Master in Informatics Engineering

MSc Dissertation

Data Privacy Protection for the Cloud

Paulo Miguel Guimarães da Silva

pmgsilva@student.dei.uc.pt

Friday 1st July, 2016

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

University of Coimbra

Faculty of Science and Technology
of the University of Coimbra

Department of Informatics Engineering

Master Dissertation

Data Privacy Protection for the Cloud

Paulo Miguel Guimarães da Silva

Supervised by

Edmundo Monteiro, University of Coimbra
Lukas Kencl, Czech Technical University in Prague

Members of the jury

Arguing member: Carlos Nuno Laranjeiro
Vowel: João P. Vilela

Friday 1st July, 2016

Acknowledgments

A research project like this Master Dissertation is never the work of anyone alone. The contributions of many different people, in different ways and different places, have made this possible. I would like to extend my appreciation especially to the following.

Firstly, I would like to thank my supervisors, Dr. Edmundo Monteiro and Dr. Lukas Kencl. Without them, this would not be possible. The interest and availability demonstrated from the first moment I approach them with the plan, even with such a short notice to set up paperwork, was crucial. For all the help, dedication, guidance and expertise they provided me, I owe them all my appreciation.

To the members of the jury, Dr. Carlos Nuno Laranjeiro and Dr. João P. Vilela for their insightful comments, encouragement and questions that helped me achieve a better final work.

I would like to thank Dr. Robert Bestak. He also helped with my arrival, the paperwork, office preparation and everything I needed throughout my staying there.

Dr. Martin Loebl for his time to receive me, for providing me with practical advises, exchange of ideas and further encouragement to achieve my objectives.

The days would have passed far more slowly without the support of my friends and colleagues, both in The Czech Republic and Portugal. They helped me overcome difficulties and maintain the focus on my work. I am also grateful to those who helped me to adjust to a new country and a new language.

I wish to thank my family for the love and support they provided me through my life, without whom I would never have enjoyed so many opportunities. I must equally acknowledge my girlfriend and best friend, Simona, who led me to think of this idea in the first place, and without whose love, encouragement and patience, I would not have finished this work.

To the Informatics Engineering Department of the University of Coimbra I thank for letting this happen and for the outstanding academic background they provided me until now.

To the Czech Technical University in Prague and the Faculty of Electrical Engineering I must thank for receiving me and providing me what I needed during my stay in The Czech Republic.

Finally, I would like to thank the EUBra-BIGSEA project. A European Union's Horizon 2020 research and innovation programme under grant agreement No 690116, which provided me financial support.

Resumo

A privacidade é uma preocupação constante sempre que se discute acerca da transmissão ou armazenamento de dados. Atualmente, com o aumento constante de dados pessoais e confidenciais a serem transmitidos ou armazenados *online*, os responsáveis pela gestão dos dados devem assegurar certas garantias de privacidade e proteção dos dados.

Esta Dissertação de Mestrado apresenta diversas técnicas de anonimização e ocultação disponíveis. São descritas as suas características e funcionalidades, bem como ferramentas para as implementar e avaliar os dados gerados. O objetivo principal deste trabalho é a avaliação da aplicabilidade do algoritmo de ocultação inspirado no ADN ("*DNA-inspired information concealing algorithm*").

Geralmente são utilizados um conjunto de métricas para medir aspetos como o risco ou a utilidade dos dados gerados. Esta Dissertação de Mestrado apresenta uma nova abordagem de avaliação dos dados gerados. Ao utilizar a Semelhança do Cosseno ("*Cosine Similarity*") como uma medida de semelhança entre os dados privados e ocultados, esta revela-se extremamente útil. Não só para a procura de informação ("*Information Retrieval*") ou mineração de texto, mas também para a análise de dados anonimizados ou ocultados.

Atualmente há uma procura crescente de serviços e de armazenamento na nuvem. A avaliação feita nesta Dissertação de Mestrado procurar descobrir o quão adequada é a aplicação do algoritmo de ocultação de informação inspirado em ADN nos dados que são armazenados ou transmitidos na nuvem.

A avaliação é feita através da análise dos resultados obtidos na ocultação dos dados bem como do desempenho do próprio algoritmo. O algoritmo é aplicado em diversos ficheiros de texto e áudio com diferentes características. No entanto, os dois tipos de ficheiro são de dados não estruturados, por sua vez aceite no algoritmo em questão. Ao contrário de grande parte dos algoritmos de anonimização que exigem dados estruturados.

Com os resultados finais e respetiva análise será possível determinar a aplicabilidade e desempenho do algoritmo para uma possível integração na nuvem.

Palavras-chave

Privacidade e Ocultação de Dados, "Algoritmo de ocultação inspirado em ADN", Computação na Núvem

Abstract

Privacy is for a long time a concern when data is being discussed. Nowadays, with an increasing amount of personal and confidential data being transmitted and stored online, data curators have to assure certain guarantees of data protection and privacy.

This Master Dissertation presents a background of anonymization and concealing techniques. Their characteristics and capabilities are described, as well as tools to implement and evaluate anonymization and concealing. The evaluation of the applicability of the DNA-inspired concealing algorithm is the main objective of this work.

Usually, various metrics are used to measure aspects like risk or utility of the anonymized data. This work presents a new approach of evaluating how well concealed is the data. By using the Cosine Similarity as a measure of similarity between the private and concealed data, this metric proves its worthiness not only in information retrieval or text mining applications but also in the analysis of concealed or anonymized files.

Nowadays there is a continuously growing demand for Cloud services and storage. The evaluation in the Master Dissertation is directed to find how suitable is the application of the DNA-inspired concealing algorithm over the data being stored or transmitted in the Cloud.

The evaluation is made by analyzing the concealing results as well as the performance of the algorithm itself. The application of the algorithm is made over various texts and audio files with different characteristics, like size or contents. However, both file types are unstructured data. Which is an advantage for being accepted as an input by the algorithm. Unlike many anonymization algorithms which demand structured data.

With the final results and analysis, it will be possible to determine the applicability and performance of the referred algorithm for a possible integration with the Cloud.

Keywords

Data Concealing and Privacy, "DNA-inspired concealing algorithm", Cloud Computing

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This work presents an evaluation of the DNA-inspired information concealing algorithm for data privacy protection and its applicability to the cloud. The following sections and chapters of the document contain the all information relative to the study of the applicability of the DNA-inspired information concealing algorithm [Kencl and Loebl, 2010].

The evaluation is made through the analysis of the concealing of various experiments and different inputs. Additionally, some performance metrics like execution time and size complexity are also analyzed in order to conclude how applicable this algorithm is with cloud services as online storage for example.

The motivation for this work if presented in the next section. The objectives are stated right after. Followed by the contributions and structure sections.

1.1 Motivation

Several methods and techniques to anonymize and conceal data have been proposed over the time. In the nineties, when these methods started to emerge, different approaches have been presented in order to give the desired anonymization. The scope of the methods varies from making a complete modification of the data, making slight changes or just concealing in specific ways. It was evident that the more anonymized it was, the less utility the data could provide. The opposite is also verified, with more utility comes less anonymity.

Since the goal of the anonymization research field was to anonymize and provide data with useful retrievable information, new algorithms and techniques started to be presented.

