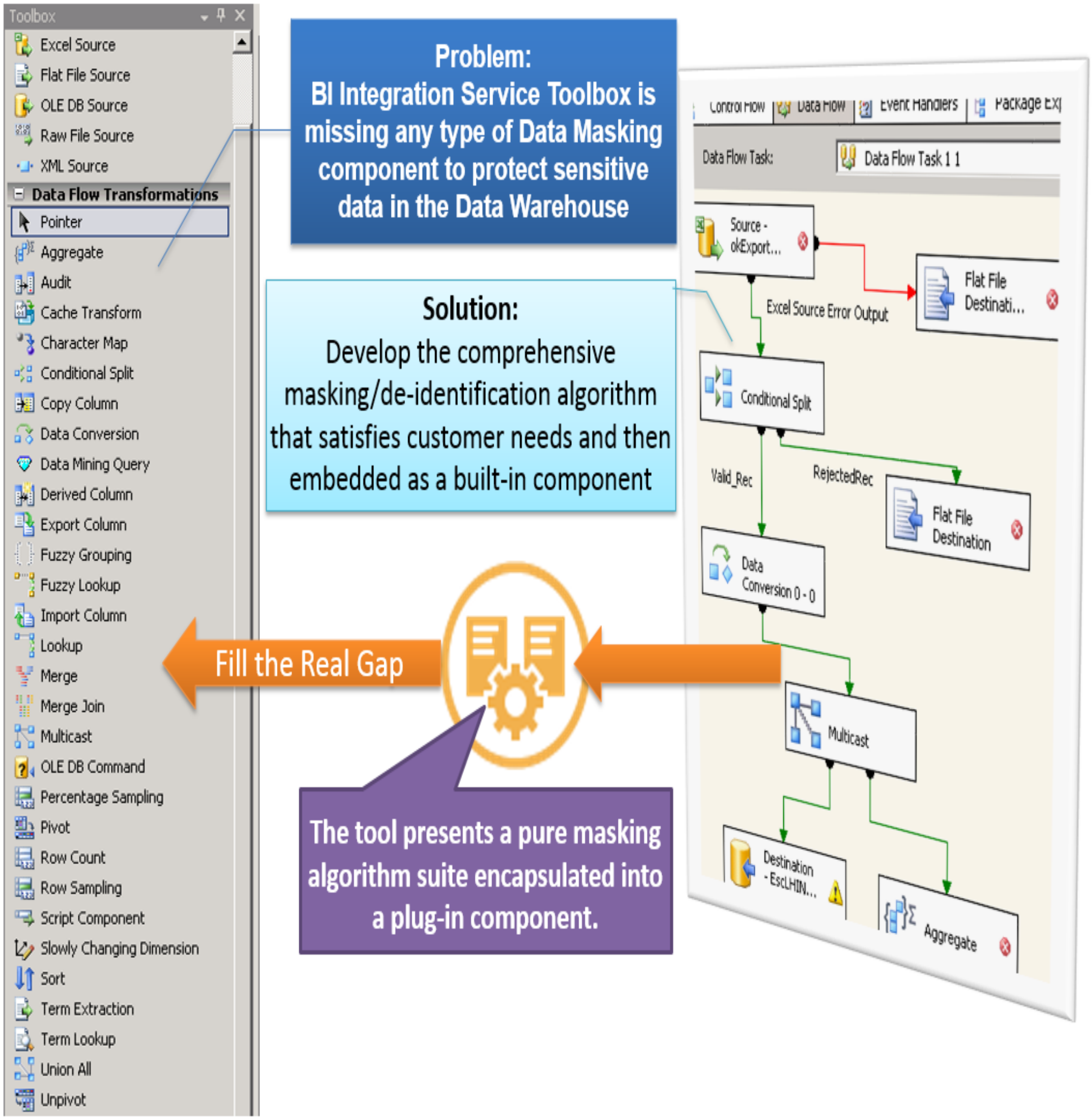


**Figure 4.1: iMaskU-COBAD lies above the acceptable trade-off area**

### 4.1.2 How COBAD Works

A COBAD technique is proposed to be a built-in component to a plug-in within the Integration Service toolbox used in any ETL process and applies a new reversible masking technique on loaded data to be saved into the destination data source as shown in Fig. 4.2.



Problem:
BI Integration Service Toolbox is missing any type of Data Masking component to protect sensitive data in the Data Warehouse

Solution:
Develop the comprehensive masking/de-identification algorithm that satisfies customer needs and then embedded as a built-in component

Fill the Real Gap

The tool presents a pure masking algorithm suite encapsulated into a plug-in component.

**Figure 4.2: iMaskU-COBAD Solution fills the gap of any BI Integration Service's Toolkit**

To retrieve the original data, a built-in function is needed to be added within DBMS. COBAD is a column-based masking algorithm for numeric, date, and alphanumeric data attributes, which is selected to be applied to each data element. It maintains the type, size, and format of the masked data attribute (Format-Preserving technique) to look realistic, furthermore it adds a functionality to re-identify the masked data at analysis services (querying/ reporting processes) as shown in Fig.4.3. [22][51].

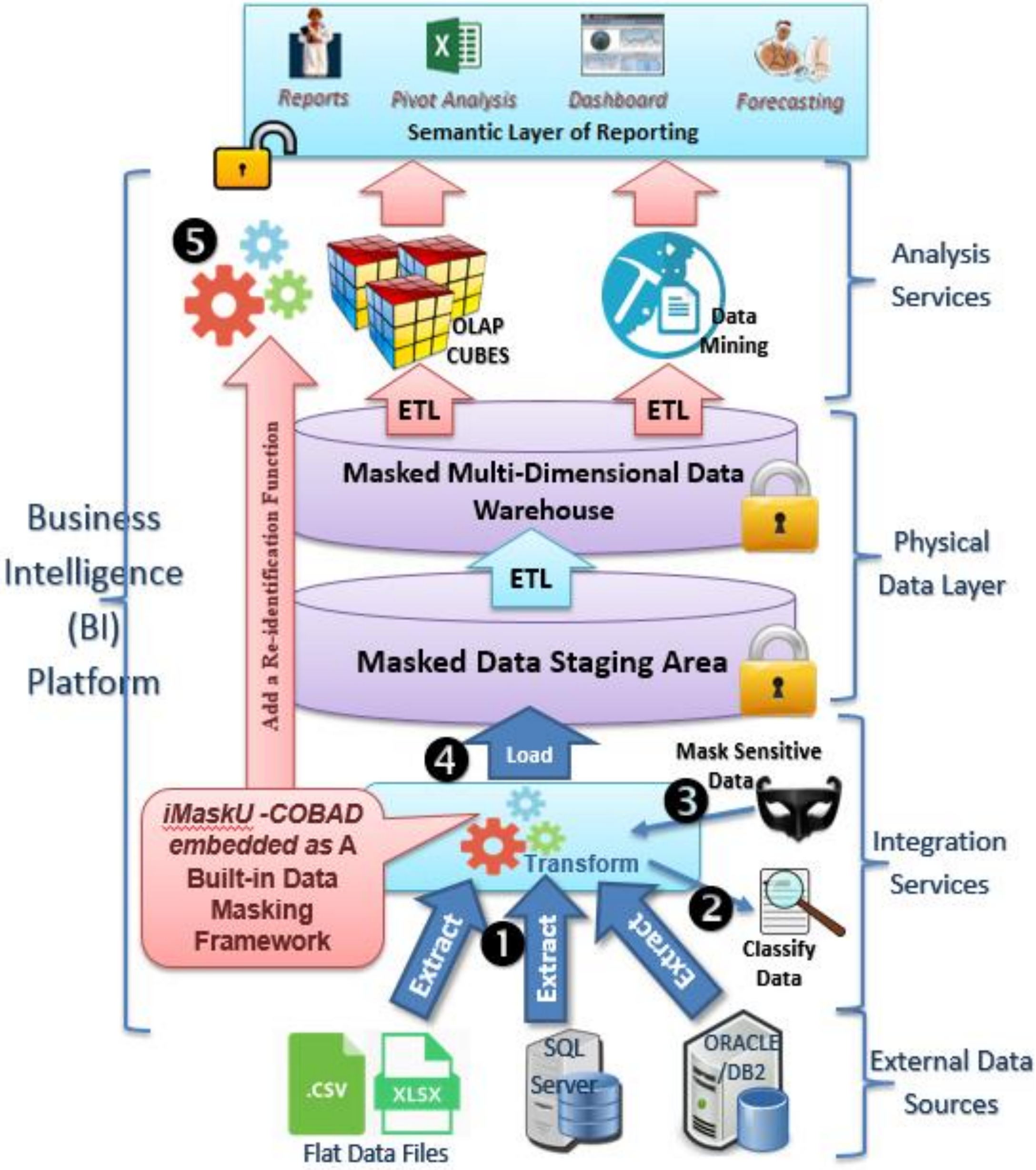


**Figure 4.3: iMaskU-COBAD as an embedded Framework within BI platform**

The benefits of the COBAD technique are as follows:

- Reduces development time with one product solution within integration services
- Presents a reversible masking algorithm (de-identification) suite encapsulated into a built-in component to de-identify the sensitive data, preserve the data format, and maintain the quality of data utility
- Adds a re-identification method as a built-in function to the querying process

- Embeds native integration into the BI Integration Service, masks data at rest, eliminating the need for third-party vendors' tools and interfaces
- Complies with the data privacy regulations and acts as defined in the privacy protection provisions of the Freedom of Information and Protection of Privacy Act (FIPPA), the Municipal Freedom of Information and Protection of Privacy Act (MFIPPA), and Health Insurance Portability and Accountability Act (HIPAA). [36][37]

## 4.2 COBAD Masking Algorithm

A request for the masking technique to minimize privacy disclosure and maintain data utility could bring a significant balance in this regard. With the recommendation of our iMaskU data masking framework, we aim to propose an acceptable column-based masking algorithm for numeric, date, and alpha-numeric data attributes.

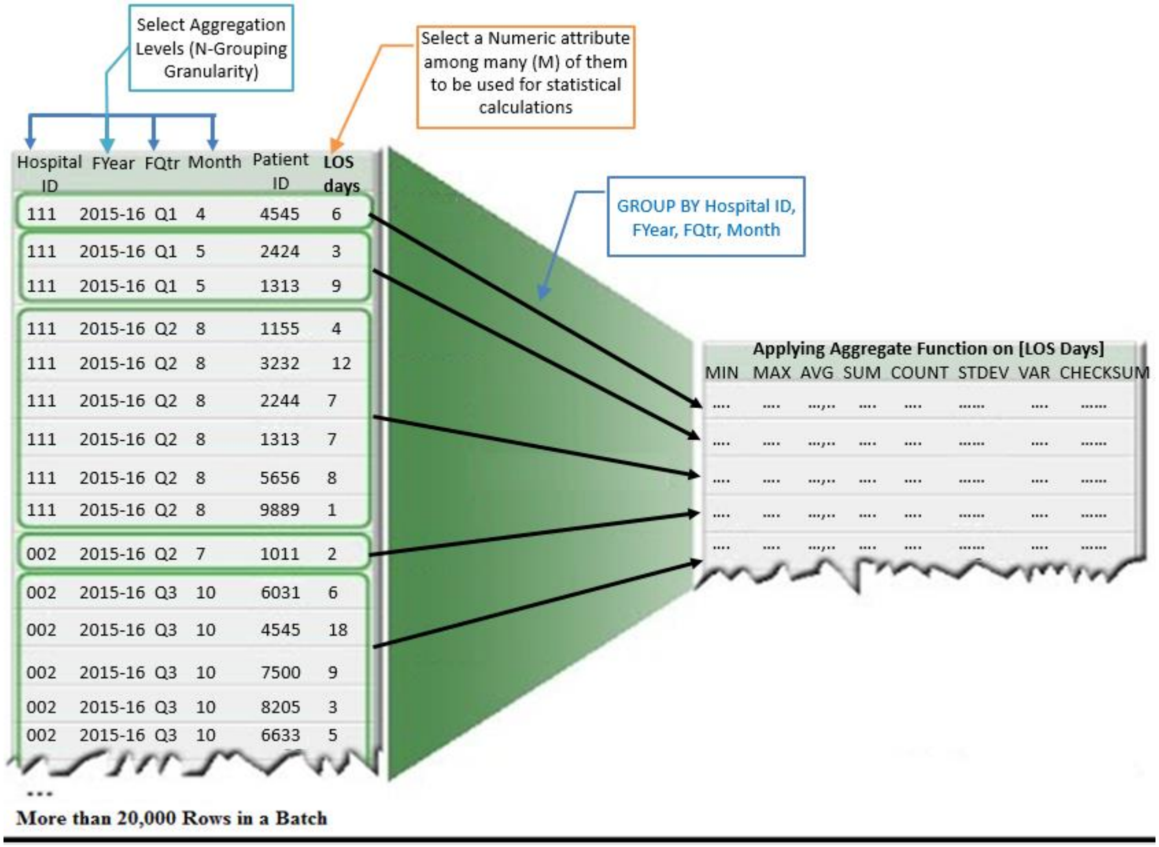
The proposed data masking algorithm is derived from the statistical content of the extracted dataset (T is a data table made up of set of columns $C_1$,$C_2$,…$C_N$ and rows $R_1$,$R_2$,…,$R_N$). Data is grouped at certain levels (micro-aggregation) based on the selection of many indirect identifiers as well as the selection of a specific numeric attribute within the dataset to be used for the statistical calculations as shown in Fig. 4.4.

The combination of the associated statistical variables (see Table 4.1) are put together in a sequence and encapsulated to construct the Masking Signature Key ($K_M$, binary data type), to be linked and saved with each row in a new column into a new destination dataset.

**Table 4.1: Statistical functions' data types and sizes**

| *Variable* | *Data Type* | *Size (Byte)* |
|---|---|---|
| *COUNT()* | Integer | 4 (32 bit) |
| *MIN()* | Small int / integer | 2 (16 bit) or 4 (32 bit) |
| *MAX()* | Small int / integer | 2 (16 bit) or 4 (32 bit) |
| *SUM()* | Integer | 4 (32 bit) |

| | | |
|---|---|---|
| *AVG()* | Real | 4 (32 bit) |
| *STDEV()* | Real | 4 (32 bit) |
| *VAR()* | Real | 4 (32 bit) |
| *CHECKSUM()* | Integer | 4 (32 bit) |



Select Aggregation Levels (N-Grouping Granularity)

Select a Numeric attribute among many (M) of them to be used for statistical calculations

GROUP BY Hospital ID, FYear, FQtr, Month

| Hospital ID | FYear | FQtr | Month | Patient ID | LOS days |
|---|---|---|---|---|---|
| 111 | 2015-16 | Q1 | 4 | 4545 | 6 |
| 111 | 2015-16 | Q1 | 5 | 2424 | 3 |
| 111 | 2015-16 | Q1 | 5 | 1313 | 9 |
| 111 | 2015-16 | Q2 | 8 | 1155 | 4 |
| 111 | 2015-16 | Q2 | 8 | 3232 | 12 |
| 111 | 2015-16 | Q2 | 8 | 2244 | 7 |
| 111 | 2015-16 | Q2 | 8 | 1313 | 7 |
| 111 | 2015-16 | Q2 | 8 | 5656 | 8 |
| 111 | 2015-16 | Q2 | 8 | 9889 | 1 |
| 002 | 2015-16 | Q2 | 7 | 1011 | 2 |
| 002 | 2015-16 | Q3 | 10 | 6031 | 6 |
| 002 | 2015-16 | Q3 | 10 | 4545 | 18 |
| 002 | 2015-16 | Q3 | 10 | 7500 | 9 |
| 002 | 2015-16 | Q3 | 10 | 8205 | 3 |
| 002 | 2015-16 | Q3 | 10 | 6633 | 5 |

...

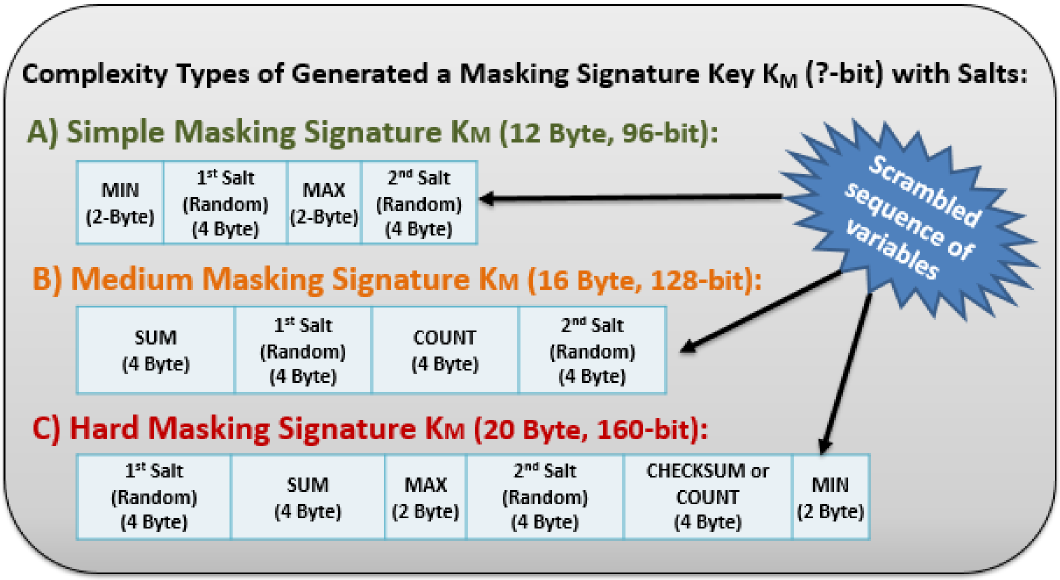
More than 20,000 Rows in a Batch

Applying Aggregate Function on [LOS Days]

| MIN | MAX | AVG | SUM | COUNT | STDEV | VAR | CHECKSUM |
|---|---|---|---|---|---|---|---|
| .... | .... | ...,.. | .... | .... | ...... | .... | ...... |
| .... | .... | ...,.. | .... | .... | ...... | .... | ...... |
| .... | .... | ...,.. | .... | .... | ...... | .... | ...... |
| .... | .... | ...,.. | .... | .... | ...... | .... | ...... |
| .... | .... | ...,.. | .... | .... | ...... | .... | ...... |

**Figure 4.4: COBAD algorithm is determining 8 statistical variables for each group level**

In addition, we can randomize the generated masking signature $K_M$ by appending or prepending two random numbers (2 × integer = 8 Byte) for each row, called salts $K_{S1}$ and $K_{S2}$. This will increase the protection of the new masked value. These pre-defined 2 × 32-bit salts will be saved within the generated masking key and are only accessible through the querying process, seen in Fig. 4.5.