Currently with improved versions and other combinations of techniques, K-Anonymity [Sweeney, 2002b] was, and still is, an important base model in the anonymization field. Other methods were introduced over the time: l-diversity, t-closeness, generalization, differential privacy and others [Sweeney, 2000].

The DNA-inspired information concealing algorithm [Kencl and Loebl, 2010] was presented in 2010. While many techniques permanently suppress content and consequently lose parts of the information, the referred algorithm does not. Also, most anonymization techniques and algorithms work based on inputs of structured data like network traces or logs, which is not the case of the DNA-inspired information concealing algorithm that also conceals unstructured data like text or audio for instance.

The algorithm, by preserving and maintaining families of repeats, is capable of concealing data in a different fashion than the methods referred above. However, a full evaluation of the applicability of the algorithm and its performance is not available. Therefore, the current work, aims to provide a thorough analysis and evaluation of the method regarding its applicability and performance.

Cloud computing it is not an entirely new concept. Many well know companies already provide these kind of services and infrastructure for some time now. As times passes, more and more companies follow the move and provide services in the cloud.

Every day, commercial users, business or academics rely on this kind of services. Not only for application or computation power, but as storage, like a database for instance. Therefore, it is crucial to provide services with quality, security and privacy.

With the non-stop growing of cloud services and demand, privacy concerns naturally arise. There are four types of cloud infrastructures: Public, Private, Community or Hybrid. [Microsoft, 2014]. Each one for different needs but similar safety and privacy requirements.

There are many methods to increase privacy levels online, from avoiding the storage of sensitive information online to encryption of the data. By developing this work, the author intends to provide an analysis of alternative ways of providing data privacy and its possible cloud applications.

1.2 Objectives

The research presented in the Master Dissertation has the main objective of testing the applicability of the DNA-inspired information concealing algorithm both in concealing and cloud application.

In order to do that, several experiences using the DNA-inspired concealing algorithm will be performed, focusing on unstructured data like text or audio files, like, for instance emails contents or an audio recording.

When concluded the experiences, there will be an evaluation. The performance of the algorithm in terms of its concealing capabilities will be evaluate recurring to the Cosine Similarity metric, which is a function of the angle between the two files (vectors) in the term vector space. Its execution capabilities such as time and size complexity of the generated output are going to be evaluated as well.

With the analysis complete it will be possible to discuss the applicability of the algorithm on the referred type of data. As well as evaluate how does it perform and how suitable it is for usage in cloud services, for instance, online storage.

1.3 Contributions

A description and characterization of existent techniques and algorithms available to perform anonymization and concealing of data sets is included in a comprehensive state of the art. To complement the previous, there is equally a description of tools and metrics available to apply and evaluate the anonymization and concealing of data.

Being most of the techniques and algorithms developed mainly for the anonymization of structured data, the work presented in this Master Dissertation shows the application of a new algorithm and the concealing of unstructured data like text or audio (i.e. email contents or voice recording).

The contributions of the author for this work are as follows:

1. A State of the Art review with definitions, algorithms, tools and metrics relevant in the field.

2. Experiments with different data sets of text and audio. Application of the DNA-inspired information concealing algorithm.

3. An evaluation of the concealing algorithm recurring to the Cosine Similarity metric (Section 2.3.1) with different settings and variation of parameters.

4. To complement the performance evaluation, measure of execution times and the size complexity of the generated files in comparison with the original private files are taken.

5. An analysis and discussion of the algorithm's performance and applicability for the cloud.

The work developed provides a novel and comprehensive study about the referred algorithm. This allows a conclusion to be drawn of both the algorithm's concealing capabilities and its possible cloud application.

1.4 Structure

The document is organized and divided into six chapters. The Introduction (Chapter 1), where a brief description of the topic is made, what is to be evaluated, the objectives, the contributions of the author and how the report is organized and structured.

Then, a chapter Anonymization and Concealing (Chapter 2) with a background of what has been made in this field and a description of methods and techniques available. As well as tools that can be used to implement some of the algorithms and measure some metrics to evaluate the data produced after the application of those techniques.

In the Research Approach (Chapter 3) there is a detailed explanation about the experiments performed, how was the approach, how to achieve them and analyze the results. The data and tools used during the whole process are also described.

The Experimental Results (Chapter 4) contain the results obtained and gathered from the experimental tests. The results are presented with the support of graphics and the respective description and analysis.

There is the Applicability to the Cloud (Chapter 5), where a discussion of the cloud applicability of the work is made.

The last chapter Conclusion (Chapter 6) is where a discussion and final considerations are presented. As well as the future work. Follow by the References and Appendices.

The Appendices contain the Work Plan and additional results from the experiments. The Work Plan (Appendix A) is divided in two sections as the Dissertation is also divided in two main parts: one in Portugal (First Semester - Section A) and another in The Czech Republic (Second Semester - Section A). In the referred Sections there is a description of the entire work plan and how was it managed. Contains also a description and explanation of the deviations from the initial plan that occurred during the second semester.

CHAPTER 2

Anonymization and Concealing

With the increasing amount of information being generated and uploaded every second, privacy guarantees have to be assured. To achieve that, there are numerous algorithms and techniques created with the purpose of making data anonymous or to concealing it providing certain guarantees of privacy.

Over the last years, several techniques to provide privacy have been researched and presented in the scientific community. However, a consensus has not yet been reached about the proper way to ensure privacy to specific data because the usability requirements are different in every case.

The Anonymization and Concealing Chapter analyzes several anonymization and concealing techniques. Tools to implement them and metrics to measure certain parameters as anonymization or usability. The explanation and listing of the existing techniques and tools is intended not only to provide a solid background of research field but also to refer the achievements of the work being made in the area and also to be a part of the experimental work.

This chapters starts by stating definitions, followed by techniques and algorithms for anonymization and concealing, as well as tool used for the effect.

2.1 Definitions

Below there is a description of the terms being referred in the document:

- Anonymization

  "A process that removes or replaces identity information from a communication or record." For instance, a subject in in communications or records can be made pseudonymous. Then, the same subject will always have the same replacement identity but cannot be identified. [Daintith, 2016]

- Concealing

  Conceal is act of keeping from sight, to hide. By keeping secret or prevent from being known or noticed, one is concealing something. [Petitcolas et al., 1999]

- Data curator

  "Data curation is the active and on-going management of data through its life cycle of interest and usefulness to scholarship, science, and education; curation activities enable data discovery and retrieval, maintain quality, add value, and provide for re-use over time." [Cragin et al., 2007] A data curator is the person in charge of management and application of the previously referred.

- Quasi identifier

  A quasi-identifier is an attribute of the private information that can be linked with external information. There are some identifiers like a person's name or address which are explicit, however a quasi-identifier is an attribute that combined with others can identify individuals. [Sweeney, 2002b]

- Data Utility (Usability)

  After the anonymization process, there is the matter of the utility of the information, which is of high importance. The utility or usability, is the representational value of the amount of information that is preserved in the anonymized data. [Sweeney, 2002b]

- Data Privacy

  Information privacy is the ability of an individual or group to stop information about themselves from becoming known to people other than those they choose to give the information to. Privacy is sometimes related to anonymity although it is often most highly valued by people who have private data publicly known. [Solove, 2009]

- Data Confidentiality

  According to the Oxford dictionary, confidential means: "intended to be kept secret" [Wiles et al., 2008]. Meaning that confidential information is information that is intended to be kept secret. Can be seen as a set of rules that limits access or imposes restrictions on certain types of information. Thereby, providing data confidentiality means to kept data secret.

With the description of the definitions above, it is possible to provide some examples of anonymization processes and researches.

It was proven [Sweeney, 2000] that combining birth date, ZIP code and gender (which are not explicit identifiers) is possible to uniquely identify 87% of the individuals in the United Stated of America. The goal in the anonymization field is to release person-specific data, such the ability to combine quasi-identifiers and link with external information is limited.

Recently, a study [Mivule and Anderson, 2015] with a hybrid anonymization heuristic was made. The application aimed towards more utility along with a good barrier against de-anonymization by an attacker. In the study there were two attributes to privatize: time stamp and IP address. Since the nature of the attributes is different, two distinct methods are used in the anonymization.

The Enumeration Technique [Slagell et al., 2006], along with the Multiplicative Noise addiction [Mivule and Anderson, 2015] was the method used for the *timestamp* in order to preserve the flow structure. Since each one of the four octets of the IP addresses are limited to a range of 0 and 255 and maintaining full IP address prefixes is prone to de-anonymization attacks this a challenging attribute to anonymize. To go over this limitation, using partial prefix-preserving heuristic was the solution. The IP address is partitioned into four octets:

- 1^{st} octet
  In the first octet generalization is applied to preserve the prefix. A transformation is done by multiplying or adding noise. After the process k-anonymity is also applied to assure that no unique values appear and that the values appear k >1 times. In the end users of the data are given some usability by getting a synthetic flow of the IP address.

- 2^{nd}, 3^{rd}, 4^{th} octets
  To the remaining octets, differential privacy is applied with constrains regarding the range (0-255). The final step was to put back together the four octets, resulting in an anonymized IP address.

With trade-offs from both parts, privacy and usability, generating anonymized data is possible, on a case-by-case basis, considering the usability and privacy requirements. With these requirements data curators can model the most suitable approach to each case, assuring the usability and preventing the leaking of information.

In the past decade there was a noticeable growth in research for anonymization techniques and ways of concealing data and provide privacy. The intrinsic objective within the research was to provide anonymization with a certain level of usability. It is not possible to ensure a completely privatized network data set without compromising its usability at some level. On the other hand, to give more usability, data privacy is at risk. Data curators have to find a balance between the privacy and the usability, so in each case, a trade-off has to be made when applying micro data privatization techniques.

It is difficult to suggest a general technique. The unique usability requirements turn this matter in a case-by-case analysis. In order to decide what kind of technique should be used in the privatization of the data traces attributes, it is essential know which ones are available. After understanding each one of them it is possible to apply the appropriated techniques to the attributes, separately, to ensure a better privatization.

There are several anonymization and concealing techniques, each one of them with its particular characteristics, features and limitations. In the section below, Techniques and Algorithms (Section 2.2), there is a description of some of the techniques available. Furthermore, there is a description about how do they work, their characteristics and examples.

2.2 Techniques and Algorithms

Below, it is a description of the most common techniques and algorithms used in the anonymization and concealing process. Some of the techniques described will be used in the research approach (Chapter 3).

2.2.1 Structured Data

Most of the techniques are pointed to structured data like network logs or traces, where data usually organized and formatted with specific fields. Accordingly, the following section contains the techniques and algorithms focused on structured data.

2.2.1.1 Black Marker

The Black Marker technique [Slagell et al., 2006] (or hiding), in theory it is like going over with a black marker over some field of information in order to conceal it. Though is a simple way of privatization, is a very strong algorithm cause simply by replacing fields by $NULL$ or 0 is an effective way of hiding sensitive information. Nevertheless, this technique does not provide a good usability level of the data due to the concealing of the information.

For instance, when applying this technique over an IP address like 192.168.87.98, it would come up with 192.168.87.0 or 192.168.0.0, depending on the level of anonymization needed.

2.2.1.2 Truncation

The Truncation technique [Slagell et al., 2006] may generate some confusion with the Black Marker duo to their similarity in operation. However, the difference of the truncation over the black marker is that the first shortens the values, while the black marker

maintains the length or structure of the privatized data. In this technique the work is done by taking a field of the data and selecting a point after which all bits are removed.

To provide a better understanding of how truncation works, two examples which can point out the difference between this and the black marker follow: the case of a string and of an IP address. In the case of an email address (string), is possible to truncate the domain information, so instead of having for example *student@domain.pt* the result could be just "student".

Truncation shorten strings, like previous example suggests, but this can be a problem for fixed length binary values. In the case of privatization of IP addresses like in the example above, it is possible to replace the intended fields with 0, however that could be called the black marker technique.

To go over this similarity with black marker, having in mind that truncating is shortening a value and deal with the binary values with fixed lengths, it is possible to choose a point after which the truncation is intended, for example 192.168.0.0, and apply right shifts until all bits to the right of the chosen point are shifted off the end, resulting in a truncated IP address like 0.0.192.168.

2.2.1.3 Enumeration

The Enumeration technique [Slagell et al., 2006] works by mapping each original value to a new value such that the order is preserved. It is a very general algorithm and would work in any well-ordered set. This algorithm is as straightforward as it looks, let's take as an example the following sequence: $3.6, 16.8, 21, 0.9, 27.9, 14.4$. When the enumeration algorithm is applied to this sequence, could generate a sequence like: $3, 12, 15, 3, 18, 9$, where the order is preserved but any specific information is removed.

Considering the example, it is possible to see that the order is preserved. This means that sorting an anonymized or private sequence would produce the same result in terms of order.

2.2.1.4 Partitioning

By using the Partitioning technique [Slagell et al., 2006], the fields chosen to anonymized are partitioned into meaningful sets. Afterwards the actual values are replaced with a fixed value from the same set. Taking as an example the TCP ports $(0-65535)$ a possible solution could be to have the port numbers within 0 and 4095 replaced with a 0, and ports within the other set, 4096 and 65535, replaced with 65535.

The black marker and truncation techniques are special cases of partitioning.

2.2.1.5 Precision degradation

Precision degradation [Mivule and Anderson, 2015] is a generalization technique that removes the most precise components of a certain field (replacing by 0 for instance). Consider the time stamp field of some data set, it is possible to have different levels of precision (days, hours, minutes, seconds, or milliseconds).

Taking the milliseconds as an example, the precision is high. However, choosing the hours or minutes, it is a low precision degradation which is equivalent to the black marker technique.

Assuming a high precision degradation (milliseconds) is applied to the following time stamp 1000001001, a result would be 1000001000 what can potentially collapse some time stamps into one. For instance, if the precision is low, minutes or seconds, a resulting time stamp would be 1000000000, which is identical to the black marker technique.

2.2.1.6 Prefix-preserving

The prefix-preserving anonymization technique [Boschi and Trammell, 2011] is a special type of permutation. It is classified like that due to the direct substitution technique that enforces but with the restriction of having to preserve the structure of the value.