Complexity Types of Generated a Masking Signature Key $K_M$ (?-bit) with Salts:

A) Simple Masking Signature $K_M$ (12 Byte, 96-bit):

| MIN (2-Byte) | 1st Salt (Random) (4 Byte) | MAX (2-Byte) | 2nd Salt (Random) (4 Byte) |
|---|---|---|---|

B) Medium Masking Signature $K_M$ (16 Byte, 128-bit):

| SUM (4 Byte) | 1st Salt (Random) (4 Byte) | COUNT (4 Byte) | 2nd Salt (Random) (4 Byte) |
|---|---|---|---|

C) Hard Masking Signature $K_M$ (20 Byte, 160-bit):

| 1st Salt (Random) (4 Byte) | SUM (4 Byte) | MAX (2 Byte) | 2nd Salt (Random) (4 Byte) | CHECKSUM or COUNT (4 Byte) | MIN (2 Byte) |
|---|---|---|---|---|---|

Scrambled sequence of variables

**Figure 4.5: Generating Masking Key Signature from derived statistical variables**

Moreover, the derived statistical variables (COUNT, MIN, MAX, SUM, AVG, STDDEV, VAR, CHECKSUM, etc.) in association with the two salts numbers ($K_{S1}$ & $K_{S2}$) will be used within our proposed mathematical masking formulas (1a, 1b, 2, and 3) and will be organized in such a way to generate the new masked value at three different complexity levels. This can be generated by using consecutive MOD (%), XOR (^), and other mathematical operators (used notations from SQL server script), as follows:

- **Simple Formula:** using 6 combinations of variables forming 12 bytes (96-bit) or 16 bytes (128-bit) size of Masking Signature Key ($K_M$):

  **For numeric data:**

  $(Ri,Cj)'=(Ri,Cj)+(((K_{S1}\wedge MAX)\ \%\ K_{S2})-MIN)*10^{-7}$ ……… (1a)

  Or we can use the Min-Max normalization formula which is a linear mapping of the original data into a new range with new lower (newMin) and upper (newMax) limits matching the range of the attribute under consideration as follows: [22][35]

  $$(R_i, C_j)' = \frac{(R_i, C_j) - MIN}{MAX - MIN} * (\,newMax - newMin) + newMax \quad \dots\dots(1b)$$

  **For alphanumeric data**, i.e., postal code, we use :

  *'A…Z':* $\quad$ *ASCII('A') - 65 + $K_{S2}$) % 26 + 65* …… (1c)

*‘0…9’:* $ASCII(‘5’) - 48 + K_{S1}) \% 10 + 48$ ……. (1d)

- **Medium Formula:** using 16 bytes (128-bit) size of variables

  **For numeric data:**

  $(Ri,Cj)'=(Ri,Cj)+((K_{S1}$^$SUM)\%CHECK+K_{S2}*10^{-7}) - VAR$ ………(2)

  **For alphanumeric data**, will add another variable to 1c & 1d equations

- **Hard Formula:** using 20 bytes (160-bit) size of variables

  **For numeric data:**

  $(R_i, C_j)' = (R_i, C_j) - \left(\left((K_{S1} \% SUM) \% VAR \wedge MIN\right) \% K_{S2}\right) + MIN$ … (3)

  **For alphanumeric data**, will add 2 more variables to 1c & 1d equations

The scrambled sequence of the statistical variables in this signature key will be used to re-identify the original value at the query stage of analysis services. The detailed processes of the “Apply”, “Sign” and “Keep testing” modules have been organized in a diagram as shown in Fig. 4.6 as well as described in the following algorithms in Fig. 4.7 and Fig. 4.8.

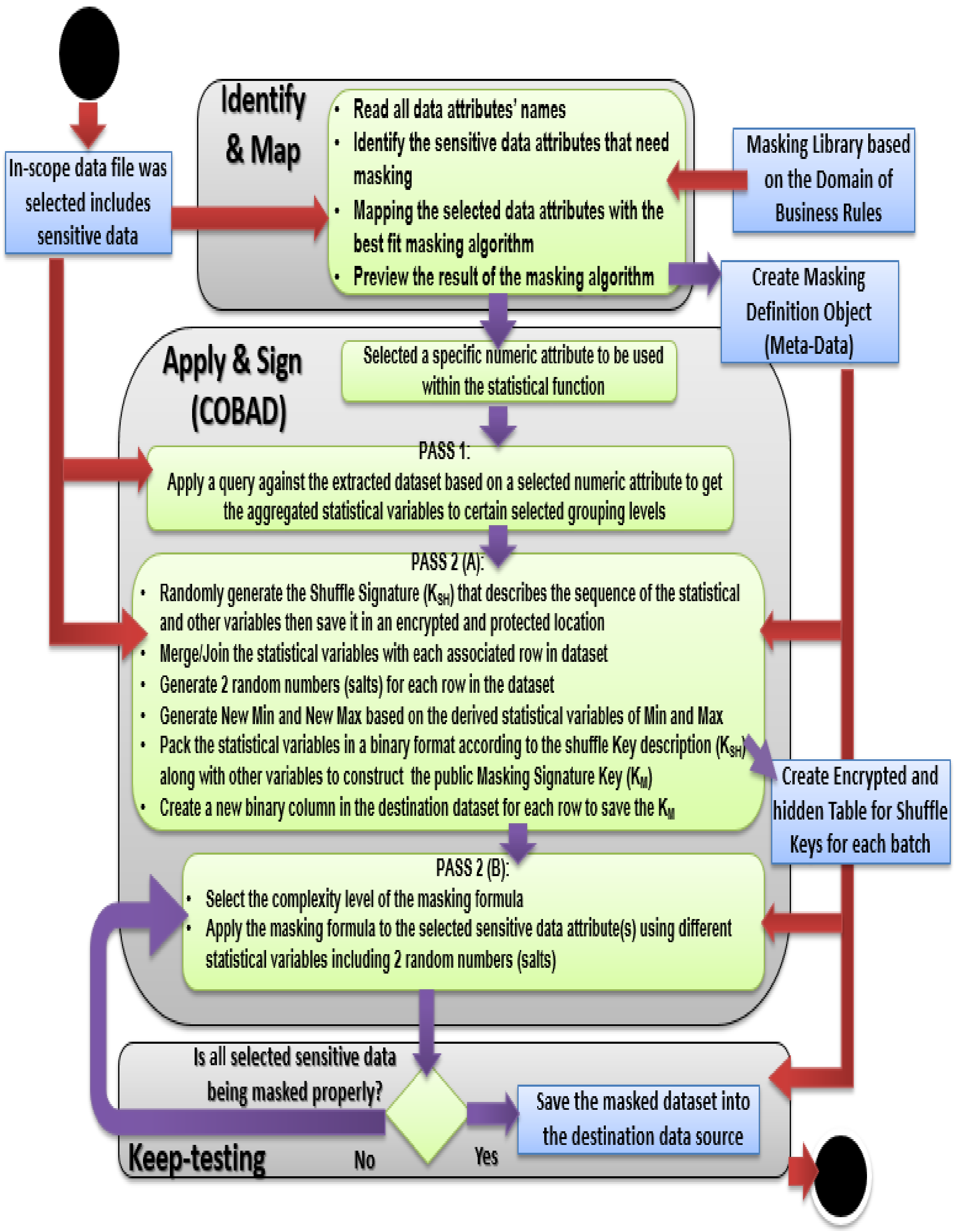
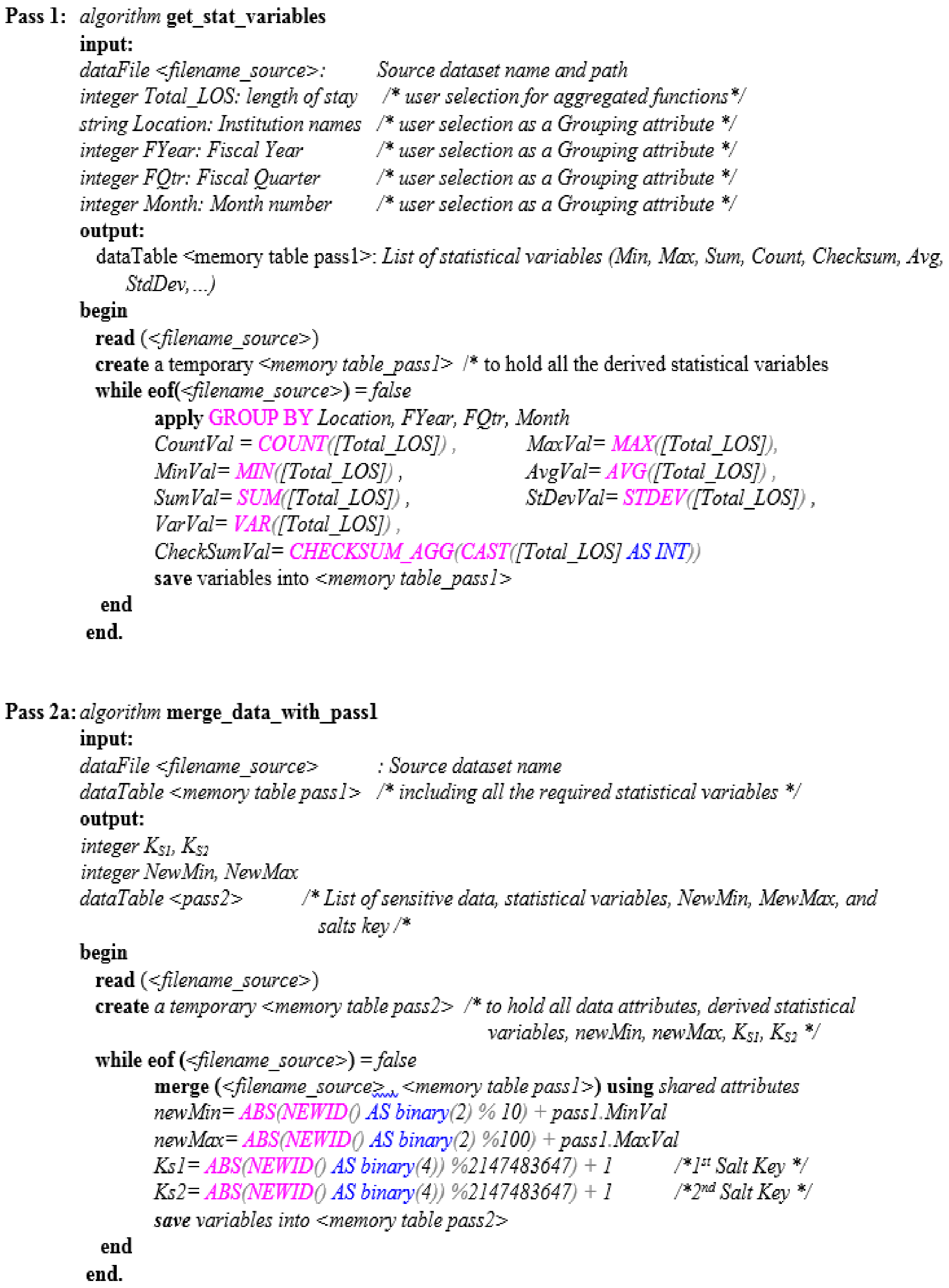

In-scope data file was selected includes sensitive data
Identify & Map
Read all data attributes' names
Identify the sensitive data attributes that need masking
Mapping the selected data attributes with the best fit masking algorithm
Preview the result of the masking algorithm
Masking Library based on the Domain of Business Rules
Create Masking Definition Object (Meta-Data)
Apply & Sign (COBAD)
Selected a specific numeric attribute to be used within the statistical function
PASS 1:
Apply a query against the extracted dataset based on a selected numeric attribute to get the aggregated statistical variables to certain selected grouping levels
PASS 2 (A):
Randomly generate the Shuffle Signature ($K_{SH}$) that describes the sequence of the statistical and other variables then save it in an encrypted and protected location
Merge/Join the statistical variables with each associated row in dataset
Generate 2 random numbers (salts) for each row in the dataset
Generate New Min and New Max based on the derived statistical variables of Min and Max
Pack the statistical variables in a binary format according to the shuffle Key description ($K_{SH}$) along with other variables to construct the public Masking Signature Key ($K_M$)
Create a new binary column in the destination dataset for each row to save the $K_M$
Create Encrypted and hidden Table for Shuffle Keys for each batch
PASS 2 (B):
Select the complexity level of the masking formula
Apply the masking formula to the selected sensitive data attribute(s) using different statistical variables including 2 random numbers (salts)
Is all selected sensitive data being masked properly?
Save the masked dataset into the destination data source
Keep-testing
No
Yes

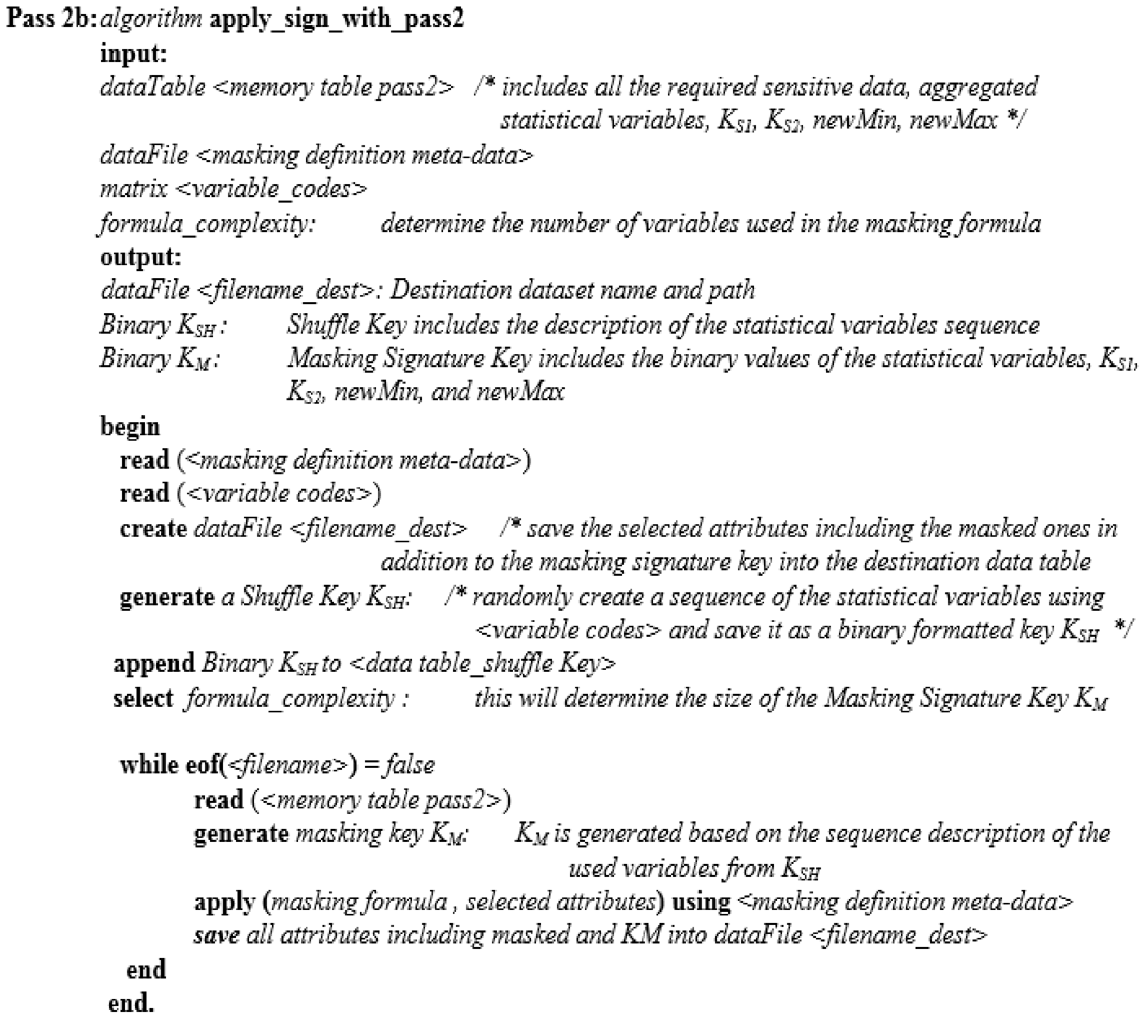
**Figure 4.6: UML Activity Diagram for the "Apply", "Sign" and "Keep" components in the Data Masking Framework for BI platform**



```
Pass 1:  algorithm get_stat_variables
         input:
         dataFile <filename_source>:         Source dataset name and path
         integer Total_LOS: length of stay    /* user selection for aggregated functions*/
         string Location: Institution names  /* user selection as a Grouping attribute */
         integer FYear: Fiscal Year          /* user selection as a Grouping attribute */
         integer FQtr: Fiscal Quarter        /* user selection as a Grouping attribute */
         integer Month: Month number         /* user selection as a Grouping attribute */
         output:
           dataTable <memory table pass1>: List of statistical variables (Min, Max, Sum, Count, Checksum, Avg,
              StdDev, ...)
         begin
           read (<filename_source>)
           create a temporary <memory table_pass1>  /* to hold all the derived statistical variables
           while eof(<filename_source>) = false
                   apply GROUP BY Location, FYear, FQtr, Month
                   CountVal = COUNT([Total_LOS]) ,          MaxVal= MAX([Total_LOS]),
                   MinVal= MIN([Total_LOS]) ,               AvgVal= AVG([Total_LOS]) ,
                   SumVal= SUM([Total_LOS]) ,               StDevVal= STDEV([Total_LOS]) ,
                   VarVal= VAR([Total_LOS]) ,
                   CheckSumVal= CHECKSUM_AGG(CAST([Total_LOS] AS INT))
                   save variables into <memory table_pass1>
            end
          end.


Pass 2a: algorithm merge_data_with_pass1
         input:
         dataFile <filename_source>           : Source dataset name
         dataTable <memory table pass1>   /* including all the required statistical variables */
         output:
         integer $K_{S1}$, $K_{S2}$
         integer NewMin, NewMax
         dataTable <pass2>           /* List of sensitive data, statistical variables, NewMin, MewMax, and
                                       salts key /*
         begin
           read (<filename_source>)
           create a temporary <memory table pass2>  /* to hold all data attributes, derived statistical
                                                       variables, newMin, newMax, $K_{S1}$, $K_{S2}$ */
           while eof (<filename_source>) = false
                   merge (<filename_source>, <memory table pass1>) using shared attributes
                   newMin= ABS(NEWID() AS binary(2) % 10) + pass1.MinVal
                   newMax= ABS(NEWID() AS binary(2) %100) + pass1.MaxVal
                   Ks1= ABS(NEWID() AS binary(4)) %2147483647) + 1          /*1st Salt Key */
                   Ks2= ABS(NEWID() AS binary(4)) %2147483647) + 1          /*2nd Salt Key */
                   save variables into <memory table pass2>
            end
          end.
```