Following with the examples of IP addresses, given two private addresses that match on a prefix of n bits, the two anonymized IP addresses that will be generated will match on a prefix of n bits as well (example: Table 2.1). Meaning that the structure of sub nets is preserved at each level while anonymizing IP addresses.

Private IP		Anonymized IP	
Source	Destination	Source	Destination
112.116.186.8	115.23.40.51	235.251.46.4	240.48.153.85

Table 2.1: Prefix-preserving example on IP Address

2.2.1.7 Hash

Hashing functions [Slagell et al., 2006] can be very useful for anonymization of both text and binary data. What a hash function does is the mapping each value to a new value. Not necessarily unique, as the permutation. The limitation, with binary data for example, is that truncating the result of a hash function to the shorter length of the value is often required. Consequently, the hash function is weaker and suitable to more collisions.

Nowadays it is possible that an attacker possesses a table of hashes of every possible IPv4 address, which turns dictionary attacks very practical. Moreover, the hash functions must be used carefully and just if it is possible to have collisions within the anonymized data. This is the reason why in some cases could be better to use a permutation technique instead of just a hashing function.

An alternative to hash functions are the Hash Message Authentication Codes or $HMACs$. In comparison to normal hash function, these have hashes seeded with secret data. Since an attacker does not have the key, it is not possible to compute the $HMAC$ by themselves, providing protection against dictionary attacks.

2.2.1.8 Random time shift

A random time shift [Mivule and Anderson, 2015] can also be considered a special case of permutation. Applied to time stamps, this method adds a random offset (i.e. seconds or milliseconds) to every record within the time stamp attribute. By doing it, an entire data set can be anonymized at once since all time stamps are shifted at once.

The duration and the chronological order of the events are preserved in this technique. However, in the presence of external knowledge about the network traffic this anonymization technique is easily reversible.

2.2.1.9 Time unit annihilation

Annihilating a time unit [Slagell et al., 2006] is combining the black marker technique with partitioning but for time and date fields. First, the values are dismantled into year, month, day, hour, minute, second or millisecond. After this step is possible to annihilate any of the fields replacing them with 0.

It is possible to remove the time information (hour, minute and second) and still have the date. The opposite can also happen: removing the date information (year, month and day) and hold the time information.

2.2.1.10 Suppression

As shown before in this document, suppression [Mivule and Anderson, 2015] works precisely as it name indicates: suppressing (deleting) sensitive fields from a data set at a cell level. There is not just one way to suppress a record: it is possible to delete the whole record (i.e. removing all the information of an attribute, either by removing it or replacing the field by zero or " $*$ ") it is possible to suppress part of an attribute (i.e. a zip code like 35684 can be suppressed as $35**$).

2.2.1.11 Generalization

Generalization [Mivule and Anderson, 2015] is a way of transforming a more sensitive or revealing attribute in a more general information. There are several ways of making that modification. In the case of ZIP codes for example, it is possible to group the specific numbers and group them as state or province, which is attributing a single value to a group of sensitive fields. Taking the case of gender, it can be generalized stating that it is a person instead of a male or female. Intervals can also be used, for example in the age or some other numerical field ("<= 25").

2.2.1.12 K-anonymity

First formulated by Latanya Sweeney [Sweeney, 2002b], the K-anonymity concept tries to answer the following: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful".[Sweeney, 2002b]

To answer that problem, K-anonymity provides data privacy by ensuring that the sensitive attributes (i.e. quasi-identifiers) are repeated k times, with K being always >1, in order to provide confidentiality and make more difficult the identification of individual values. To achieve k-anonymity, this algorithm relies of the combination of both generalization and suppression.

For example, one has age, state and religion as the quasi-identifiers of a given data set. It is said that a table has 4-anonymity when any combination of the quasi-identifiers occurs at least 4 times (a minimum of 4 rows of the privatized table have those exactly 3 identifiers: age, state and religion).

2.2.1.13 Data Swapping

Exchanging sensitive cell values with other cell values within the same attribute is the core concept of data swapping [Dalenius and Reiss, 1982]. This method is a data transformation technique where the underlying statistics and frequency of the data are preserved which turns to be a difficulty for an attacker to map the original values along with the anonymized values.

What needs to be considered in this case, is the selection of the swapping rate, the attributes to be swapped and the respective candidate data records for the data swapping.

2.2.1.14 MinGen Algorithm

The MinGen algorithm [Sweeney, 2002a] was developed having as a default condition to adhere to K-Anonymity with a minimal generalization. The author demonstrates how is it possible to have records adhere to K-Anonymity making use of generalization and suppression.

Following there is an explanation of the Preferred Minimal Generalization Algorithm (MinGen). Provides K-Anonymity with minimal distortion combining generalization and suppression.

Parameters

- Original Table (private data)
- Quasi-identifiers
- K-Anonymity constraint
- Domain Generalization Hierarchies

Steps

- Determine if the original table itself satisfies the K-requirement. If so, the table is the k-minimal distortion;
- Store all possible generalizations of the table, considering the quasi-identifiers;
- Store the generalizations that adhere to K-Anonymity;
- Store the k-minimal distortions (based on the Precision metric);
- A single k-minimal distortion table is returned;

In this case, is it possible to see the use of generalization with suppression. The difference is that in this case, parts of the attribute or the whole attribute is suppressed. As simple example is the case of ZIP codes: considering 35684 as the original number, it is possible to obtain different levels of suppression, for instance $3568*$, $356**$, until full suppression $(*****)$.

Following the above described, the concepts of Domain Generalization Hierarchy and Value Generalization Hierarchy take place. As illustrated below in Figure 2.1, there are levels going from 0 to 3 in this case, being 0 when the value remains unaltered and 3 the full suppression.

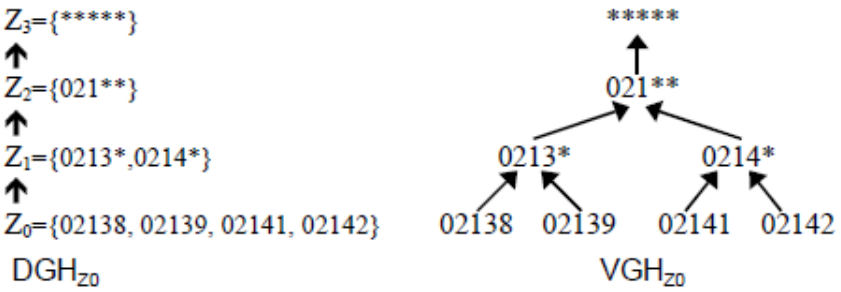


Figure 2.1: Domain and value generalization hierarchies [Sweeney, 2002a]

The Domain and Value Generalization Hierarchies show that is possible to have different levels of generalization and suppression, so several tables can be created with these modifications. The key point in the algorithm is guarantying that the tables being used are the ones with the least possible modifications, to give the most usability of the data assuring the privacy at the same time.

Two steps are necessary to assure the above described: verifying if the table generated adhere to K-Anonymity and comparing the precision metric of each new table of modified values. Assume that after generalizing a private table, three tables with different levels of generalization adhering to K-Anonymity can be used. The best one in terms of usability has to be chosen. This is where the precision metric is used. A value between 0 and 1, maximum distortion and minimum distortion, respectively. With this knowledge it is possible to select a table adhering to K-Anonymity and provide better usability (less distortion of data). It is possible to assign different weights to different attributes in the calculation of the precision.

The algorithm provides excellent theoretical results. However, with respect to complexity, it is not efficient. Real-world algorithms have different outcomes. Considering this fact, comparing MinGen with two real-world algorithms Datafly and μ-Argus (Section 2.4), reveals some differences. MinGen theoretically provides the best solution given that all possible generalizations are generated which is not efficient in real world systems. Even on modest sized tables, searching for all the possible generalizations is impractical. That is why the author refers Datafly and μ-Argus considering that they are real world systems and it is possible to compare the results of each one of them.

2.2.1.15 Differential privacy

The differential privacy [Mivule and Anderson, 2015] aims to provide means to maximize the accuracy of queries from a given data set, while minimizing the chances of identifying its records. There are similarities with the noise addition. In this method, anonymization is assured by the addition of Laplace noise to the queries performed on the data set. With this it is not possible to distinguish if a certain value was modified or not.

2.2.2 Unstructured Data

The following section analyzes the techniques and algorithms that allow the anonymization and concealing of unstructured data (i.e. text or audio).

2.2.2.1 DNA-inspired Information Concealing Algorithm

Inspired on the DNA structure of repeats, Lukas Kencl and Martin Loebl developed and presented an algorithm [Kencl and Loebl, 2010] about information concealing based on the introduction and maintenance of families of repeats.

Consider the information concealing problem where a certain sequence w and a small integer k are given and $|w|$ represents the length of the sequence. One wants to transform w into a new sequence wF so that is computationally hard to reconstruct w from wF; the length of wF is linear in $|w|$ and if s would be a segment of w, when $|s| <= k$, then s would be a segment of wF.

With a concealing problem, an attacker problem is inherent: how much information about the private sequence w can an attacker reconstruct from the final and concealed sequence wF.

In order to provide a solution to this problem, the authors developed an algorithm composed by five procedures. Before the application of the procedures, the input sequence is turned into a cyclic sequence and only then, the five procedures can begin to be applied. A cyclic sequence is a sequence that has the last item connected to the first.

Even though the first procedure has preparatory function, the five of them have a basic pattern: partition the input sequence into consecutive disjoint blocks; in front of each block, the terminal part of its predecessor is added (overlap); dust can be added in the end of each block; rearrange the blocks in to an output sequence wF. In this application dust is considered a random part of the sequence itself. With a length related with the length of the sequences to preserve (K).

The procedures are:

- S: Has a preparatory function, several runs of this procedure break the sequential order of the input sequence. Figure 2.2 contains an exemplification of the S procedure (for clarity purposes, the following example was taken from the DNA-inspired information concealing algorithm's paper [Kencl and Loebl, 2010]).

Example 1: Procedure S(ω, o, *lb*, *ub*)
Input ω = **theEaimEofEthisEpaperEisEtoEpresentEanEinformationEconcealingEalgorithm**, parameters $o = 3$, $lb = 4$, $ub = 6$.

1. First, the input sequence is partitioned randomly into blocks of length 4, 5 or 6. The blocks are divided by '+' below:
   theEai+mEofEt+hisE+pape+rEisE+toEpr+esent+EanEi+nfor+matio+nEco+ncea+lingE+algor+ithm+
2. Next we add overlap (of length $o = k - 1 = 3$) in front of each block:
   thmtheEai+EaimEofEt+fEthisE+isEpape+aperEisE+isEtoEpr+Epresent+entEanEi+nEinfor+formatio+tionEco+Econcea+cealingE+ngEalgor+gorithm+
3. Next we add the dust behind each block (of length approximately 2), and we get the cards:
   thmtheEaip+EaimEofEtim+fEthisEcon+isEpapeEin+aperEisEa+isEtoEproEp+Epresentese+entEanEilgo+nEinforiE+formatiofo+tionEcoE+EconceaEci+cealingEpa+ngEalgorEp+gorithmap+
4. Finally the output is given by arranging the cards in a random order (here we use the order 14, 9, 10, 13, 5, 3, 12, 1, 6, 4, 7, 11, 8, 15, 2):
   ngEalgorEpnEinforiEformatiofocealingEpaaperEisEafE thisEconEconceaEcithmtheEaipisEtoEproEpisEpapeEinE presentesetionEcoEentEanEilgogorithmapEaimEofEtim

Figure 2.2: Example of S procedure

- S1: Similar to S, however the overlap is the whole preceding block. Here, the cyclic order of the blocks is defined by a set of acceptable permutations between them. Figure 2.3 contains an exemplification of the S1 procedure (for clarity purposes, the example was taken from the DNA-inspired information concealing algorithm's article [Kencl and Loebl, 2010]).

- S1+: The purpose of this procedure is to generalize its output, preserving the overlaps (confusing to an attacker).

- S2: Assuming as an input the output of S1, here the blocks are partitioned into two blocks. The overlap in front of each block is added and the blocks are rearranged in a random order. As a last exemplification, Figure 2.4 contains the demonstration of the S2 procedure (for clarity purposes, the example was taken from the DNA-inspired information concealing algorithm's article [Kencl and Loebl, 2010]).

- S2+: Assuming as in input the output of S1+, S2+ is similar to S2. Here however, the type of partitioning of the blocks is different.

With the five procedures previously indicated, there two types of application of this algorithm: a weak concealing and strong concealing, depending on the type of the input. These two types of application vary on the number of repeats of S and which combination of procedures are applied.

The weak concealing is intended for non-specific inputs. For instance, where does not exist any outside knowledge about the likelihood or presence of some segments in the input. The strong concealing is intended for all other applications.

The algorithm described above is the one being evaluated in the experimental part of the this Dissertation.