**Figure 4.7: Pass1) Get Statistical Variables' algorithm. Pass 2a) Merge extracted data with the statistical variables' algorithm**



```
Pass 2b: algorithm apply_sign_with_pass2
    input:
    dataTable <memory table pass2>   /* includes all the required sensitive data, aggregated
                                        statistical variables, $K_{S1}$, $K_{S2}$, newMin, newMax */
    dataFile <masking definition meta-data>
    matrix <variable_codes>
    formula_complexity:      determine the number of variables used in the masking formula
    output:
    dataFile <filename_dest>: Destination dataset name and path
    Binary $K_{SH}$:      Shuffle Key includes the description of the statistical variables sequence
    Binary $K_M$:       Masking Signature Key includes the binary values of the statistical variables, $K_{S1}$,
                    $K_{S2}$, newMin, and newMax
    begin
      read (<masking definition meta-data>)
      read (<variable codes>)
      create dataFile <filename_dest>    /* save the selected attributes including the masked ones in
                                  addition to the masking signature key into the destination data table
      generate a Shuffle Key $K_{SH}$:    /* randomly create a sequence of the statistical variables using
                                  <variable codes> and save it as a binary formatted key $K_{SH}$ */
     append Binary $K_{SH}$ to <data table_shuffle Key>
     select formula_complexity :       this will determine the size of the Masking Signature Key $K_M$

      while eof(<filename>) = false
            read (<memory table pass2>)
            generate masking key $K_M$:      $K_M$ is generated based on the sequence description of the
                                         used variables from $K_{SH}$
            apply (masking formula , selected attributes) using <masking definition meta-data>
            save all attributes including masked and KM into dataFile <filename_dest>
      end
     end.
```

**Figure 4.8: Pass 2b) Apply, Sign, and Keep with Pass2 algorithm**

In theory, the masking technique is based on the following terminology
Lets A is the original data set, A = $\{a_1,a_2,\ldots,a_n\}$
And B is the masked data set, B = $\{b_1,b_2,\ldots,b_n\}$
Also, statistical variable values are: $V_S = \{V_{S1},V_{S2},\ldots.,V_{Sn}\}$
And, statistical variable codes are: $V_C = \{V_{C1},V_{C2},\ldots,V_{Cn}\}$

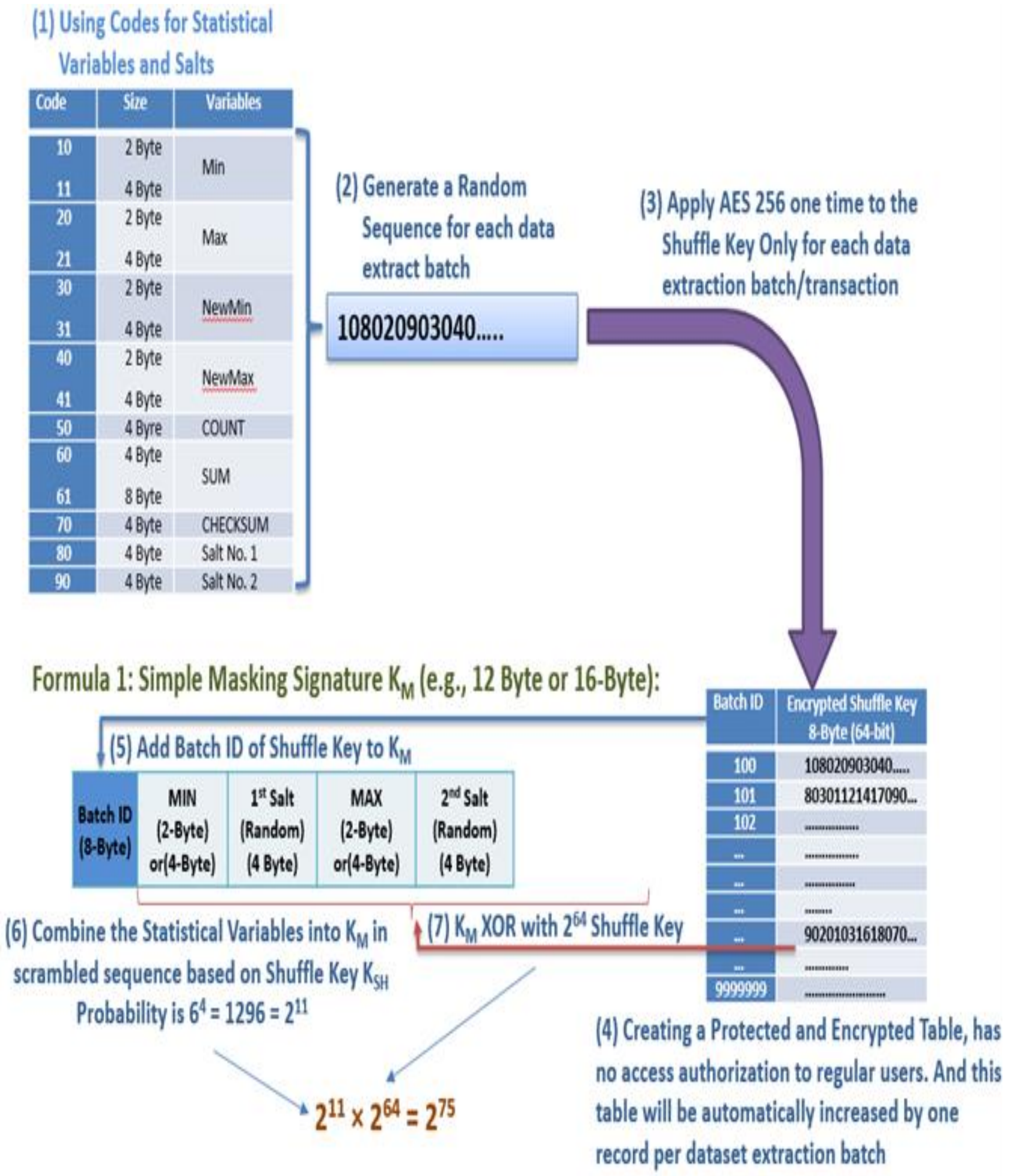
The general masking idea is:
**B = A +** $\boldsymbol{f(K_M, K_{SH})}$ , *$K_M$ is the Masking Key Signature and $K_{SH}$ is the Shuffle Key*

$$K_M = \bigcup_{i=1}^{n} V_{Si} \qquad\qquad K_{SH} = \bigcup_{i=1}^{n} V_{Ci}$$

Note: the symbol U is used to denote to the concatenation operation

**Hence:** $\mathbf{B = A +} \boldsymbol{f}( \bigcup_{i=1}^{n} V_{Ci} \rightarrow K_M = \bigcup_{i=1}^{n} V_{Si})$

To dig deeper into Pass 2 (A) and (B) of the COBAD algorithm, the following Fig. 4.9 visually illustrates the detailed steps of the automated process for generating the Shuffle Key ($K_{SH}$) and the Masking Signature Key ($K_M$).



(1) Using Codes for Statistical Variables and Salts
(2) Generate a Random Sequence for each data extract batch
108020903040.....
(3) Apply AES 256 one time to the Shuffle Key Only for each data extraction batch/transaction
Formula 1: Simple Masking Signature $K_M$ (e.g., 12 Byte or 16-Byte):
(5) Add Batch ID of Shuffle Key to $K_M$
(6) Combine the Statistical Variables into $K_M$ in scrambled sequence based on Shuffle Key $K_{SH}$ Probability is $6^4 = 1296 = 2^{11}$
(7) $K_M$ XOR with $2^{64}$ Shuffle Key
$2^{11} \times 2^{64} = 2^{75}$
(4) Creating a Protected and Encrypted Table, has no access authorization to regular users. And this table will be automatically increased by one record per dataset extraction batch

| Code | Size | Variables |
|---|---|---|
| 10 | 2 Byte | Min |
| 11 | 4 Byte | |
| 20 | 2 Byte | Max |
| 21 | 4 Byte | |
| 30 | 2 Byte | NewMin |
| 31 | 4 Byte | |
| 40 | 2 Byte | NewMax |
| 41 | 4 Byte | |
| 50 | 4 Byre | COUNT |
| 60 | 4 Byte | SUM |
| 61 | 8 Byte | |
| 70 | 4 Byte | CHECKSUM |
| 80 | 4 Byte | Salt No. 1 |
| 90 | 4 Byte | Salt No. 2 |

| Batch ID (8-Byte) | MIN (2-Byte) or(4-Byte) | 1st Salt (Random) (4 Byte) | MAX (2-Byte) or(4-Byte) | 2nd Salt (Random) (4 Byte) |
|---|---|---|---|---|

| Batch ID | Encrypted Shuffle Key 8-Byte (64-bit) |
|---|---|
| 100 | 108020903040..... |
| 101 | 80301121417090... |
| 102 | ............... |
| ... | ............... |
| ... | .............. |
| ... | ........ |
| ... | 90201031618070... |
| ... | ............ |
| 9999999 | ...................... |

**Figure 4.9: Automation Process of generating the Masking Signature Key $K_M$ (n-bit) and associated Shuffle Key $K_{SH}$ (64-bit)**

## 4.3 Discussion

A second COBAD module (Apply and Sign) of the built-in data masking framework within the BI platform is proposed in this chapter to add extra protection for the data at rest (within the Data warehouse and staging area) to prevent internal risks of re-identification. The proposed practical built-in data masking framework (iMaskU), focused on the implementation and testing modules for the new efficient and strong masking technique (COBAD), will dramatically reduce the cost associated with privacy disclosure (re-identification risk), whether from internal or external data breaches.

The new COBAD data-masking algorithm (reversible de-identification) is based on the derived statistical variables of the extracted data content, and how to shuffle the sequence of these variables and pack them in a binary Shuffle Key $K_{SH}$. Furthermore, the value of the statistical variables has been packed in a binary Masked Signature Key $K_M$. The $K_M$ publicized and saved with each row into the destination data table.

The proposed COBAD algorithm is practically applicable to different data types such as; numeric (age, length of stay), date stamps (birth date, discharge date), character (sex), and alphanumeric (postal code, diagnosis code).

This technique is used to efficiently de-identify the sensitive data, preserve the data format, and maintain the data utility. It is highly recommended to use the medium complexity of masking formula that generates a 128-bit masking signature key or higher, which will ensure the impossibility of re-identifying the sensitive data and lower the risk factor to an acceptable minimum rate.

It is also worth mentioning here the theoretical comparison of the main features between our COBAD masking technique and other vendors / stand-alone solutions as shown in Table 4.2 [75]. This table indicates the similarity and differences of the data masking functionality when it comes to fulfil the BI platform requirements, such as maintain the statistical data utility, preserve the data format, identify the sensitive data fields (direct and quasi-identifiers), and apply reversible masking algorithm to protect data at rest.

**Table 4.2: Comparison of different features between our COBAD masking technique and other vendors and stand-alone solutions**

| Features | COBAD<br>(Numeric and Alphanumeric) | IBM<br>(InfoSphere Optim) | ORACLE<br>(Data Masking) | SQL Server<br>(2016 and up) | MOBAT<br>(Numeric) |
|---|---|---|---|---|---|
| Static data masking (SDM) | ☑ | ☑ | ☑ | -- | ☑ |
| Reversible Masking to maintain the statitacally data utility | ☑ | -- | -- | -- | ☑ |
| Maintain the contextual data utility | ☑ | ☑ | ☑ | ☑ | ☑ |
| Heterogeneous data source support (Numeric, Date, alphanumeric) | ☑ | ☑ | ☑ | ☑ | -- |
| Built-in Framework in BI or DBMS platforms | ☑ | ☑ | ☑ | ☑ | -- |
| Maintain the data consistency and the referential integrity DW wide | -- | ☑ | ☑ | -- | -- |
| Discover sensitive data dataset wide | ☑ | ☑ | ☑ | -- | -- |
| Best match masking algorithm and flexibility to change | ☑ | ☑ | ☑ | -- | -- |
| Auto-generation of Integration Service ETL package | ☑ | -- | -- | -- | -- |
| Preserve data format | ☑ | ☑ | ☑ | ☑ | ☑ |
| Dynamic data masking (DDM) | -- | ☑ | ☑ | ☑ | -- |
| Key geneation and management | ☑ | -- | -- | -- | ☑ |
| Comply with the Privacy policies and regulations | ☑ | ☑ | ☑ | ☑ | ☑ |
| Manage the risk of sensitive data re-identification | ☑ | ☑ | ☑ | ☑ | ☑ |

The COBAD reversible masking technique is use the persistent or static data masking strategy, that creates a new copy of dataset to which the data masking rules have been applied and save it into the DW at the beginning of process in the context of focusing data-centric security at rest.

The beauty of this strategy is the sensitive data permanently obfuscated due to application of the data masking techniques during the extract, transform, and load process. Moreover, no need to any process when data is retrieved during the usage phase (if the accuracy of the result is not matter). Also, turn to secured data method as a means of complying with the required privacy regulations. This resulted in safe to share your data with internal and external stakeholders.

Whereas, the drawbacks of adopting the static masking strategy is, the masking process during ETL process may take minutes to complete depending on the size of extracted dataset. In addition, it cannot be easily used to back-up the production datasets as it needs to apply de-masking algorithm to the entire masked dataset and this may take time.

# Chapter 5

# 5 Re-identification Process and Security Validation

The sharing of data or discovery of patterns and associations in large data sets (healthcare, finance, industry) through applying statistical and data mining algorithms is an integral facet of corporate needs. In this situation, front-end users, such as, researchers, analysts, and consultants need access to huge quantities of record-level data, for this purpose, many organizations have established data warehouse to hold the utmost needed sensitive data [56][59].

Given the utility of integrated DW for end-user, it is important to re-identify the masked sensitive data in a manner that ensure the real data will be available during the analysis process. In this chapter, the re-identification formulas are proposed and the final results have been compared with the outcomes of the analysis of the original data to measure the error rate. Furthermore, the achievement of the security requirements of the COBAD masking algorithm by using simple formula is verified. In addition, the performance of the algorithm is assessed.

## 5.1 Re-identification Process (Utilize Component)

The re-identification process is the last component in our iMaskU framework, which is called “Utilize”. The utilize component is a reverse engineering process that adds a built in functionality in which acts as a de-masking tool to enable the authorized end-users to retrieve data and apply the required analysis process to get the right results to be utilized by analytical tools as shown in Figure 5.1.

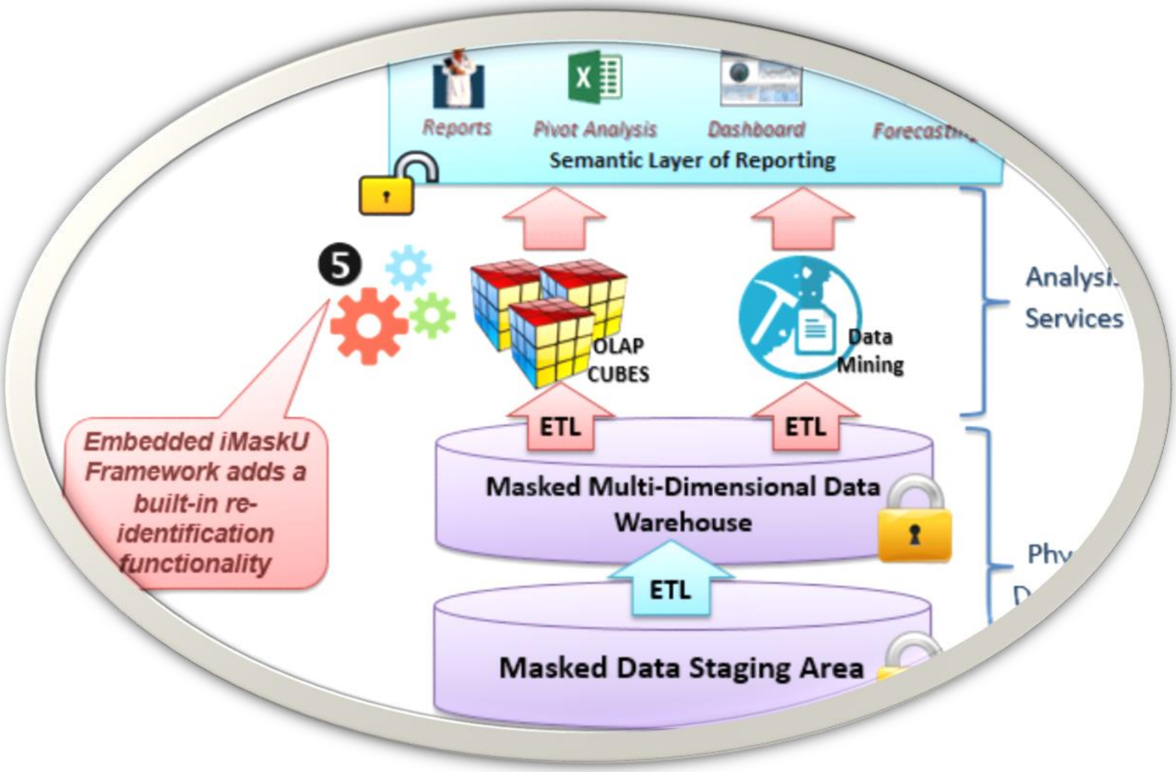

Reports
Pivot Analysis
Dashboard
Forecastin
Semantic Layer of Reporting
5
Embedded iMaskU Framework adds a built-in re-identification functionality
OLAP CUBES
Data Mining
Analysi
Services
ETL
ETL
Masked Multi-Dimensional Data Warehouse
ETL
Phy
D
Masked Data Staging Area

**Figure 5.1: Re-identification (Utilize) Component in iMaskU Framework**

The algorithm of the de-masking process relies on reading the public masking signature key ($K_M$) for each record and then interprets its contains by using the associated shuffle key ($K_{SH}$) to get the right order and size of the statistical variable that had created from the extracted data content. In addition, from the size of $K_M$, the re-identification function determines the complexity of the masking formula in which used to de-identify the sensitive data attributes. The following steps and Figure 5.2 illustrates the algorithm and the UML design for the re-identification process within the utilize component:

1. Read the masked data table/ file
2. Identify the attributes that needs to be re-identifies in a query level
3. Read the $K_M$, separate the Batch ID from other part
4. If Batch ID exists in the temporary memory table, use the decoded statistical variables, otherwise fetch the $K_{SH}$ from the DBMS

5. Decode the $K_{SH}$ to get the sequence and the size of the saved statistical variables ($V_{S1}$…$V_{Sn}$)
6. Read the rest part of the $K_M$ in order to get the values of each statistical variable $V_S$
7. Save the $V_{S1}$…$V_{Sn}$ into a temporary memory table to reduce the fetch time to the shuffle keys' in a physical private table and eliminate also the decoding time
8. From the business rules library, use the right de-masking (re-identification) formula

   a) For a character attribute (e.g Sex):

```
CHAR(CAST((ASCII(Sex_Masked)-65-cast(substring( [MaskedKeySignature],1,4)
as int))%26 +65+26 AS INTEGER))
```

   b) For an integer attribute (eg. Age, or Length of Stay):

```
Case
When CAST((((([Age_Masked]-newMin)*1./(newMax-newMin)*1.)*(MaxVal -
MinVal)+MinVal)/.8 AS INTEGER) >0 Then CAST((((([Age_Masked]-newMin)*1./(newMax-
newMin)*1.)*(MaxVal -MinVal)+MinVal)/.8 AS INTEGER) +1
When CAST((((([Age_Masked]-newMin)*1./(newMax-newMin)*1.)*(MaxVal -
MinVal)+MinVal)/.8 AS INTEGER) <0 Then 0
Else CAST((((([Age_Masked]-newMin)*1./(newMax-newMin)*1.)*(MaxVal -
MinVal)+MinVal)/.8 AS INTEGER) end

OR

[Total_LOS_Masked]-(((Ks1^MaxVal)%Ks2)-MinVal)/100000000
```

9. Apply the masking formula to the specific masked attributes taking into the consideration the rules of the classification module.
10. Repeat the above steps to the end of data file/table
11. Display the result of the query

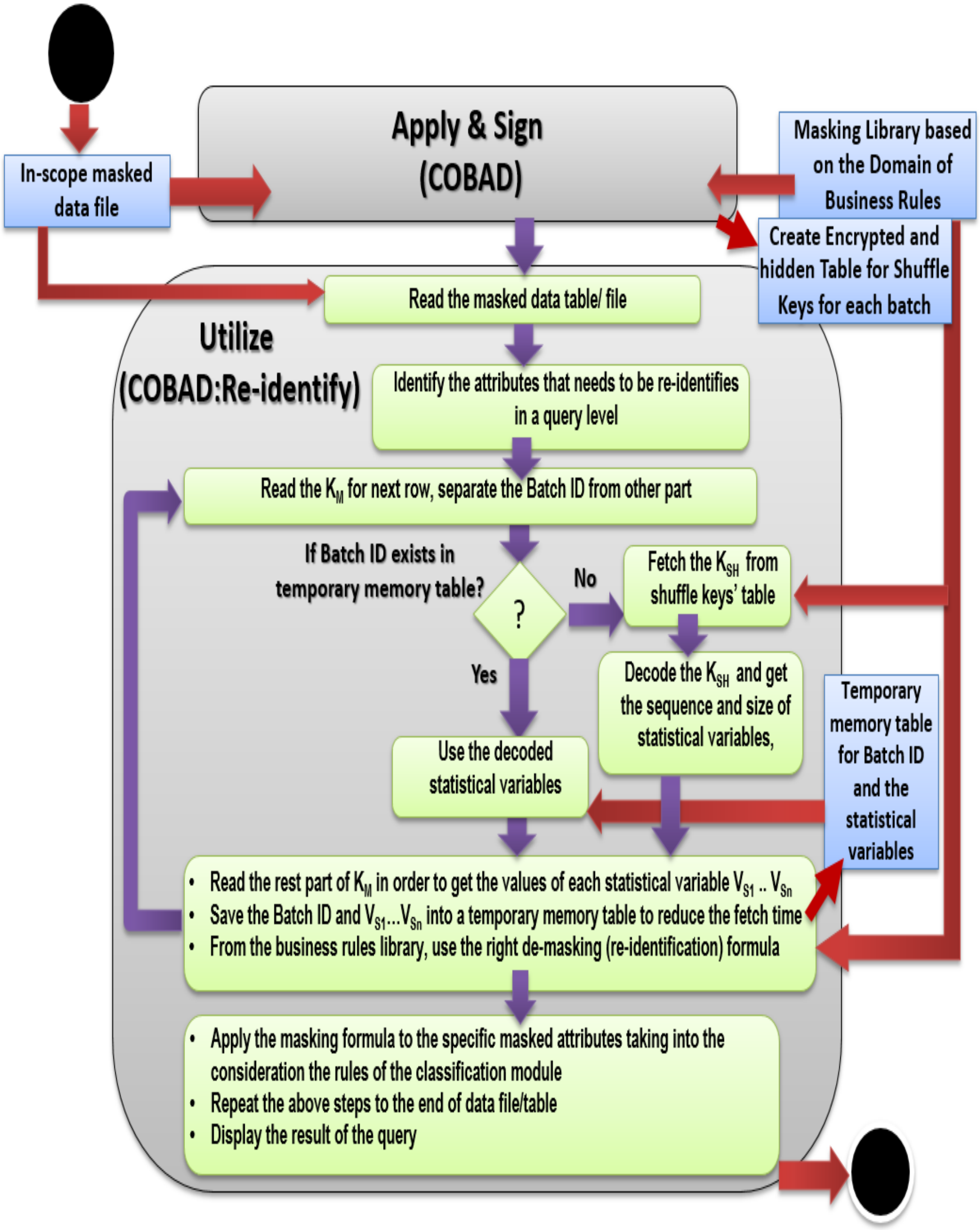

In-scope masked data file
Apply & Sign (COBAD)
Masking Library based on the Domain of Business Rules
Create Encrypted and hidden Table for Shuffle Keys for each batch
Read the masked data table/ file
Utilize (COBAD:Re-identify)
Identify the attributes that needs to be re-identifies in a query level
Read the $K_M$ for next row, separate the Batch ID from other part
If Batch ID exists in temporary memory table?
?
No
Yes
Fetch the $K_{SH}$ from shuffle keys' table
Decode the $K_{SH}$ and get the sequence and size of statistical variables,
Use the decoded statistical variables
Temporary memory table for Batch ID and the statistical variables
Read the rest part of $K_M$ in order to get the values of each statistical variable $V_{S1}$ .. $V_{Sn}$
Save the Batch ID and $V_{S1}$...$V_{Sn}$ into a temporary memory table to reduce the fetch time
From the business rules library, use the right de-masking (re-identification) formula
Apply the masking formula to the specific masked attributes taking into the consideration the rules of the classification module
Repeat the above steps to the end of data file/table
Display the result of the query

**Figure 5.2: UML Activity Diagram of Re-identification "Utilize" component in the iMaskU Framework for BI platform**

## 5.2 Re-identification Accuracy Measure

The accuracy is the quality of the data of being correct or the degree to which the result of an analysis conforms to the correct values. To calculate the accuracy of the re-identification process, we are going to calculate the ratio of an error of de-masked value versus the original one. From the sample masked health data, let us assume that we have two integer attributes, eg. Length of Stay (LOS) and Age. The steps of the accuracy calculation as follows:

1. Consider a data base D consist of two tuples T (tables): D={$T_1$, $T_2$}.
2. Each tuple T consist of set of attributes: $T_1$={$A_1$, $A_2$,…, $A_{11}$} , Original Table
   $T_2$={A’$_1$, A’$_2$, …,A’$_{11}$} , Masked Table
3. Identify the sensitive numeric attributes, in our case study {$A_1$:Age, $A_2$:LOS}
   This attribute has been masked using four statistical variable and two salt keys (12 Byte: Min, Max, newMin, newMax, $K_{S1}$, $K_{S2}$) in a simple formula as follows:

```
AGE:    CAST((((([Age]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax    -
newMin) + newMin)*.8 AS INTEGER) Age_Masked
```

```
LOS: CAST([Total_LOS]+(((Ks1^MaxVal)%Ks2)-MinVal)*0.00000001
```

4. Apply the following de-masking (re-identification) formulas to retrieve the original values:

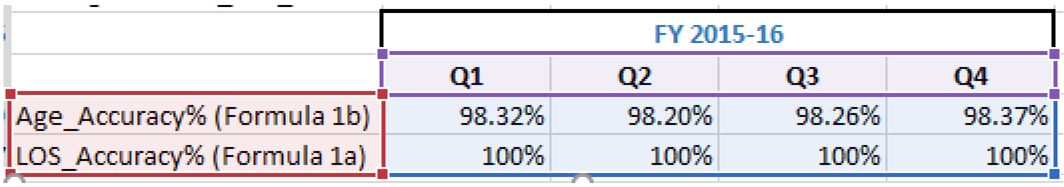
```
AGE: case when ks1>ks2 then [Dschg_Age] + ((Ks1 % MaxVal)
%Ks2)*.1 - MinVal else [Dschg_Age] + ((Ks2 % MaxVal) %Ks1) -
MinVal end
```

```
LOS: [Total_LOS_Mask]-(((Ks1^ MaxVal)% Ks2)- MinVal)/100000000
```

5. Compare the re-identified value with the original to determine the accuracy ratio
   %Accuracy = de-mask / original × 100%

   See Table 5.1 and Figure 5.3.

**Table 5.1: Accuracy of Re-identification Ratio**



| | FY 2015-16 | | | |
|---|---|---|---|---|
| | Q1 | Q2 | Q3 | Q4 |
| Age_Accuracy% (Formula 1b) | 98.32% | 98.20% | 98.26% | 98.37% |
| LOS_Accuracy% (Formula 1a) | 100% | 100% | 100% | 100% |

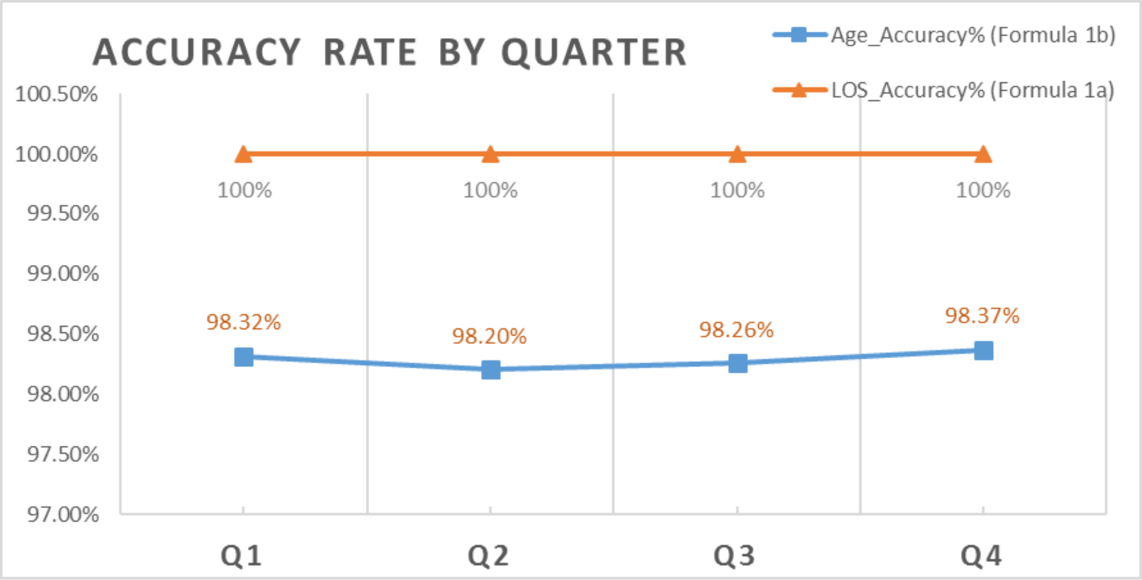


**Figure 5.3: Accuracy of Re-identification Ratio**

The accuracy results shows, by applying our COBAD's simple masking formula (Min, Max, $K_{S1}$, $K_{S2}$) on LOS will get 100% accuracy result. On the other hand, if we apply Min-Max normalization formula (Min, Max, newMin, newMax) to the Age attribute, the re-identification accuracy reaches to ~98% (reversible here is something like lossy compression and decompression and due to quantization), which is acceptable range from statistical analysis perspective [76].

## 5.3 Security Validation of COBAD (Re-identification Risk Analysis)

Re-identification risk [31][38] is the measure of the risk that the sensitive information about individuals in the dataset can be retrieved from the de-identified data. It is difficult to quantify this risk, as the ability to re-identify depends on many factors such as, general knowledge of the original dataset, the de-identification algorithm, the attacker's available resources and skills, and the availability of additional open data that can be linked with the masked data. In many cases, the risk of re-identification will be increased over time as techniques getting improved and more contextual information becomes available (e.g., publicly or through a purchase). [39][42][43]

Our method uses many derived statistical variables along with salts numbers to generate a masking key signature for each row to be used during the de-identification process and is freely distributable with no privacy issues. This section demonstrates that the proposed technique is secured to protect the sensitive data (for instance the numeric attributes). A hacker might use several possible ways that could crack the masking signature and re-identify data as follows:

### 5.3.1 A Brute Force Attack

The combination length of the key signature used in the masking process determines the feasibility of performing a brute-force attack, with using longer key becomes exponentially more difficult to crack than shorter ones. Brute-force attack involves systematically checking all possible key combinations until the correct one is found.

Here is an example of a brute force attack on a 4-bit key:

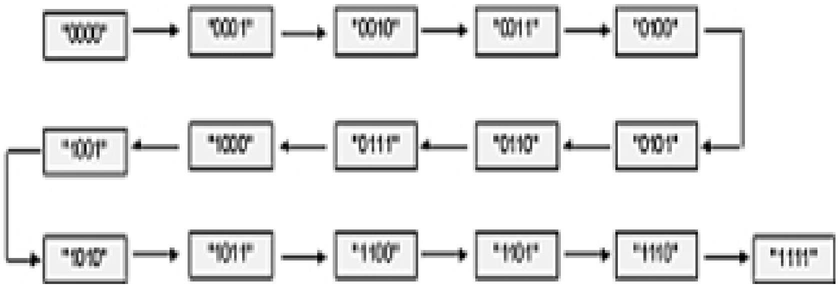


**Figure 5.4: Brute Force attack on 4-bit key**

As shown in Fig. 5.4, it will take a maximum 16 rounds to check every possible key combination starting with "0000" and ending with “1111”. Given sufficient time, a brute force attack is capable of cracking any known algorithm. Table 5.2 shows the possible numbers of key combinations with respect to key size:

**Table 5.2: Key combinations versus Key size**

| Key Size | | Possible combinations |
|---|---|---|
| **8-bit** | $2^8$ | 256 |
| **16-bit** | $2^{16}$ | 65536 |
| **32-bit** | $2^{32}$ | $4.2 \times 10^9$ |

| | | |
|---|---|---|
| **64-bit** | $2^{64}$ | 1.8 x $10^{19}$ |
| **72-bit** | $2^{72}$ | 4.7 x $10^{21}$ |
| **75-bit** | $2^{75}$ | 3.7x$10^{22}$ (4 variables of 2-byte/4-byte size each and using simple masking formula in addition to $6^4$ combination of variables position within the formula and XOR with 64-bit Shuffle Key) |
| **96-bit** | $2^{96}$ | 7.9 x $10^{28}$ |
| **128-bit** | $2^{128}$ | 3.4 x $10^{38}$ |
| **256-bit** | $2^{256}$ | 1.1 x $10^{77}$ |
| **512-bit** | $2^{512}$ | 1.3 x $10^{154}$ |

Notice the exponential increase in possible combinations as the masking key and shuffle key sizes increase. Using the simple masking formula which contains 4 variables of 2-byte/4-bytes size each (96-bit) along with $6^4$ ($1296=2^{11}$) probabilities of variables' positions within the pre-defined formula (assuming that the attacker knows the arithmetic operators), in addition to using an encrypted 64-bit shuffle key $K_{SH}$ to be XOR with the Masking Key $K_M$, this will increase the number of checking rounds up to $3.7 \times 10^{22}$

In total the masking key size is 64-bit × $2^{11}$ variable positions = $2^{75} = 3.7 \times 10^{22}$ of maximum possibilities to crack the key and find the variables' sequence by using the brute force attack. We will prove that this method is secure to be used to protect the privacy of sensitive data within the DW. Let us consider the following: [42]

The fastest supercomputer's speed [40] (as per Wikipedia) is approximately 10.51 Pentaflops = $10.51\times10^{15}$ Flops [Flops = Floating point operations per second]

No. of Flops required per combination check: 1000 (very optimistic assumption)

No. of combination checks per second = $(10.51 \times 10^{15}) / 1000 = 10.51 \times 10^{12}$

No. of seconds in one Year = $365\times24\times60\times60 = 31536000$

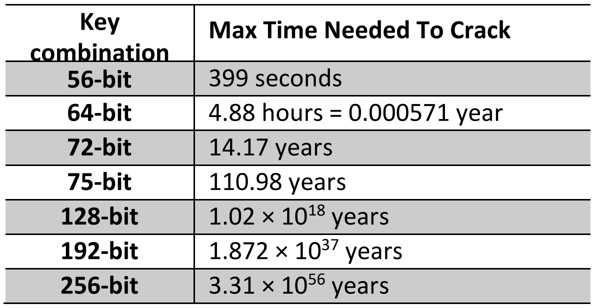
No. of Years to crack iMaskU COBAD with 75-bit Combination = $(3.7 \times 10^{22}) / ((10.51 \times 10^{12}) \times 31536000)$

$= (0.35 \times 10^{10}) / 31536000$

= 110.98 Years needed to crack a key combination for one record

As shown in Table 5.3, even with using a supercomputer, it will take a maximum of 110.98 years + (0.000571 × No. of Grouping levels for the dataset) ≈ 111 years to crack the 75-bit generated masking combination key (for simple masking formula) using a brute force attack. On average, you can crack the key after testing 50% of the possibilities ≈ 55 years, so, it is not feasible to use an intensive computation by internal or external attackers to retrieve the original data.