Example 2: Procedure $S^1(\omega, lb, ub)$
Input ω = **ngEalgorEpnEinforiEformatiofocealingEpaaper EisEafEthisEconEconceaEcithmtheEaipisEtoEproEpisEpape EinEpresentesetionEcoEentEanEilgogorithmapEaimEofEtim**, parameters $lb = 6$ and $ub = 8$.
First, the input sequence is partitioned randomly into blocks of length 6, 7 or 8. The blocks are divided by '+' below:
ngEalgo+rEpnEinf+oriEfor+matiofo+cealing+Epaape+ rEisEa+fEthisEc+onEconce+aEcithm+theEaipi+sEtoEpro +EpisEp+apeEin+Epresent+esetionE+coEentE+anEilg+ ogorith+mapEai+mEofEtim+
Next we add overlap in front of each block. For procedure S^1 the overlap is always the whole preceding block. We get the following cards; to make the example easier to understand we indicate by '*' the division of each card into two blocks:
mEofEtim*ngEalgo+ngEalgo*rEpnEinf+rEpnEinf*oriEfor+ oriEfor*matiofo+matiofo*cealing+cealing*Epaape+ Epaape*rEisEa+rEisEa*fEthisEc+fEthisEc*onEconce+ onEconce*aEcithm+aEcithm*theEaipi+theEaipi*sEtoEpro+ sEtoEpro*EpisEp+EpisEp*apeEin+apeEin*Epresent+ Epresent*esetionE+esetionE*coEentE+coEentE*anEilg+ anEilg*ogorith+ogorith*mapEai+mapEai*mEofEtim+
Finally the output is given by rearranging the cards by an acceptable permutation, i.e. by a permutation whose corresponding bipartite graph consists of a lot of cycles. The smallest length of a cycle is 4. It is not difficult to see that the following permutation π creates nine 4-cycles and one 6-cycle. In the following description of π, the cycles are grouped together; for instance the first 4-cycle has edges $(v_1, u_{10}), (v_9, u_2), (v_1, u_2), (v_9, u_{10})$. The first two of them belong to perfect matching M_3, the last two belong to perfect matching M_2.
$[\pi(1) = 10, \pi(9) = 2]$; $[\pi(2) = 6, \pi(5) = 3]$; $[\pi(3) = 9, \pi(8) = 4]$; $[\pi(7) = 13, \pi(12) = 8]$; $[\pi(14) = 11, \pi(10) = 15]$; $[\pi(11) = 18, \pi(17) = 12]$; $[\pi(19) = 14, \pi(13) = 20]$; $[\pi(16) = 21, \pi(20) = 17]$; $[\pi(15) = 19, \pi(18) = 16]$; $[\pi(21) = 7, \pi(4) = 5, \pi(6) = 1]$.
Hence the final sequence (for ease of understanding we preserve the separation symbols '*', which in reality would not be present) is:
ngEalgo*rEpnEinfEpaape*rEisEaEpisEp*apeEintheEaipi* sEtoEprofEthisEc*onEconcerEpnEinf*oriEforonEconce* aEcithmEpresent*esetionEmEofEtim*ngEalgoaEcithm* theEaipianEilg*ogorithapeEin*Epresentogorith* mapEaicoEentE*anEilgesetionE*coEentEsEtoEpro* EpisEpmapEai*mEofEtimrEisEa*fEthisEcmatiofo* cealingoriEfor*matiofocealing*Epaape

Figure 2.3: Example of S1 procedure

Examlpe 3: Procedure $S^2(\omega, o)$
Input ω = **ngEalgo*rEpnEinfEpaape*rEisEaEpisEp* apeEintheEaipi*sEtoEprofEthisEc*onEconcerEpnEinf* oriEforonEconce*aEcithmEpresent*esetionEmEofEtim* ngEalgoaEcithm*theEaipianEilg*ogorithapeEin* Epresentogorith*mapEaicoEentE*anEilgesetionE* coEentEsEtoEpro*EpisEpmapEai*mEofEtimrEisEa* fEthisEcmatiofo*cealingoriEfor*matiofocealing*Epaape**, parameter $o = 3$.
A consistent partitioning into blocks is indicated below:
ngEa+lgo*rEpnEi+nfEpa+ape*rEis+EaEpis+Ep*ape+ EintheEa+ipi*sEtoE+profEthisE+c*onEco+ncerEpnEi+ nf*ori+EforonEco+nce*aEci+thmEpre+sent*eseti+ onEmEofE+tim*ngEa+lgoaEci+thm*theEa+ipianEi+ lg*ogori+thape+Ein*Epre+sentogori+th*mapE+aicoEe+ ntE*anEi+lgeseti+onE*coEe+ntEsEtoE+pro*Epis+EpmapE+ ai*mEofE+timrEis+Ea*fEthisE+cmati+ofo*ceal+ingori+ Efor*mati+ofoceal+ing*Epa+ape
Next we add overlap (of length o) in front of each block (and we delete the 'helpful symbol' *):
apengEa+gEalgorEpnEi+nEinfEpa+aperEis+EisEaEpis+ pisEpape+apeEintheEa+eEaipisEtoE+toEprofEthisE+ isEconEco+EconcerEpnEi+nEinfori+oriEforonEco+ EconceaEci+EcithmEpre+presenteseti+etionEmEofE+ ofEtimngEa+gEalgoaEci+EcithmtheEa+eEaipianEi+ nEilgogori+orithape+apeEinEpre+presentogori+ orithmapE+apEaicoEe+oEentEanEi+nEilgeseti+ etionEcoEe+oEentEsEtoE+toEproEpis+pisEpmapE+ apEaimEofE+ofEtimrEis+EisEafEthisE+isEcmati+ofoceal+ ealingori+oriEformati+atiofoceal+ealingEpa+Epaape
Finally we rearrange the cards in a random order. The resulting sequence is as follows:
apengEagEalgorEpnEinEinfEpaatiofocealaperEisEisEaE pispisEpapeEisEafEthisEapeEintheEanEilgesetieEapisEtoE toEprofEthisEisEconEcoEconcerEpnEiEpaapenEinforisE cmatiofocealealingorioriEforonEcopisEpmapEEconceaEciE cithmEprepresentesetiofEtimngEagEalgoaEcioriEformatiE cithmtheEaeEaipianEinEilgogoriorithapeapeEinE prepresentogoriorithmapEetionEmEofEapEaicoEeealingE paoEentEanEisetionEcoEeoEentEsEtoEtoEproEpisapEaimE ofEofEtimrEisisEcmatiofocealealingori

Figure 2.4: Example of S2 procedure

Figures 2.2, 2.3 and 2.4 are examples taken from the original paper [Kencl and Loebl, 2010].

2.2.2.2 Permutation

A Permutation [Mivule and Anderson, 2015], in the most common sense, is a one-to-one mapping of values. It is a direct substitution technique that replaces each value with some other value selected within a possible range, resulting in a unique anonymized value for each original value. This method is useful when is necessary to preserve the count or the order of the data sets, without preserving the information of the values themselves.

The selection function of the permutation is often random, although is not necessary. There are several variations of permutation functions with different characteristics like performance or guarantees of non-collision or security. Nevertheless, a general characteristic in all permutation functions should be the difficulty in reversing the it without knowing the parameters.

For instance, using a hash function as a selection function while anonymizing an IP address can be dangerous if the hash function is known. Given the small space of IPv4 addresses, if additional parameters are not applied, the hash function could be retrieved through brute force.

2.2.2.3 Noise addition

There are some anonymization methods like Noise Addition [Mivule and Anderson, 2015] or Multiplicative Noise that privatize data by adding or multiplying random values. In the case of Noise Addition, the random values, picked between the mean and standard deviation of the original data. They are then added to the sensitive attribute values, in the original data set, providing confidentiality.

2.2.2.4 Multiplicative noise

The Multiplicative Noise [Mivule and Anderson, 2015] is similar to the noise addition. However, instead of adding the values to the original data, the random values are typically picked between the mean and the variance and afterwards multiplied with the original data.

With a description of the anonymization techniques and algorithms, below follows a table (Table 2.2) with some properties of the previously described techniques and algorithms.

Techniques	**Destructive**	**Preserves Order**	**Preserves Structure**
Black Marker	Yes	No	Yes
Truncation	Yes	No	No
Enumeration	No	Yes	Yes
Partitioning	No	No	No
Precision Degradation	Yes	No	Yes
Prefix preserving	No	Yes	Yes
Hash	No	No	Yes
Random Time Shift	No	Yes	Yes
Time Unit Annihilation	Yes	No	Yes
Suppression	Yes	No	Yes
Generalization	No	No	No
K-Anonymity	No	No	Yes
Data Swapping	No	No	Yes
Differential Privacy	No	No	Yes
Permutation	No	Yes	Yes
Noise Addition	No	No	Yes
Multiplicative Noise	No	No	Yes
DNA-inspired concealing algorithm	No	No	No

Table 2.2: Properties of the anonymization and concealing techniques

Table 2.2 gathers some of the properties of the previously described techniques and algorithms. Most of the techniques are non-destructive, which means that they transform the data and do not suppress it. On the other hand, most of them do not preserve the order of the data but preserve the structure. For instance, one has an enumerated sequence where order is important. Most techniques break the ordering of sequences.

In the following section, Metrics (Section 2.3), there are some of the metrics used to quantify the data anonymized or concealed by the techniques described above.

2.3 Metrics

In this section there is a description of some of the main metrics used to quantify usability and anonymization or concealing. By analyzing these values, a data curator can evaluate if the data is privatized and/or useful for the user of the data in terms of usability.

2.3.1 Cosine Similarity

Similarity between two documents, or files, is a function of the angle between their vectors in the term vector space. The vectors, containing information, are used the calculate their inner product and squared lengths. [Singhal, 2001]

The function is the inner product of the vectors, divided by the product of their lengths results in a normalized value between zero and one, the Cosine Similarity. The files being compared have exactly the same information if the Cosine Similarity is one. On the other hand, the files are completely different if the Cosine Similarity is zero.

The Cosine Similarity [Singhal, 2001] can be defined as:

$$\cos(\theta) = \frac{f_1 \cdot f_2}{|f_1||f_2|} \ , \ \theta \in [0, 1]$$

The angle between the two vectors is represented by θ. The files being compared are represented by the vectors f_1 and f_2.

Consider the following example:

File	**Text**	**Terms**
f_1	bear bear deer	bear deer
f_2	goat deer goat lamb goat bear goat	bear deer goat lamb
f_3	coon wolf goat lion mule	coon goat lion mule wolf

Table 2.3: Three files with different text and its respective term identification

File \ Terms	bear	coon	deer	goat	lamb	lion	mule	wolf	**Length**
f_1	2	0	1	0	0	0	0	0	$\sqrt{5}$
f_2	1	0	1	4	1	0	0	0	$\sqrt{19}$
f_3	0	1	0	1	0	1	1	1	$\sqrt{5}$

Table 2.4: Term count and squared length

Considering for example f_1 and f_2. First, the squared length of the two files is necessary:

$$f_1 = (2^2 + 1^2)^{1/2} = \sqrt{5}$$
$$f_2 = (1^2 + 1^2 + 4^2 + 1^2)^{1/2} = \sqrt{19}$$

Internal product between the two vectors (f_1 and f_2):

$$f_1 \cdot f_2 = 2 * 1 + 1 * 1 = 3$$

With the above, it is possible to calculate the Cosine Similarity between f_1 and f_2:

$$\cos(\theta) = \frac{f_1 \cdot f_2}{|f_1||f_2|} = \frac{3}{\sqrt{5} * \sqrt{19}} = 0.31$$

2.3.2 The Shannon Entropy

The Shannon's Entropy [Shannon, 2001] is a way of measuring the amount of information in a particular block of information. It returns the amount of information based on the uncertainty or randomness of data.

Suppose there is a data set were the values contain belong just to one attribute or category, then the entropy in this case would be zero. In order to quantify the randomness of the data, probability is used. There is a possibility to normalize the entropy, where a range between 0 and 1 are the possible results. When the entropy is equal to 1, it means that all the probabilities of the values are equal (there is a great variety of distinct values) as closer it gets to 1, the bigger the uncertainty of the data set.

2.3.3 Correlation

The Pearson's Correlation Coefficient [Mivule and Anderson, 2015], or correlation metric, measures the level of a linear correlation between two data points, the original data set and the anonymized one. It measures, as well, the direction of the correlation, being positive or negative.

This method returns values between -1 and 1. The signal indicates the direction of the correlation, if it is positive or negative and the value indicates the strength of the relation.

When there is a correlation of 1 between the two data points, then it is perfect and positive; if the value is -1 then there is a perfect negative correlation between the two data points, it is also possible to say that if one increases, the other decreases; if the result is in the middle, zero, it means that the data points do not depend linearly on each other, however it is possible to exit a nonlinear dependency.

2.3.4 Descriptive Statistics

The utilization of descriptive statistics is a general but also effective way of getting to know the amount of privacy granted to data or the usability of it.

With this method, several measures can be taken to analyze the anonymized data: mean, standard deviation, variance, co variance, dispersion, etc., are some of the values that can be measured in order to quantify the distortion between anonymized and original data.

Like in most anonymization methods, there is a trade-off, here there is no exception. The bigger the difference between the original and private data, the bigger the anonymization is, with a trade-off of less usability; the same happens in opposite way, a smaller difference - a lower anonymization but more usability.

2.3.5 Euclidean Distance

The reader may ask why to use a metric that measures the distance between two points if the values needed are not distances. However, working with anonymization techniques it is possible to implement clustering for purposes of privatization performance, for instance k-means.

In this situation, the Euclidean Distance becomes handy in measuring the distances within the original and privatized cluster. Furthermore, with these results, it is possible to interpret how well did the anonymization went.

2.3.6 The Davies Bouldin Index

Like the Euclidean Distance, the Davies Bouldin Index [Davies and Bouldin, 1979] is used to evaluate how good the clustering of data is. There three main factors to have in account with this metric: the quantification of how good the clustering is the main one; furthermore, there is the distance between clusters and the distances within cluster which can be useful for further analysis. The index 0-1 is an indicative of how well the clustering performed, being closer to 0 a better clustering indicator.

2.3.7 Classification Error Metric

With the Classification Error Metric [Mivule and Anderson, 2015] there is a process similar to the descriptive statistics for example, i.e. measuring the classification error returned and compare it to the original data, being the trade-off between data privacy and usability present also.

In this case, both original data and anonymized data are passed through a machine learning algorithm which returns a classification error for original and anonymized data. The error classification is subtracted to the one resultant from the original data where the bigger the difference, the more privacy with less usability, and vice versa.

2.3.8 Mutual Information

When taking the Mutual Information (MI) [Kraskov et al., 2003] into consideration it is possible to observe the great utility it can provide. Using this metric, there are several ways to improve the assumptions taken of the privatized data and, with the same principle, provide better anonymizations when the MI is used in the anonymization algorithm.

For instance, the K-Anonymity algorithm with enhanced utility, uses the MI to improve its anonymizations by comparing several combination of attributes and choosing the best option based on the MI.

It is also possible to use the MI to formulate models in order to measure the risk and the information loss of a privatized data set.

To summarize, there are different metrics that can be used in the classification. However, they all have the purpose of providing a classification metric in order to analyze the privacy and usability of the data.

The next section, Tools that implement algorithms and techniques (Section 2.4), presents some of the main tools used in the field until the current date.

2.4 Tools that implement algorithms and techniques

After describing the algorithms and metrics in the previous sections, a research about the tools and frameworks available to implement those techniques and measure their output is presented.

The following sections present some of the main tools found and their respective characteristics. It is important to notice that some of them are already out of date. However, their description together with the up-to-date tools is for comparison purposes and chronological background.

The following tools are the ones that implement algorithms and techniques previously mentioned.