**Table 5.3: Max Time needed to crack Masking Key versus Key combination**



| Key combination | Max Time Needed To Crack |
|---|---|
| **56-bit** | 399 seconds |
| **64-bit** | 4.88 hours = 0.000571 year |
| **72-bit** | 14.17 years |
| **75-bit** | 110.98 years |
| **128-bit** | 1.02 × $10^{18}$ years |
| **192-bit** | 1.872 × $10^{37}$ years |
| **256-bit** | 3.31 × $10^{56}$ years |

### 5.3.2 A Dictionary Attack

This type of attack attempts to guess the key combinations of a de-identified attribute by using many different common and possible values that are likely to be used by a human or even acceptable value for the selected numeric attribute that is being used by aggregation functions. For example, let us assume that the Maximum and Minimum Value for the selected attribute "Patient Length of Stay" will not exceed 256 ($2^8$), which means, there is no need to try 65535 ($2^{16}$) or 2,147 483,647 ($2^{32}$) possibilities to guess the number. In this case we reduced the attack cycles to $2^{48}$×$2^8$= $2^{56}$ attempts. By using the same brute-force calculation, we will get an approximate crack time from Table 5.2:

Time to crack 56-bit Key = 399 seconds for each record

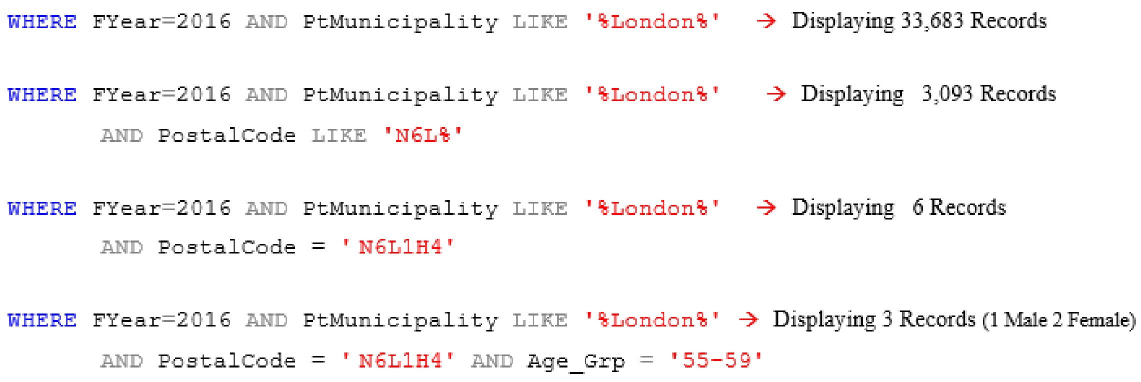
If we masked 50,000 records in one batch of data extraction along with using different salt numbers for each record, then an attacker would need: 399 × 50000 /(60 × 60 × 24) = 231 days to re-identify the sensitive numeric attributes for the entire dataset.

However, in our masking algorithm, we used two salt numbers ($K_{S1}$ & $K_{S2}$) as random data as an extra addition to the masking signature key ($K_M$). So in this case, we have to decide whether to increase the size of the salt number to 32-bit, or use the 2nd complexity degree of the masking formula with more statistical variables. Therefore, the primary function of using salt numbers is to defend against dictionary attacks.

### 5.3.3 A Data Linkage Attack

A linkage attack involves linking each record in the masked dataset to the similar records in another open dataset (that contains the identity of the data subject) by using some common attributes available to link (e.g., age, sex, postal codes), including the degree to which they individually and collectively are able to identify an individual uniquely. (in this case study we use the original dataset). [41]

Let us assume an internal hypothetical data attacker (or curious employee) is trying to retrieve protected health information (PHI) from DW for a specific individual who is living in his/her neighborhood. The attacker is in possession of some additional background information about the patient ("Prosecutor Scenario"), such as Year of discharge = 2016, Municipality = London, Postal Code = N6L1H4, Age_Group=55-59, and Sex. The attempt results as follows:



```
WHERE FYear=2016 AND PtMunicipality LIKE '%London%'   → Displaying 33,683 Records

WHERE FYear=2016 AND PtMunicipality LIKE '%London%'    → Displaying   3,093 Records
      AND PostalCode LIKE 'N6L%'

WHERE FYear=2016 AND PtMunicipality LIKE '%London%'   → Displaying   6 Records
      AND PostalCode = ' N6L1H4'

WHERE FYear=2016 AND PtMunicipality LIKE '%London%'  → Displaying 3 Records (1 Male 2 Female)
      AND PostalCode = ' N6L1H4' AND Age_Grp = '55-59'
```

Meanwhile, if the attacker has another open dataset, such as electoral list, Ontario Registered Person (ORP), or even yellow pages, by linking them, he/she will retrieve the patient's name, phone number, and address and will be able to retrieve their sensitive health information, such as, their Health Card Number, diagnosis codes, intervention codes, Length of Stay, Chronic Disease, discharge status, etc.

We are going to measure the re-identification risk by using “Marketer Scenario” which is another way to test and calculate the percentage of identities in the masked dataset that can be correctly re-identified. In this case we applied our COBAD masking technique to four main attributes, Age, Sex, Postal Codes, and Total Length of Stay. Table 5.4 illustrates the results of the comparison/matching process of the re-identified attributes individually and collectively against the original dataset.

**Table 5.4: The re-identification risk rate for data linkage attack using three main common attributes**

| Linked Attribute(s)<br>(Masked with Original) | (#) Total Number of Matching | (%) Percentage of Matching |
|---|---|---|
| Sex | 1,990 | 3.88% |
| Age | 2,868 | 5.59% |
| Age Group (5 years) | 8,170 | 15.92% |
| Postal Code (6 characters) | 211 | 0.41% |
| Postal Code + Age Group | 32 | 0.06% |
| Sex + Age | 103 | 0.2% |
| Sex + Age Group | 321 | 0.63% |
| Sex + Age + Posta Code | 0 | 0.0% |
| Sex + Age Group + Postal Code | 2 | 0.004% |
| FSA (First 3 char of Postal Code) + Age Group + Sex | 321 | 0.63% |
| FSA + Sex | 1,987 | 3.87% |

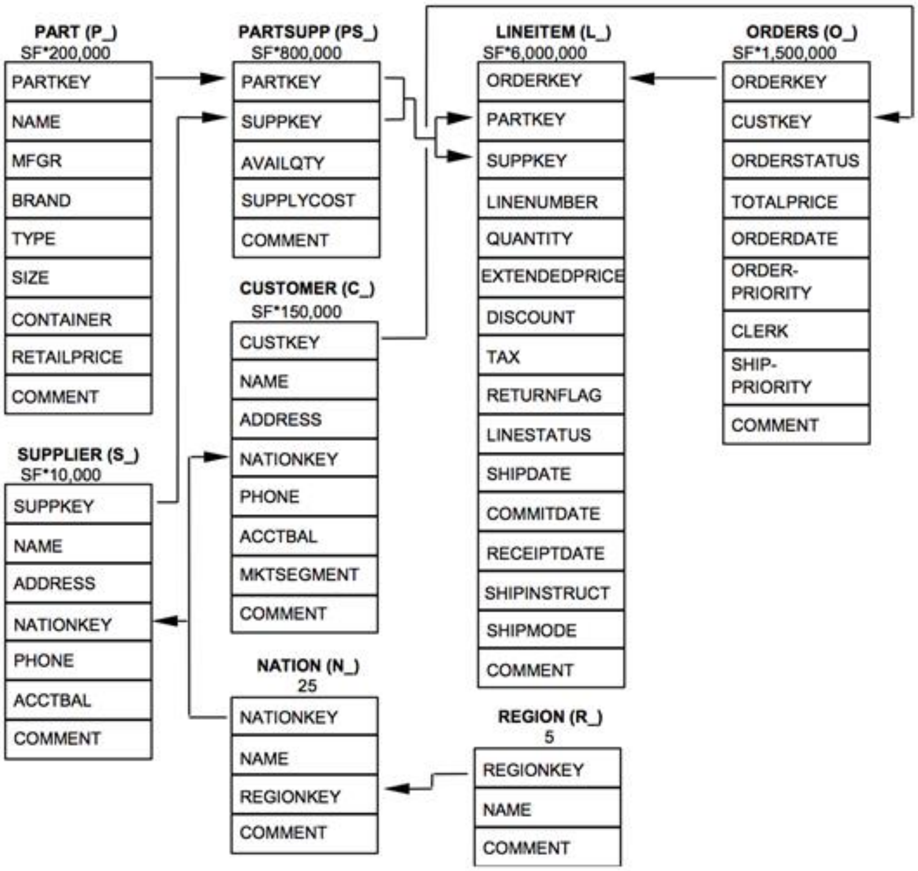
Note: Total Number of Masked Records for the selected dataset = 51,308

## 5.4 COBAD Performance Validation

Most of database management systems (DBMS) provide standard Transparent Data Encryption technology (often abbreviated to TDE) to be used to protect data at rest only. TDE is a technology employed by Microsoft, IBM and Oracle to encrypt database files at file, table and column levels [74]. The best practices guidelines by vendors state that the encryption is the best way for protection purpose. However, since integrated DW started to be the integral part of BI analytics platform, which includes billions of records in their fact tables along with running many ad-hoc queries to access large amounts of numerical and textual data. Subsequently, the data encryption and decryption (AES and 3DES) overhead is considered a major concern from the DW perspective, such as, overhead query response time, extra storage space, and large processing time [71].

In this section, we explain and analyze the performance comparison among COBAD (simple, medium, complex) technique, MOBAT Masking technique, AES128, and 3DES encryption algorithms using the well known TPC-H benchmark. TPC-H is a decision support benchmark in which consist of datasets and ad-hoc queries that examine large volume of data [72]. TPC-H benchmark consist of eight individual tables linked in entity relational (ER) diagram as shown in Figure 5.5 below.

**Figure 2: The TPC-H Schema**



**Legend:**

- The parentheses following each table name contain the prefix of the column names for that table;
- The arrows point in the direction of the one-to-many relationships between tables;
- The number/formula below each table name represents the cardinality (number of rows) of the table. Some are factored by SF, the Scale Factor, to obtain the chosen database size. The cardinality for the LINEITEM table is approximate

**Figure 5.5: TPC-H Schema [73]**

We are going to test and measure the performance impact on a workload of some benchmarks queries that access the fact table "LineItem" with its linked dimension tables. The response time for TPC-H queries shown in Table 5.5.

**Table 5.5: Standard query response time vs. AES128 & 3DES using TPC-H Benchmark (in seconds)**

| Query No. | Standard Query Exec. Time (sec) | AES128 Query Exec. Time (sec) | 3DES168 Query Exec. Time (sec) |
|---|---|---|---|
| **Q1** | 15 | 94 | 100 |
| **Q8** | 308 | 369 | 373 |
| **Q9** | 128 | 190 | 195 |
| **Q10** | 12 | 23 | 23 |
| **Q17** | 36 | 49 | 51 |
| **Q19** | 92 | 122 | 129 |
| **Q20** | 102 | 180 | 184 |

Although, SQL Server claims that the using of encryption and decryption algorithms will increase 5%-10%, however, the above results show that average response time overhead is higher than 50%.

In our experiment to evaluate the performance of our COBAD data masking technique, we used 1GB scale sizes data schema of the TPC-H decision support benchmark along with three scenarios: a) simple masking formula; b) medium masking formula; and c) hard masking formula versus the AES128 and 3DES encryption algorithms. The number of rows and the storage size for each used table in TPC-H schema is shown in Table 5.6.

**Table 5.6: Table sizes of TPC-H decision support benchmark.**

| Table Name | Type | Table Size (MB) | No. of Rows |
|---|---|---|---|
| LineItem | Fact | 740 | 6,001,2015 |
| Orders | Dimension | 152 | 1,500,000 |
| Customers | Dimension | 32 | 150,000 |
| Suppliers | Dimension | 4 | 10,000 |
| Part | Dimension | 32 | 200,000 |
| PartSupp | Dimension | 112 | 800,000 |
| Nation | Dimension | <1 | 25 |
| Region | Dimension | <1 | 5 |
| TOTALS | | ~1 GB | 8,661,245 |

To implement the performance evaluation test using the TPC-H schema, we selected three numerical data fields (L_Quantity, L_ExtendedPrice, L_Discount) within

“LineItem” fact (original size is 772 MB and loading time is 310 sec), as well as running seven queries (Q1, Q8, Q9, Q10, Q17, Q19, Q20). Table 5.7 and Figure 5.6 present the results of the data storage size and loading time for each scenario.

**Table 5.7: TPC-H 1GB LineItem fact’s storage size and data loading.**

| | Origina dataset | Using COBAD | | | Using SQL Server Encryption | | Using MOBAT |
|---|---|---|---|---|---|---|---|
| | | Simple Formula | Medium Formula | Hard Formula | AES128 | 3DES168 | Masking (AddCol) |
| Storage Space (MB) | 772 | 825 | 831 | 836 | 1,458 | 1,205 | 816 |
| %Space Overhead | | 6.87% | 7.64% | 8.29% | 88.86% | 56.09% | 5.70% |
| Loading Time (sec) | 310 | 347 | 353 | 359 | 623 | 681 | 334 |
| %Load Time Overhead | | 11.94% | 13.87% | 15.81% | 100.97% | 119.68% | 7.74% |

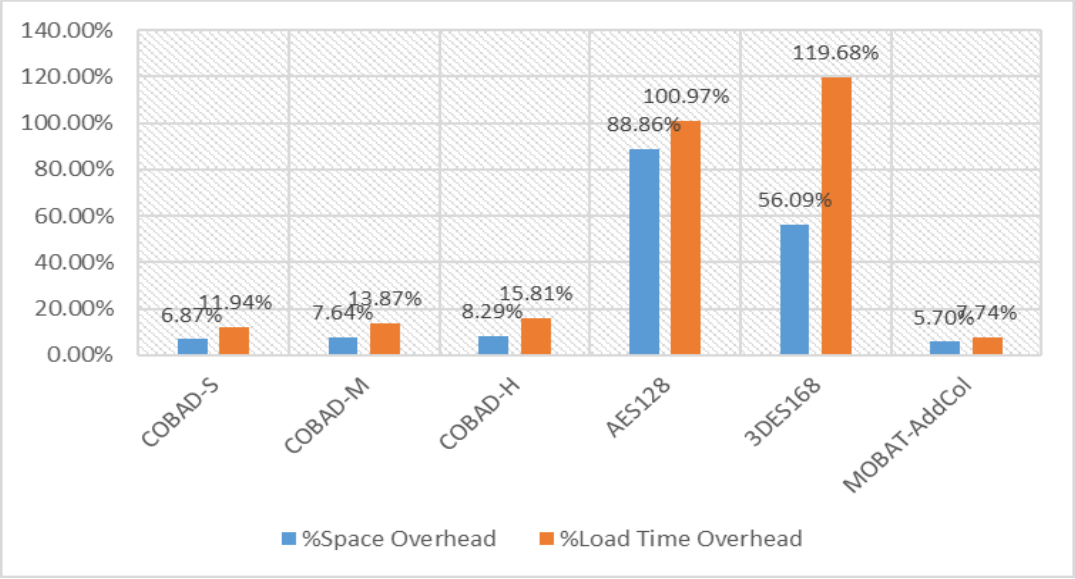


**Figure 5.6: TPC-H 1GB LineItem fact’s storage size and data loading**

From the results above, both storage space and loading time overheads for TDE encryption algorithms are much greater than the COBAD masking technique’s formulas.

The performance evaluation for the execution time of the selected TPC-H queries as shown in Figure 5.7. It is obvious that the our COBAD results are better than AES128 & 3DES168 encryption, and closer to MOBAT Masking technique (in this algorithm, the masked value has a significant variance from the actual value).

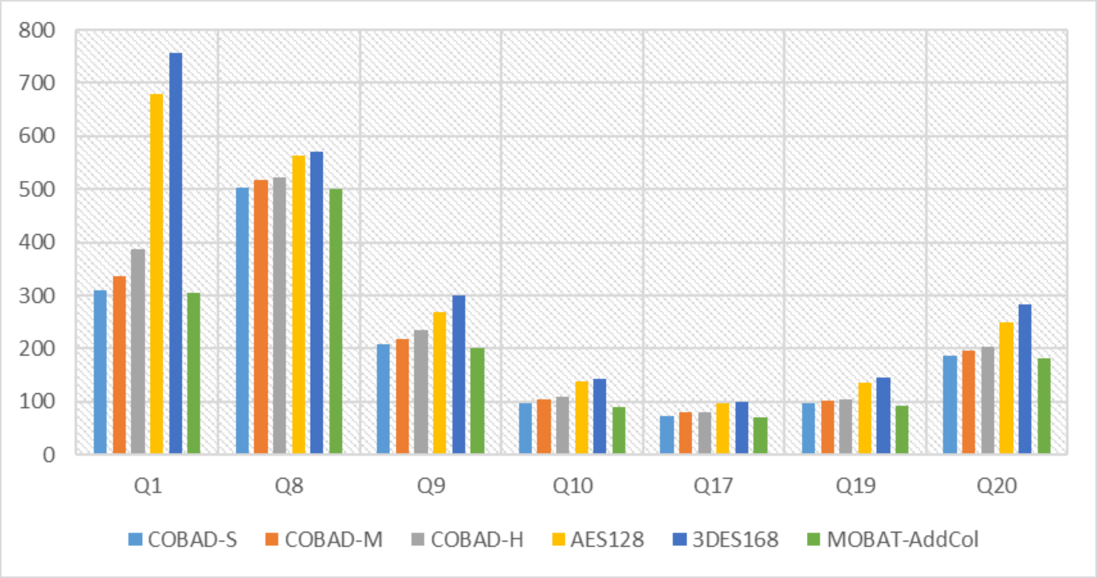


**Figure 5.7: TPC-H 1GB dataset query execution times**

## 5.5 Results

The new COBAD data masking algorithm (reversible de-identification) has been applied to data types such as; numeric (age, length of stay), date stamps (birth date, discharge date), character (sex), and alphanumeric (postal code, diagnosis code). This technique is used to efficiently de-identify the sensitive data, preserve the data format, and maintain the data utility. The strength of the re-identification risk factor for the COBAD technique has been computed using a theoretical super computer speed (using the simple masking formula that uses two statistical variables and two salt keys, along with $2^{11}$ variables' combination making then XORing with the 64-bit shuffle key, the total key combination reached to $2^{75}$).

Three attacker methods are considered, a) using a brute force attack, needs, on average, 55 years to crack the key of each record; b) using the dictionary attack, needs 6.6 minutes to crack the key for a single record, and 231 days for the entire extracted dataset (containing 50,000 records), c) using a data linkage attack, the re-identification risk is very low when the common linked attributes are used (e.g., postal code, age group, and sex). Only two records have been identified out of 51,000 (0.004%). However, with the removal of the last three characters of postal codes and using the approximate match, the re-identification risk jumps up to 0.63% (321 records have been re-identified out of 51,000).

We conducted an empirical assessment to prove the accuracy of the proposed algorithm by applying our masking and de-masking algorithms COBAD to the real world and use our health data to compare between the accuracy of the averages of original Length of Stay (LOS) versus Masked values of LOS as shown in Figure 5.8 (A). This shows the masked numbers are approximately 44.2% shifted positively from the actual values, practically, this means the masked data is acceptable contextually but not statistically. Figure 5.8 (B) depicts that the accuracy rate is 100% when matching de-masked values with the original one.

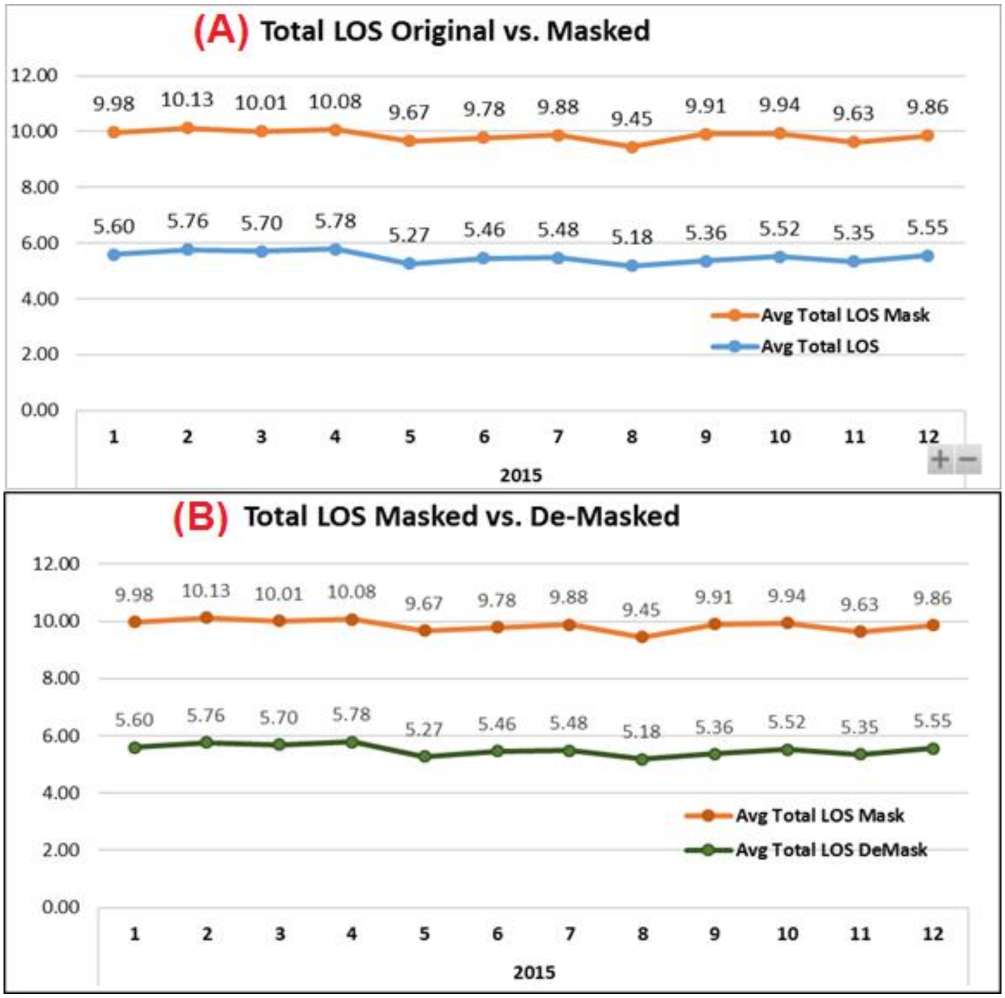


**Figure 5.8: Empirical assessment of COBAD's masking and de-masking algorithms**

# Chapter 6

# 6 Conclusions and Future Work

The implementation of a Business Intelligence (BI) platform in different types of organizations, has become an important project. One common implementation is to extract sensitive data from production databases and then load them into an enterprise Data Warehouse (DW). However, the internal privacy breach that occurs by accessing this DW by developers, researchers, and testers a serious threat that should be taken into consideration.

The use of data masking software in a non-production environment is increasingly common. Integrating data masking framework within DBMS and BI platform is under investigation and currently the industry focus on using traditional techniques to irreversibly mask data in one-way without taking in their consideration the data utility. The use of traditional masking technique in BI platform is not efficient. Therefore, a built-in data masking framework and reversible technique to protect the privacy of the sensitive data as well as maintain the data utility was herein proposed. The iMaskU framework identifies the sensitive data and securely save them into staging area and/or DW. Furthermore, iMaskU-COBAD functions as a strong de-identification technique to protect the data at rest against any risks of disclosure of internal or external attack.

## 6.1 Summary of Contributions

The main goal of this thesis is designing and developing a novel data masking framework and technique to securely save the sensitive data into an integrated DW and to prevent risks of external and internal attacks. The objectives that support our goal are achieved through the following tasks:

1. **Identify the different types of traditional data masking techniques and explore their uselessness in BI analytics platform:**

    - The usage of existing types of traditional data masking techniques were investigated and their taxonomy were identified.

- A case scenario to demonstrate the uselessness of such masking techniques has been studied and the results were examined as an evidence of proof of concept (Chapter 2)

2. **Analysis and design of a data classifier module for masking framework iMaskU**

    - Automate the process of identification the sensitive data and then mapping them with the proper masking technique.

    - After a clear analysis of masking methods for sensitive data and the privacy requirement, a set of business rules established based on intensive investigation (Chapter 3)

    - A conference paper has been published: O. Ali, A. Ouda, "A Classification Module in Data Masking Framework for Business Intelligence Platform in Healthcare", IEEE 7th Annual Conference (IEMCON), 2016

3. **Design, develop, and validate a new reversible data masking technique (COBAD) that de-identifies the sensitive data and protect it from internal or external attacks.**

    - The second module is the core algorithm of the new masking technique based on the statistical content of the extracted data has been investigated and designed (Chapter 4).

    - Apply the proper proposed algorithm based on the type of the sensitive data using different complexity level of the masking formula in which relies on the number of the statistical variables used
    - Then construct the shuffle key and masking signature keys to be used for re-identification purposes
    - Validate the disclosure risk of COBAD technique by using three attack methods, the results of a validation indicated that COBAD algorithm is secure and hard to get cracked especially when the medium or high complexity level of masking formula were considered. (Chapter 4)

- A conference paper has been published: O. Ali, A. Ouda, “A Content-Based Data Masking Technique for A Built-In Framework in Business Intelligence Platform”, IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017

4. **Design, develop, and validate a re-identification functionality and embed it into iMAskU framework.**

- The algorithm of the de-masking process relies on reading the public masking signature key ($K_M$) for each record and then interprets its contains by using the associated shuffle key ($K_{SH}$) to get the right order and size of the statistical variable that had created from the extracted data content. In addition, from the size of $K_M$, the re-identification function determines the complexity of the masking formula in which used to de-identify the sensitive data attributes.

- Validating the re-identification accuracy through calculating the accuracy of the re-identification process, which is, a ratio of an error of de-masked value versus the original one (Chapter 5).

- Performance validation of the COBAD masking technique has been conducted to compare it against the common encryption algorithms within DBMS. The results proved that COBAD is working more efficient than AES128 and 3DES168 encryption methods.

- A journal paper has been submitted and it is under review: “Preserving Privacy in a Business Intelligence Analytics Platform: an iMaskU Framework with Content-Based Data Masking Technique”. Journal of Information Security and Applications, November 2018

## 6.2 Future Work

This study examines the ability of a framework to detect sensitive data during data extraction process, then how to map it with the appropriate masking algorithm. Apply the selected masking algorithm using the statistical features of the dataset and test the result. The proposed future work will address the following key aspects of this research topic.

- The proposed work in this research considers using health data to build the classifier module to identify the sensitive data within the dataset. However, this is not enough if we need to use it against other financial and industrial datasets. Future work should consider the integration and using of different datasets with existing data masking technique in this research.

- In the design of identification, content-based masking technique, and re-identification modules, the execution time across the consecutive processes to accomplish the required masking functionality is an open research area. Further algorithm optimization method is needed to determine the acceptable execution time when a big size of data used in one extract-load-transform batch.

- The present research does not consider existing implementation and deployment of the final software product. Conducting such a practical approach to commercialize the proposed framework and masking technique remains a challenging software development area that future work should address.

# References

[1] MAIA Intelligence [Online]. Business intelligence – an endless story. pp. 1-14. 2011. Available: http://www.maia-intelligence.com/pdf/BI-An¬endless-story-wp.pdf.

[2] J. Hagerty, R. L. Sallam and J. Richardson. Magic quadrant for business intelligence platforms. Gartner [Online]. 2012. Available: http://www.microstrategy.com/download/files/whitepapers/open/gartner¬magic-quadrant-for-bi-platforms-2012.pdf.

[3] W. H. Inmon. What is a data warehouse? Prism Solutions, Inc [Online]. 1995. Available: www.cait.wustl.edu/cait/papers/prism/vol1_no1/.

[4] Y. Naddaf. Data mining in health informatics. [Online]. Available: http://yavar.naddaf.name/downloads/Data°/"20Mining°/"20in°/"20Health°/"20Informatics.pdf.

[5] Voltage Security Solution for Data De-Identification, White Paper 2014. Available: https://www.voltage.com/wp-content/uploads/Voltage_UC_Data_De-Identification.pdf

[6] A. Sen and A. P. Sinha, "A Comparison of Data Warehousing Methodologies", Communications of the ACM, vol. 48, pp. 79-84, 2005.

[7] R. Guro. Components of business intelligence. The Business Intelligence Guy [Online]. 2011. Available: http://www.the-business-intelligence¬guy.com/components-of-business-intelligence-bi/...

[8] K. El Emam and L. Arbuckle. "Anonymizing Health Data", O'Reilly Media, pp. 5,2013

[9] Ravikumar G K, Dr. B. Justus Rabi, Dr MGR University, “Experimental Study of Various Data Masking Techniques with Random Replacement using data volume”,

(IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 8, August 2011, p.154-158.

[10] Dataguise inc., “Why Add Data Masking To Your Best Practices for Securing Sensitive Data”, www.dataguise.com, 2010.

[11] Ravikumar G. K. ,Manjunath T. N., Ravindra S. Hegadi, Umesh I. M, “A Survey on Recent Trends, Process and Development in Data Masking for Testing”, (IJCSI) International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[12] K. Muralidhar and R. Sarathy. "Data shuffling procedure for masking data", Patent US 7200757 B1, April 2007.

[13] A. H. Ouda and M. R. El-Sakka, "Localization and Security Enhancement of Block-based Image Authentication", IEEE publication, 2005

[14] IRI CoSort Company. " Format-Preserving Encryption (FPE), Encrypt Data Without Changing Its Appearance", http://www.iri.com/solutions/data-masking/encryption/format-preserving-encryption

[15] WekiPedia, " Format-preserving encryption", https://en.wikipedia.org/wiki/Format-preserving_encryption.

[16] M. Li, Z. Liu, C. Jia, Z. Dong, "Data Masking Generic Model", Fourth International Conference on Emerging Intelligent Data and Web Technologies, 2013

[17] Oracle Inc., "Data Masking Best Practices", White Paper, July 2010

[18] R. J. Santos, J. Bernardino, M. Vieira, "A Data Masking Technique for Data Warehouses", ACM Communication, IDEAS11 2011, September 21-23, Lisbon, Portugal

[19] Microsoft Inc., "Dynamic Data Masking", SQL Server 2016, https://msdn.microsoft.com/en-CA/library/mt130841.aspx

[20] R. Dobson, "Masking Personal Identifiable SQL Server Data", November 2013, https://www.mssqltips.com/sqlservertip/3091/masking-personal-identifiable-sql-server-data/

[21] Z. Arthur, "SSIS Data Flow Transformation Component To Provide Basic Data Masking Capabilities", Feb 2012, http://ssisdatamasker.codeplex.com/

[22] O. Ali, A. Ouda, "A Classification Module in Data Masking Framework for Business Intelligence Platform in Healthcare", IEEE 7th Annual Conference (IEMCON), 2016.

[23] Canadian Institute for Health Information (CIHI), “Types of Care”, https://www.cihi.ca/en/types-of-care

[24] Training materials, “IntelliHEALTH – Inpatient Discharge User Guide”, Ontario Ministry of Health and Long – Term Care, Version 1.0, September 2010.

[25] Training materials, “IntelliHEALTH – Ambulatory Visits User Guide”, Ontario Ministry of Health and Long – Term Care, Version 1.0, November 2010.

[26] Training materials, “IntelliHEALTH – Complex Continuing Care User Guide”, Ontario Ministry of Health and Long – Term Care, Version 1.0, September 2011.

[27] Training materials, “IntelliHEALTH – Inpatient Rehabilitation User Guide”, Ontario Ministry of Health and Long – Term Care, Version 1.0, September 2011.

[28] O. Ali, A. B. Nassif, L. F. Capretz, "Business Intelligence Solutions in Healthcare A Case Study: Transforming OLTP system to BI Solution", in 23rd IEEE International Conference on Tools with Artificial Intelligence, Florida, USA, 2011, pp. 393-398.

[29] R. Guro. Components of business intelligence. The Business Intelligence Guy [Online]. 2011. Available: http://www.the-business-intelligenceguy.com/components-of-business-intelligence-bi/ ..

[30] University of Alabama at Birmingham, " Three levels of data classification proposed", Announcement January 2017. Available: https://www.uab.edu/it/home/about-uab-it/announcements/item/819-three-levels-of-data-classification-proposed

[31] S. L. Grafinkle, “De-Identification of Personal Information”, National Institute of Standards and Technology [NISTIR 8053], http://dx.doi.org/10.6028/NIST.IR.8053 , U.S. Department of Commerce, 2015

[32] Text of the regulation, “Personal Health Information Protection Act”, Ontario Government, CHAPTER 3 Schedule A, e-laws Ontario Government, http://www.e-laws.gov.on.ca , 2004.

[33] Canadian Institute for Health Information (CIHI), “Discharge Abstract Database (DAD) Metadata / Data Elements”, https://www.cihi.ca/en/types-of-care/hospital-care/acute-care/dad-metadata

[34] O. Ali, A. Ouda, “A Content-Based Data Masking Technique for A Built-In Framework in Business Intelligence Platform”, IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017

[35] N A Sarada, G Manikandan, Dr.N.Sairam, “A Few New Approaches for Data Masking”, IEEE, International Conference on Circuit, Power and Computing Technologies [ICCPCT], 2015

[36] “De-Identification Guidelines for Structured Data”, Information and Privacy Commissioner of Ontario, June 2016, https://www.ipc.on.ca/wp-content/uploads/2016/08/Deidentification-Guidelines-for-Structured-Data.pdf

[37] R Fraser, D Willison, “Tools for De-Identification of Personal Health Information”, Prepared for the Pan Canadian Health Information Privacy (HIP) Group, September 2009.

[38] Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufman, First Edition, 2010

[39] K. El Emam, F. K Dankar1, A. Neisa, E. Jonker, "Evaluating the risk of patient re-identification from adverse drug event reports", BMC Medical Informatics and Decision Making Journal, 2013, 13:114.

[40] L Xu, C Jiang, "Privacy or Utility in Data Collection? A Contract Theoretic Approach", IEEE Journal of Selected Topics in Signal Processing, Vol. 9, No. 7, October 2015

[41] J Domingo-Ferrer, V Torra, "Disclosure risk assessment in statistical microdata protection via advanced record linkage", Statistics and Computing 13: 343–354, 2003

[42] M Arora, "How secure is AES against brute force attacks?", EE Times, https://www.eetimes.com/document.asp?doc_id=1279619 , July 2012

[43] B Lubarsky, "RE-IDENTIFICATION OF "ANONYMIZED DATA", CITE AS: 1 GEO. L. TECH. REV. 202 (2017), AS: 1 GEO. L. TECH. REV. 202 (2017), https://perma.cc/86RR-JUFT

[44] Office of the Privacy Commissioner of Canada, https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/02_05_d_15/

[45] Personal Health Information Protection Act, 2004, S.O. 2004, c. 3, Sched. A, https://www.ontario.ca/laws/statute/04p03

[46] IMPERVA, Data Masking, https://www.imperva.com/data-security/data-security-101/data-masking/

[47] G C. Deshmukh, S. M. Patil, "Study for Best Data Obfuscation Techniques using Multi-Criteria Decision-Making Technique", International Journal of Computer Applications (0975 – 8887) Volume 180 – No.43, May 2018

[48] Securosis, L.L.C., "Understanding and Selecting Data Masking Solutions: Creating Secure and Useful Data", Version 1.0, August 2012

[49] R Parameswaran, D M Blough, "Privacy Preserving Collaborative Filtering using Data Obfuscation", IEEE International Conference on Granular Computing, 2007

[50] L Mittal, "Data Masking Techniques for Insurance", NIIT Technologies White Paper, www.niit-tech.com, 2016

[51] R Fraser, D Willison, "Tools for De-Identification of Personal Health Information", Pan Canadian Health Information Privacy (HIP) Group, September 2009

[52] D Sanders, D Protti, "Data Warehouses in Healthcare: Fundamental Principles. Electronic Healthcare", 2008, 6: 1-16. Available at: http://www.longwoods.com/product.php?productid=19510&cat=524&page=1

[53] R Numeir, A Lemay, J-M Lina, "Pseudonymization of radiology data for research purposes. Journal of Digital Imaging", 2007; 20(3): 284-295

[54] K El Emam, B Brown, P Abdelmalik, "Evaluating Predictors of Geographic Area Population Size Cut-offs to Manage Re-identification Risk", J Am Med Inform Assoc. 2009;16:256–266. DOI 10.1197/jamia.M2902.

[55] Canadian Institutes of Health Research Privacy Advisory Committee. CIHR Best Practices for Protecting Privacy in Health Research - September 2008. Available at: http://www.cihrirsc. gc.ca/e/documents/pbp_sept2005_e.pdf. 2005. Ottawa, Public Works and Government Services Canada.

[56] B Lubarsky, "Re-Identification Of Anonymized Data", Cite as: 1 GEO. L. TECH. REV. 202 (2017), https://perma.cc/86RR-JUFT.

[57] Microsoft SQL Server 2017, "Dynamic Data Masking", https://docs.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-2017

[58] WayFare Innovative Nearshoring, "Is your Progress database GDPR compliant?", 2016, https://wayfare.ro/progress-database-gdpr-compliant/

[59] S. Pomroy, " Static Versus Dynamic Data Masking", Imperva I.c Blog, https://www.imperva.com/blog/static-versus-dynamic-data-masking/

[60] "Data Snapshot of Health and Healthcare Utilization in Alberta", Data Statistics and Reporting. https://www.albertahealthservices.ca/about/Page13342.aspx

[61] "Measuring System Performance-Indicator Library", Health Quality Ontario, https://www.hqontario.ca/System-Performance/Measuring-System-Performance/

[62] "Report on Performance Scorecard and Big Dots", South West Local Health Integration Network, http://www.southwestlhin.on.ca/accountability/Performance.aspx

[63] "Release Notes", IDS BI Platform (internal documnetation), Hamilton Health Science, HNHB LHIN, July 2016

[64] A project charter, "Data Management System for Infection Prevention and Control", IPAC Project Team, London Health Sciences Centre, 2013

[65] A project specification, "BI Solution- Technical Design", Systemgroup Consulting Inc, 2014

[66] D Friedland, "Which Data Masking Function Should I Use?", IRI Total Data Management, https://www.iri.com/blog/data-protection/data-masking-function-use/

[67] "Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule", Health Information Privacy. https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html#assess

[68] J. Haldeman, "Compare IBM data masking solutions: InfoSphere Optim and DataStage", Nov 2012. https://www.ibm.com/developerworks/data/library/techarticle/dm-1211maskingsolution/index.html

[69] "IBM InfoSphere Optim Data Masking solution", https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IMS14076USEN.

[70] "Masking Sensitive Data", Oracle Database Real Application Testing User's Guide,

[71] R. Jorge Santos, J. Bernardino, M. Vieira, "Using Data Masking for Balancing Security and Performance in Data Warehousing", University of Coimbra, Portugal, Handbook of Research on Computational Intelligence for Engineering, Science, and Business. 2013

[72] "Sample Data: TPC-H Benchmark", Snowflake Documentation, 2012. https://docs.snowflake.net/manuals/user-guide/sample-data-tpch.html#business-question

[73] "TPC BENCHMARK$^{TM}$ H, Decision Support Standard Specification", Transaction Processing Performance Council (TPC), 2014. http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-h_v2.17.1.pdf

[74] "Transparent Data Encryption", WikiPedia, https://en.wikipedia.org/wiki/Transparent_Data_Encryption

[75] "Comparisions of Data Masking products", FireCompass, https://www.firecompass.com/security/comparisons/data-masking-dm/ibm-infosphere-optim-data-privacy-vs-mentis-imask-vs-oracle-data-masking-and-subsetting

[76] " How to Determine a Statistically Valid Sample Size", Qluctch, June 2017, https://qlutch.com/marketing-tips/determine-statistically-valid-sample-size/

[77] C. Rodgers, "Data Classification: Why it is Important for Information Security", Infosec Island, April 2012, http://www.infosecisland.com/blogview/20881-Data-Classification-Why-it-is-Important-for-Information-Security.html

[78] "HIPAA Security Rule", HIPAA Survival Guide, http://www.hipaasurvivalguide.com/hipaa-security-rule.php

[79] "Protecting Telephone-based Payment Card Data", Security Standards Council, March 2011, https://www.pcisecuritystandards.org/documents/protecting_telephone-based_payment_card_data.pdf

[80] "HIPAA vs PIPEDA, Mandatory Protection", MediPENSE

# Appendices

## Appendix A: The results of the simulation using GreenCloud simulator tool

The figure below is depicts the patient flow diagram at Emergency Department with time stamp to understand the Wait Time.

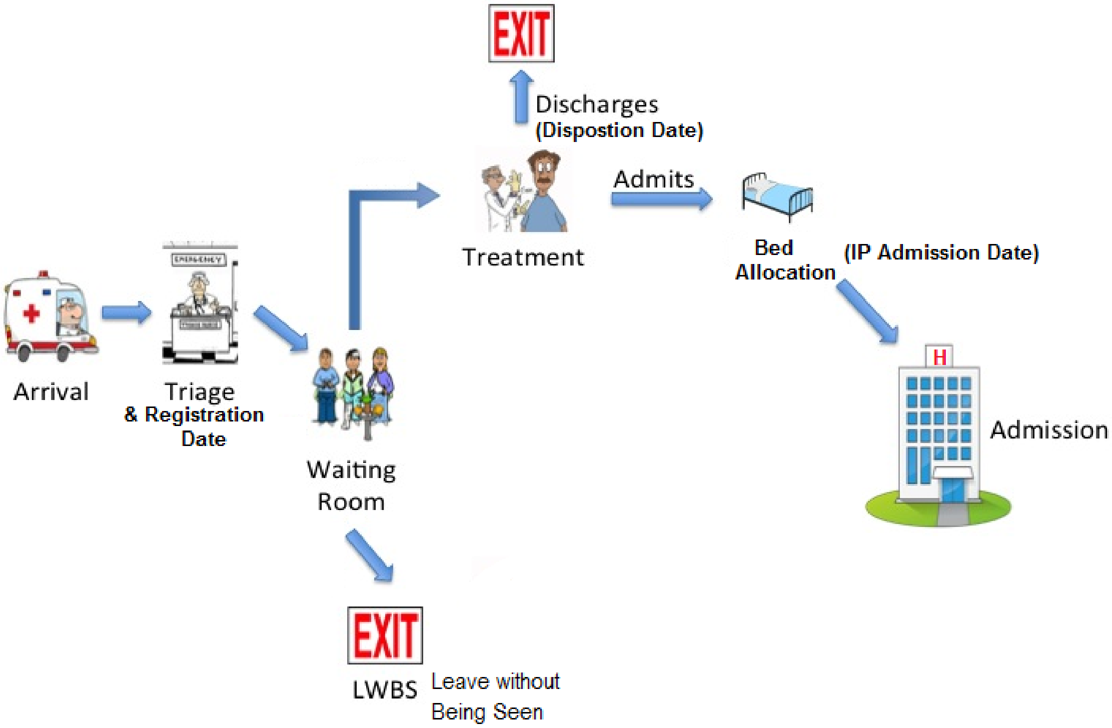


**Figure A.1: Brute Force attack on 4-bit key**

## Appendix B: Appling random number generation function within SQL

Appling random number generation function within SQL Server on numeric data attributes taking in the considerations the ± range of each data element

**Table A.1. Appling random number generation function within SQL**

| SQL Statement | Comment |
|---|---|
| INSERT INTO [CKHA_DW].[dbo].[DART_RawData_Masked] | |
| SELECT | |
| DATEADD(DD,CAST(NEWID() AS binary(6)) %10, [VisitDate]) | -- Visit Date ± 10 days |
| ,[ED_Visits]+CAST(NEWID() AS binary(6)) %10 [ED_Visits] | -- ED Patients Volume ± 10 |
| ,[ED_CTAS1]+CAST(NEWID() AS binary(6)) %3 | -- ED_CTAS1 Volume ± 3 |
| ,[ED_CTAS2]+CAST(NEWID() AS binary(6)) %5 | -- ED_CTAS2 Volume ± 5 |
| ,[ED_CTAS3]+CAST(NEWID() AS binary(6)) %8 | -- ED_CTAS3 Volume ± 8 |
| ,[ED_CTAS4]+CAST(NEWID() AS binary(6)) %10 | -- ED_CTAS4 Volume ± 10 |
| ,[ED_CTAS5]+CAST(NEWID() AS binary(6)) %5 | -- ED_CTAS5 Volume ± 3 |
| ,[IP_Admits]+CAST(NEWID() AS binary(6)) %3 | -- IP_Admits Volume ± 5 |
| ,[ED_ALOS_AllDisp] + CAST(NEWID() AS binary(6)) %2 | -- Avg. LOS for All Disp ± 5 |
| ,[EDALOS_NonAdmit] | -- Avg. LOS for Non Admitted Patients |
| ,[EDALOS_Admits] | -- Avg. LOS for Admitted Patients |
| FROM [Test_DW].[dbo].[DART_RawData] | |
| ORDER BY [VisitDate] | |
| GO | |

# Appendix C: Example of Masking and De-Masking of a numeric field (i.e, Patient Length of Stay LOS) using simple COBAD formulas including 4 variables (2 statistical + 2 random salt keys)

## Masking of LOS

- Formula 1: Simple Masking - Total_LOS_Mask

```
  CAST( [Total_LOS] + (( (Ks1 ^ MaxVal) % Ks2) - MinVal) *0.00000001 AS INTEGER)
```

- Formula 2: LOS Min-Max Normalization - Total_LOS_Masked

```
 CAST((( ([Total_LOS]-MinVal)*1. / (MaxVal-MinVal)*1.) * (newMax -newMin) + newMin)*1.0 AS
INTEGER)
```

- Formula 3: LOS Min-Max Normalization - Total_LOS_Masked1

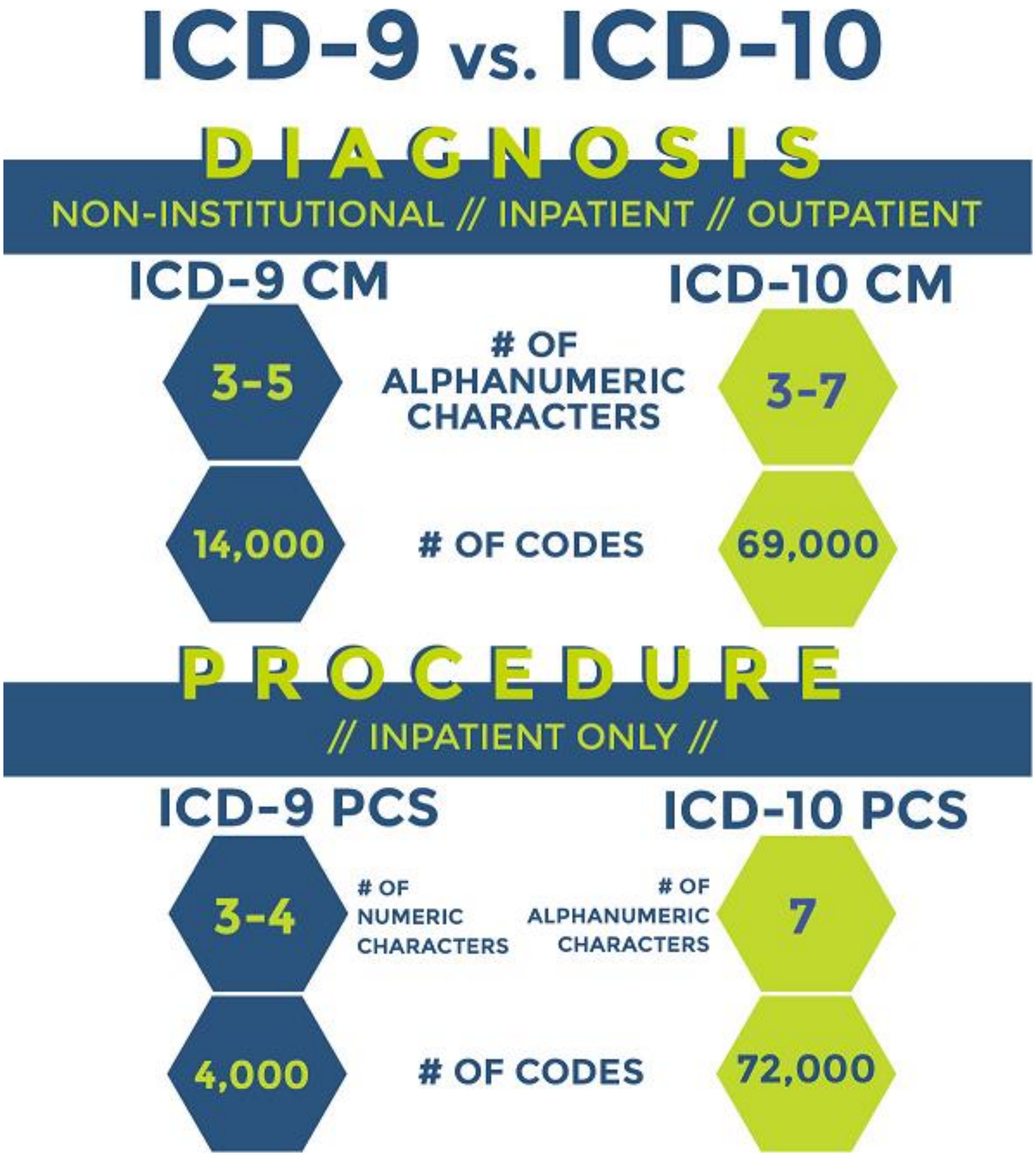
```
 CASE WHEN Ks1<Ks2 THEN
  CAST(((([Total_LOS]-MinVal)*1./(MaxVal-MinVal)*1.)*(Ks2*1./100 -Ks1*1./100)+Ks1*1./1000)*
0.00001 AS INTEGER)
 ELSE
  CAST(((([Total_LOS]-MinVal)*1./(MaxVal-MinVal)*1.)*(Ks1*1./100 -Ks2*1./100)+Ks2*1./1000)*
0.00001 AS INTEGER)
```

## De-Masking of LOS:

- Formula 1: Simple Masking - [Total_LOS_DeMask_F1]

```
  [Total_LOS_Mask]-(( (Ks1 ^ MaxVal) % Ks2) -MinVal) / 100000000
```

## Appendix D: International Statistical Classification of Diseases and Related Health Problems 10th Revision



ICD-9 vs. ICD-10

DIAGNOSIS

NON-INSTITUTIONAL // INPATIENT // OUTPATIENT

ICD-9 CM

ICD-10 CM

3-5

# OF ALPHANUMERIC CHARACTERS

3-7

14,000

# OF CODES

69,000

PROCEDURE

// INPATIENT ONLY //

ICD-9 PCS

ICD-10 PCS

3-4

# OF NUMERIC CHARACTERS

# OF ALPHANUMERIC CHARACTERS

7

4,000

# OF CODES

72,000

**Figure 0.2: Brute Force attack on 4-bit key**

## Appendix E: Using SQL to apply data masking on numeric field after extracting the statistical variables

```
---- Data Masking using Statistical Functions ------
---- Save Masked Data into dbo.Data_Masking_COBAD1 Table ----
Declare @lhin varchar(80)='(01) Erie St. Clair',
      @fy int=2015,
      @fq varchar(2)='Q1',
      @key int = 123456789,
      @SeedKey int = (DATEPART(hh, GETDATE())*10000000) + (DATEPART(n,
GETDATE())*100000) + (DATEPART(s, GETDATE())*1000) + DATEPART(ms,
GETDATE()),
      @pass varchar(128) ='0123456789',
      @new_min int = ABS(CAST(NEWID() AS binary(6)) %10), -- +  1,
      @new_max int = ABS(CAST(NEWID() AS binary(6)) %100);-- +100;
--SELECT @SeedKey, RAND(@SeedKey)*100.0, @new_min , @new_max
--SELECT NEWID(), CAST(NEWID() AS binary(6)), CAST(NEWID() AS
binary(6)) %10,ABS(CAST(NEWID() AS binary(6)) %10)

WITH Stat AS (
SELECT
      [Hospital_Corporation_LHIN]
      ,[Hospital]
      ,[FYear]
      ,[FQtr]
      ,[CMonth]
      ,COUNT([Total_LOS]) CountVal
      ,MAX([Total_LOS]) MaxVal
      ,MIN([Total_LOS]) MinVal
      ,ROUND(AVG([Total_LOS]*1.0),2) AvgVal
      ,SUM([Total_LOS]) SumVal
      ,ROUND(STDEV([Total_LOS]),2) StDevVal
      ,ROUND(VAR([Total_LOS]),2) VarVal
      ,CHECKSUM_AGG(CAST([Total_LOS] AS INT)) CheckSumVal
FROM [DW_STG].[dbo].[EscLHIN_DAD_Main]
WHERE Hospital_Corporation_LHIN = '(01) Erie St. Clair' --@lhin
      AND FYear=2015--@fy
      --AND FQtr = @fq
GROUP BY [Hospital_Corporation_LHIN]
      ,[Hospital]
      ,[FYear]
      ,[FQtr]
      ,[CMonth]
), pass1 AS(
SELECT
      [Encrypted_HN]
      ,AdmitDate
      --,[Discharge Status]
      ,[DischargeDate]
      ,[Sex]
      ,[Dschg_Age]
      ,AgeGrp_5yr
      ,[PostalCode]
      ,m.[Hospital_Corporation_LHIN]
      ,m.[Hospital]
      ,m.[FYear]
      ,m.[FQtr]
      ,m.[CMonth]
      ,[ICD10-CA_Code_MRDx]
      ,m.[Total_LOS]
      --,@new_min newMin
      --,@new_max newMax
      ,ABS(CAST(NEWID() AS binary(2)) %10) + stat.MinVal newMin
      ,ABS(CAST(NEWID() AS binary(2)) %100)+ stat.MaxVal newMax

      ,ABS(CAST(NEWID() AS binary(4)) %2147483647)+1 Ks1 --32553
      ,ABS(CAST(NEWID() AS binary(4)) %2147483647)+1 Ks2
      --,stat.Ks1
      --,stat.Ks2
        ,(DATEPART(hh, GETDATE())*10000000) + (DATEPART(n,
GETDATE())*100000) + (DATEPART(s, GETDATE())*1000) + DATEPART(ms,
GETDATE()) SeedKey

      ,stat.CountVal
      ,stat.MinVal
      ,stat.MaxVal
      ,stat.SumVal
      ,stat.AvgVal
      ,stat.StDevVal
      ,stat.VarVal
      ,stat.CheckSumVal
      --,CONVERT(varbinary(4),stat.MinVal,4)+
CONVERT(varbinary(4),stat.MaxVal,2)+CONVERT(varbinary(4),stat.AvgVal,2)
MaskedKeySignature
FROM [DW_STG].[dbo].[EscLHIN_DAD_Main] m
LEFT JOIN Stat
      ON m.Hospital = stat.Hospital AND m.FYear=stat.FYear AND
m.FQtr=stat.FQtr AND m.CMonth =stat.CMonth --.Hospital_Corporation_LHIN
= stat.Hospital_Corporation_LHIN
WHERE m.Hospital_Corporation_LHIN = @lhin
      AND m.FYear=@fy
)

--TRUNCATE TABLE [DW_STG].[dbo].[Data_Masking_COBAD1]
--INSERT INTO [DW_STG].[dbo].[Data_Masking_COBAD1]
SELECT
      CAST([Encrypted_HN] AS BIGINT) [Encrypted_HN]
      ,AdmitDate
      --,[Discharge Status]
      ,[DischargeDate]
      ,[Sex]
      ,CHAR(CAST((ASCII([Sex])-65 + Ks1)%26 +65 AS INTEGER)) Sex_Masked
      --Age Min-Max Normalization
      ,[Dschg_Age]
      ,CAST(((([Dschg_Age]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax -
newMin)+newMin)*.8 AS INTEGER) Age_Masked
      ,AgeGrp_5yr
        --,case when ks1>ks2 then [Dschg_Age] - ((Ks1 % MaxVal) %Ks2) +
MinVal
      --    else [Dschg_Age] + ((Ks2 % MaxVal) %Ks1) + MinVal
      --    end Age_MaskF1

      ,[PostalCode]
```

```
,LEFT([PostalCode],3)+CHAR(CAST((ASCII(SUBSTRING([PostalCode],4,1))-48
+ Ks1)%10 +48 AS INTEGER)) +
CHAR(CAST((ASCII(SUBSTRING([PostalCode],5,1))-65 + Ks2)%26 +65 AS
INTEGER)) +CHAR(CAST((ASCII(SUBSTRING([PostalCode],6,1))-48 + Ks1)%10
+48 AS INTEGER)) PostalCode_Masked
      ,[Hospital_Corporation_LHIN]
      ,[Hospital]
      ,[FYear]
      ,[FQtr]
      ,[CMonth]

      --,@new_min = ABS(CAST(NEWID() AS binary(6)) %10) + 1 --newMin
      --,@new_max = ABS(CAST(NEWID() AS binary(6)) %100) +100 --newMax

      ,[Total_LOS]
      -- Formula 1: Simple Masking
      ,CAST([Total_LOS]+(((Ks1^MaxVal)%Ks2)-MinVal)*0.00000001 AS
INTEGER) Total_LOS_Mask
      --,[Total_LOS]-((Ks1%maxVal)%CAST(AvgVal AS
INT)^MinVal)+CAST(StDevVal AS SMALLINT) AS Total_LOS_Mask --   -
CAST(StDevVal AS INT)+MaxVal

      -- Formula 2: LOS Min-Max Normalization
      ,CAST(((([Total_LOS]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax -
newMin)+newMin)*1.0 AS INTEGER) Total_LOS_Masked

      ,case when Ks1<Ks2 then CAST(((([Total_LOS]-MinVal)*1./(MaxVal-
MinVal)*1.)*(Ks2*1./100 -Ks1*1./100)+Ks1*1./1000)* 0.00001 AS INTEGER)
                                     else CAST(((([Total_LOS]-
MinVal)*1./(MaxVal-MinVal)*1.)*(Ks1*1./100 -Ks2*1./100)+Ks2*1./1000)*
0.00001 AS INTEGER)
       end Total_LOS_Masked1

        ,SeedKey
      ,newMin
      ,newMax
      ,Ks1
      ,Ks2
      ,CountVal
      ,MinVal
      ,MaxVal
      ,SumVal
      ,AvgVal
      ,StDevVal
      ,VarVal
      ,CheckSumVal

,CONVERT(binary(4),Ks1)+CONVERT(binary(2),MinVal)+CONVERT(binary(2),Max
Val)+CONVERT(binary(4),SumVal)+CONVERT(binary(4),Ks2)+CONVERT(binary(2)
,newMin)+CONVERT(binary(2),newMax)+CONVERT(binary(4),CountVal)
MaskedKeySignature
FROM pass1
--Hospital_Corporation_LHIN
WHERE Hospital_Corporation_LHIN = @lhin
      AND FYear=@fy
      --AND m.FQtr = @fq
      --AND Encrypted_HN = '9903481'
ORDER BY Hospital_Corporation_LHIN
      ,[Hospital]
      ,FYear
      ,FQtr
      ,CMonth
      ,Encrypted_HN
```

## Appendix F: Using SQL to measure re-identification risk for the data-linkage attack

```
WITH pass1 AS(
SELECT [Encrypted_HN]
      ,[AdmitDate]
      ,[DischargeDate]
      ,[Sex]
      ,[Sex_Masked]
      ,CASE WHEN [Sex]=[Sex_Masked] THEN 1 ELSE 0 END SexMatch
      ,[Dschg_Age]
      ,[Age_Masked]
      ,CASE WHEN [Dschg_Age]=[Age_Masked] THEN 1 ELSE 0 END AgeMatch
      ,'Age '+AgeGrp_5yr AgeGrp_5yr
      ,[Age_Grp_5yrs]
        ,CASE WHEN 'Age '+AgeGrp_5yr=[Age_Grp_5yrs] THEN 1 ELSE 0 END
AgeGrpMatch
      ,CASE WHEN [Sex]=[Sex_Masked] AND [Dschg_Age]=[Age_Masked] THEN 1
ELSE 0 END SexAgeMatch
      ,CASE WHEN [Sex]=[Sex_Masked] AND 'Age
'+AgeGrp_5yr=[Age_Grp_5yrs] THEN 1 ELSE 0 END SexAgeGrpMatch
      ,[PostalCode]
      ,[PostalCode_Masked]
      ,CASE WHEN [PostalCode]=[PostalCode_Masked] THEN 1 ELSE 0 END
PostalCodeMatch

      ,CASE WHEN [PostalCode]=[PostalCode_Masked] AND 'Age
'+AgeGrp_5yr=[Age_Grp_5yrs] THEN 1 ELSE 0 END PostalCodeAgeGrpMatch
      ,CASE WHEN [Sex]=[Sex_Masked] AND [Dschg_Age]=[Age_Masked] AND
[PostalCode]=[PostalCode_Masked] THEN 1 ELSE 0 END
SexAgePostalCodeMatch
      ,CASE WHEN [Sex]=[Sex_Masked] AND 'Age
'+AgeGrp_5yr=[Age_Grp_5yrs] AND [PostalCode]=[PostalCode_Masked] THEN 1
ELSE 0 END SexAgeGrpPostalCodeMatch
        ,CASE WHEN LEFT([PostalCode],3)=LEFT([PostalCode_Masked],3) AND
'Age '+AgeGrp_5yr=[Age_Grp_5yrs] AND [Sex]=[Sex_Masked] THEN 1 ELSE 0
END FSACodeAgeGrpSexMatch
        ,CASE WHEN LEFT([PostalCode],3)=LEFT([PostalCode_Masked],3) AND
'Age '+AgeGrp_5yr=[Age_Grp_5yrs] THEN 1 ELSE 0 END FSACodeAgeGrpMatch
        ,CASE WHEN LEFT([PostalCode],3)=LEFT([PostalCode_Masked],3) AND
[Sex]=[Sex_Masked] THEN 1 ELSE 0 END FSACodeSexMatch

      ,[Hospital_Corporation_LHIN]
      ,[Hospital]
      ,[FYear]
      ,[FQtr]
      ,[CMonth]
      ,[Total_LOS]
      --,[Total_LOS_Mask]
      ,[Total_LOS_Masked]
      ,CASE WHEN [Total_LOS]=[Total_LOS_Mask] THEN 1 ELSE 0 END
LOSMatch
      ,[Total_LOS_Masked1]
      ,CASE WHEN [Total_LOS]=[Total_LOS_Masked1] THEN 1 ELSE 0 END
LOS1Match
      ,[SeedKey]
      ,[newMin]
      ,[newMax]
      ,[Ks1]
      ,[Ks2]
      ,[CountVal]
      ,[MinVal]
      ,[MaxVal]
      ,[SumVal]
      ,[AvgVal]
      ,[StDevVal]
      ,[VarVal]
      ,[CheckSumVal]
      ,[MaskedKeySignature]
FROM [DW_STG].[dbo].[Data_Masking_COBAD1] ds
      LEFT JOIN [LHIN_DW_DAD].[Core].[DimAgeGroup] ag
            ON ds.[Age_Masked] = ag.AgeKey
)

SELECT
      [FYear]
      ,[FQtr]
      ,COUNT(*) rec_cnt
      ,SUM(SexMatch) [#Sex]
      ,ROUND(SUM(SexMatch)*1./COUNT(*)*100.,3) [%Sex Match]
      ,SUM(AgeMatch) [#Age]
      ,SUM(AgeMatch)*1./COUNT(*)*100. [%Age Match]
      ,SUM(AgeGrpMatch) [#AgeGrp]
      ,SUM(AgeGrpMatch)*1./COUNT(*)*100. [%AgeGrp Match]

      ,SUM(PostalCodeMatch) [#PostalCode]
      ,SUM(PostalCodeMatch)*1./COUNT(*)*100. [%PostalCode Match]
      ,SUM(PostalCodeAgeGrpMatch) [#PCodeAgeGrp]
      ,SUM(PostalCodeAgeGrpMatch)*1./COUNT(*)*100. [%PCodeAgeGrp Match]

      ,SUM(SexAgeMatch) #SexAge
      ,SUM(SexAgeMatch)*1./COUNT(*)*100. [%SexAge Match]
      ,SUM(SexAgeGrpMatch) #SexAgeGrp
      ,SUM(SexAgeGrpMatch)*1./COUNT(*)*100. [%SexAgeGrp Match]
      ,SUM(SexAgePostalCodeMatch) #SexAgePCode
      ,SUM(SexAgePostalCodeMatch)*1./COUNT(*)*100. [%SexAgePCode Match]
      ,SUM(SexAgeGrpPostalCodeMatch) #SexAgeGrpPCode
      ,SUM(SexAgeGrpPostalCodeMatch)*1./COUNT(*)*100. [%SexAgeGrpPCode
Match]

      ,SUM(FSACodeAgeGrpMatch) #FSACodeAgeGrpMatch
      ,SUM(FSACodeAgeGrpMatch)*1./COUNT(*)*100. [%FSACodeAgeGrp Match]
      ,SUM(FSACodeSexMatch) #FSACodeSexMatch
      ,SUM(FSACodeSexMatch)*1./COUNT(*)*100. [%FSACodeSex Match]
      ,SUM(FSACodeAgeGrpSexMatch) #FSACodeAgeGrpSexMatch
      ,SUM(FSACodeAgeGrpSexMatch)*1./COUNT(*)*100. [%FSACodeAgeGrpSex
Match]
FROM pass1
GROUP BY FYear
      ,[FQtr]
ORDER BY FYear

      ,[FQtr]
```

## Appendix G: Using SQL to de-mask the masked dataset

```
---- De-Masking data by using Reversible Functions ------
---- Load Masked Data from dbo.Data_Masking_COBAD1 Table ----
SELECT --TOP 10000
        [Encrypted_HN]
      ,[Sex]
      -- Masking Formula for Sex
      --,CHAR(CAST((ASCII([Sex])-65 + Ks1)%26 +65 AS INTEGER))
Sex_Masked
      ,[Sex_Masked]
      -- De-Mask Formula for Sex
      ,CHAR(CAST((ASCII(Sex_Masked)-65-
cast(substring([MaskedKeySignature],1,4) as int))%26 +65+26 AS
INTEGER)) Sex
      ,[Dschg_Age]
        -- MAsking formula for Age
        --,CAST(((([Dschg_Age]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax -
newMin)+newMin)*.8 AS INTEGER) Age_Masked
      ,[Age_Masked]
        ,case when ks1>ks2 then [Dschg_Age] + ((Ks1 % MaxVal) %Ks2)*.1
- MinVal
             else [Dschg_Age] + ((Ks2 % MaxVal) %Ks1) - MinVal
         end Age_MaskF1

        -- De-Mask Formula for Age
        --,- ((K3, j MOD K1) MOD K2, i) + K2, i
      --,(((([Dschg_Age]-MinVal)/(MaxVal-MinVal))*(newMax -
newMin)+newMin)+newMin)*5
      --,((([Dschg_Age]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax -
newMin)+newMin)
        ,case when CAST(((([Age_Masked]-newMin)*1./(newMax-
newMin)*1.)*(MaxVal -MinVal)+MinVal)/.8 AS INTEGER) >0
                        then CAST(((([Age_Masked]-newMin)*1./(newMax-
newMin)*1.)*(MaxVal -MinVal)+MinVal)/.8 AS INTEGER) +1
                  when CAST(((([Age_Masked]-newMin)*1./(newMax-
newMin)*1.)*(MaxVal -MinVal)+MinVal)/.8 AS INTEGER) <0 then 0
                  else CAST(((([Age_Masked]-newMin)*1./(newMax-
newMin)*1.)*(MaxVal -MinVal)+MinVal)/.8 AS INTEGER)
            end DeMask_Age

      ,[Total_LOS]
      -- 1st Masking Formula
      -- CAST([Total_LOS]+(((Ks1^MaxVal)%Ks2)-MinVal)*0.00000001 AS
INTEGER) Total_LOS_Mask
      ,[Total_LOS_Mask]
        ,[Total_LOS_Mask]-(((Ks1^MaxVal)%Ks2)-MinVal)/100000000
[Total_LOS_DeMask_F1]
      -- LOS Min-Max Normalization
      -- CAST(((([Total_LOS]-MinVal)*1./(MaxVal-MinVal)*1.)*(newMax -
newMin)+newMin)*1.0 AS INTEGER) Total_LOS_Masked
      ,[Total_LOS_Masked]


      -- De-Mask LOS
      ,[Total_LOS_Masked1]
      --if Ks1<Ks2 then CAST((((([Total_LOS]-MinVal)*1./(MaxVal-
MinVal)*1.)*(Ks2*1./100 -Ks1*1./100)+Ks1*1./100)* 0.1 AS INTEGER)
      ,case when Ks1<Ks2 then
                   (([Total_LOS_Masked1]*10.)-Ks1*1./100)/(Ks2*1./100 -
Ks1*1./100)*(MaxVal-MinVal)+MinVal
            --else
                   --CAST((((([Total_LOS]-MinVal)*1./(MaxVal-
MinVal)*1.)*(Ks1/100 - Ks2/100)+Ks2/100* 0.1) AS INTEGER)
       end Total_LOS_DeMasked1
        --Retrive the values from Masking Signature
        --,[Ks1]
      ,cast(substring([MaskedKeySignature],1,4) as int) Ks1
      --,[MinVal]
      ,cast(substring([MaskedKeySignature],5,2) as int) MinVal
      --,[MaxVal]
      ,cast(substring([MaskedKeySignature],7,2) as int) MaxVal
      --,[SumVal]
      ,cast(substring([MaskedKeySignature],9,4) as int) SumVal
      --Convert Varbinary to 4 bytes float
      --,SIGN(CAST(substring([MaskedKeySignature],7,4) AS INT)) * (1.0
+ (CAST(substring([MaskedKeySignature],7,4) AS INT) & 0x007FFFFF) *
POWER(CAST(2 AS REAL), -23)) * POWER(CAST(2 AS REAL),
(CAST(substring([MaskedKeySignature],7,4) AS INT) & 0x7f800000) /
0x00800000 - 127) AvgVal
      --,Ks2
      ,cast(substring([MaskedKeySignature],13,4) as int) Ks2
      --,[newMin]
      ,cast(substring([MaskedKeySignature],17,2) as int) newMin
      --,[newMax]
      ,cast(substring([MaskedKeySignature],19,2) as int) newMax
      --,[CountVal]
      ,cast(substring([MaskedKeySignature],21,4) as int) CountVal
      ,[AgeGrp_5yr]
      ,[PostalCode]
      ,[PostalCode_Masked]
      ,[Hospital_Corporation_LHIN]
      ,[Hospital]
      ,[FYear]
      ,[FQtr]
      ,[CMonth]
      ,[SeedKey]
      ,[AvgVal]
      ,[StDevVal]
      ,[VarVal]
      ,[CheckSumVal]
      ,[MaskedKeySignature]

FROM [DW_STG].[dbo].[Data_Masking_COBAD1]
--WHERE Encrypted_HN='9903481'
ORDER BY Hospital_Corporation_LHIN
      ,[Hospital]
      ,FYear
      ,FQtr
      ,CMonth
      ,Encrypted_HN
```

# Curriculum Vitae

| | |
|---|---|
| **Name:** | Osama Ali -Ozkan |
| **Post-secondary Education and Degrees:** | University of Western Ontario<br>London, Ontario, Canada<br>2012-2018 Ph.D. Electrical and Computer Engineering |
| | Fanshawe College<br>London, ON<br>2009-2011 Project Management Diploma |
| | Al-Nahrain University<br>Baghdad, Iraq<br>1991-1994 M.Sc. Computer Science |
| | University of Mosul<br>Mosul, Iraq<br>1985-1989 Hon. B.Sc. Physics Science (Ranked 1st ) |
| **Honours and Awards:** | Diane Y. Stewart Endowed Scholarship Award:<br>London Health Science Centre / Innovative Idea Award<br>Design of a software solution for Emergency room<br>2013 |
| | Best Paper Award<br>IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)<br>British Columbia University<br>2016 |
| **Related Work Experience:** | Business Intelligence and Decision Support Specialist<br>Erie St. Clair LHIN/ Chatham-Kent Hospital<br>2014-present |
| | Part time Professor<br>Fanshawe College (London, ON)<br>2018-present |
| | Decision Support Analyst<br>London Health Sciences Centre<br>2007-2014 |

Lecturer/ Programmer/ Project Lead
Higher Institute of Technical Vocations/
AlGabal Al-Gharby University/ Computer Science Department
Libya
1997-2006

**Publications:**

Ali O, Ouda A, “Preserving Privacy in a Business Intelligence Analytics Platform: an iMaskU Framework with Content-Based Data Masking Technique“, Journal of Information Security and Applications, Elsevier, November 2018, (under review).

Ali O, Ouda A, “A Content-Based Data Masking Technique for A Built-In Framework in Business Intelligence Platform”, IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE 2017

Ali O, Crvenkoviski P, Johnson H, “Using a Business Intelligence Data Analytics Solution in Healthcare A case study: Improving Hip Fracture Care Processes in a Regional Rehabilitation System”, 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE 2016.

Ali O, Ouda A, “A Classification Module in Data Masking Framework for Business Intelligence Platform in Healthcare”, 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE 2016.

Ali O, Capretz L, Bou-Nassif “Business intelligence solutions in healthcare, a case study: Transforming OLTP system to BI solution”, IEEE (ICCIT) Third International Conference on Communications and Information Technology, 2013.

Ali O, Alfigi B, "Toward of Design an Expert System for Machine Translation System", 2nd Annual Translation Conference, Academy of Graduate Studies, Benghazi-Libya, 2006

Text Book: "8086/8088 Assembly Language Programming", Dar-Aljamieyah Library, Libya, 2005.

Ali O, "Design of a Lexicon for Machine Translation System", 1st Annual Translation Conference, Academy of Graduate Studies, Tripoli-Libya, 2005

Ali O, "Design of Intelligent System for Natural Language Understanding", Industrial Researches Journal, Libya, Vol. 5, issue 9, page 35, 2000