2.4.1 μ-Argus

Anco Hundepool and Leon Willenborg developed a tool which is still being used nowadays (2015). μ-Argus [Hundepool and Willenborg, 1996], a software program designed to create safe micro-data files.

In this tool, that implements a greedy algorithm [Sweeney, 2002a], the data curator specifies a value for 'k' and selects how sensitive is an attribute, within a value between 0 and 3, not identifying and identifying, respectively. The rare (unsafe) combinations are detected by testing 2 and 3 combinations of attributes. Generalization and cell suppression are used in order to eliminate unsafe combinations of attributes. As μ-Argus suppresses data at a cell level the output data usually contains all the tuples but with missing values in some cells.

As it tests 2 and 3 combinations the algorithm does not assure k-anonymity due to the possibility of existing 4 combinations that are unique thus may fail to provide adequate privacy. As the k-anonymity requirement is not enforced on suppressed values, it becomes vulnerable to linking and to inference attacks.

Over the time the algorithm/tool was being improved and developed in different platforms as C, C++, Visual Basic or R. It still has a wide use currently, being the last update dated from October of 2015.

2.4.2 TCPdprive

This is a light weight C program created by Greg Minshall at Ipsilon Networks in 1996 which anonymizes data by eliminating confidential information from packets traces collected from a network. It eliminates the sensitive information by replacing the sensitive fields with fabricated information which avoids the reconstruction of the sensitive information.

It works over *tcpdump* '-w' files. With different levels of anonymization, TCPdprive goes from level 0 to level 99. More or less secure outputs, being the 99 a level where the information is released as it is and 0 the most secure level.

The latest version is dated from 2005 with some limitations. For example, the system compatibility (SunOS, Solaris, and FreeBSD) or the non-preservation of subnet broadcast information.

2.4.3 Datafly

Developed by Latanya Sweeney in 1997, the Datafly algorithm [Sweeney, 1997], was created with the purpose of privatizing medical records. It works using generalization, suppression, inserting or removing information without losing the useful details contained in the data and guarantying K-Anonymity adherence.

There is no public implementation available at the time of this writing. Below it is described the algorithm used by this tool.

The algorithm starts by creating a frequency list with the distinct sequences of values of the private data. In the next step the attribute having the most distinct values is going to be generalized. Until there is a minimum of 'k' or fewer tuples with distinct sequences of values in the frequency list the generalization proceeds, making it a greedy algorithm. If is there some sequence of values occurring less than 'k' times, it is suppressed.

Since this algorithm can over distort the data it does not necessary provide k-minimal generalizations or distortions but the solution is always adhering to K-Anonymity. It also generalizes all the values associated with an attribute and suppresses all values within a tuple which makes it computationally efficient. The heuristic used to select which attribute to generalize is the verification of the attribute with the greater number of distinct values.

The latest version implementation publicly available is dated from 2012.

2.4.4 Ip2anonIP

Based in TCPdprive, Ip2anonIP is a simple filter for CSV files created by Dave Plonka in 2003. It is a tool that turns IP addresses into host names or anonymous IP addresses. This filter has the possibility of adding some arbitrary field rewrites. However, it can take hours to prepare a data set of one full day which is not good regarding the time consumption.

2.4.5 Crypto-PAn

At Georgia Tech College of Computing, Jinliang Fan, Jun Xu, Mostafa H. Ammar and Sue Moon (Sprint) a new prefix preserving anonymization tool was developed in C++.

Crypto-PAn [Fan et al., 2004] allows the data curators to anonymize network traces applying a prefix preserving anonymization technique. The tool was maintained updated until 2010 along with different versions and platforms.

As the its own name indicates, it is a cryptography based method, where the data curators provide the tool with a secret key. With the use of the same key consistency is achieved in multiple network traces meaning that the same IP address in different traces is anonymized with the same resultant IP address. As mentioned before, in the metrics section, the prefix preserving method allows to preserve subnet structures, let's say that if two private IP addresses have a prefix of 'n' bits, the anonymized IP's will have the same 'n' bits prefix.

Since this algorithm uses a bit wise anonymization, with all the privatized IP addresses being dependent on previous anonymizations (consistency), this leads to security flaw. As the anonymized IP's share a common prefix with the private addresses, if one is able to de-anonymize one IP, all the others with same prefix are affected. Nevertheless, when injection attacks are not likely to happen, this is a good option due to the maintenance of IP structures and possibility of anonymization across different locations (with the sharing of the key).

2.4.6 CANINE

When researchers concluded that there was a need to anonymize a wider variety of data logs, new solutions were proposed.

At NCSA[1] was introduced the Converter and ANonymizer for Investigating Netflow Events (CANINE) [Li et al., 2005], an anonymization tool aiming at the privatization and conversion of different NetFlow[2] formats.

Written in Java, with a Graphical User Interface (GUI), CANINE is user friendly, giving the possibility of clicking in the method to use, truncation, random permutation or prefix preserving.

2.4.7 SCRUB-PA

One year after, in 2005, the same authors of CANINE presented the SCRUB-PA (Process Accounting) with the intent of anonymizing Process Accounting logs. SCRUB-PA is one of the four parts of the SCRUB infrastructure. This infrastructure is composed by:

SCRUB-tcpdump: based and built on tcpdump, is designed to provide the application of multi-level anonymization to packet traces. Allowing the management of packet traces whilst protection sensitive information from being disclosed. [Yurcik et al., 2007]

[1]National Center for Supercomputing Applications

[2]NetFlow is a feature that was introduced on Cisco routers that provides the ability to collect IP network traffic as it enters or exits an interface

SCRUB-PA: as mentioned above and based of the Java code used in CANINE, this is intended for the anonymization of Process Accounting logs. [Luo et al., 2005]

SCRUB-NetFlows: a NetFlow anonymization tool. It fixes the flaws found in previous tools and uses several options to anonymize the fields of standard NewFlows. [Yurcik et al., 2008]

SCRUB-Alerts: built for anonymizing intrusion detection system alerts, for example firewall or virus alerts.

2.4.8 FLAIM

With time and changes in the requirements of data curators, the previous mentioned tools were becoming limited in terms of usage. In a way that they could not be called by a command line script. To overcome that barrier, a new modular command line UNIX tool was need.

In 2006, the LAIM (Log Anonymization and Information Management) Working Group at the National Center for Supercomputing Applications under the direction of Adam Slagell, developed a C++ solution called FLAIM [Slagell et al., 2006] to surpass the limitations of the previous.

This specific framework primes for being particularly modular and not bounded by specific logs being anonymized. As well as possibilities given regarding the anonymization level, where data curators or the system administrators can tune the trade-off between the information loss and the anonymization level. Since this framework includes several anonymization techniques (truncation, prefix-preserving, enumeration, etc.), it enhances the range of possible applications.

FLAIM was widely used and was getting support by the developers and community. However, the last updated version dates from 2014. Nowadays, more complete and up-to-date solutions are being used and developed (see further down).

2.4.9 IPanonymous

IPanonymous is a Perl port of Crypto-PAn created in the year of 2005. The tool provided the possibility of making a one to one mapping of the private IP address to the anonymized one, support for prefix-preserving, consistency across traces (over time and location) and a cryptography based anonymization.

The tool's logic is similar to the one used in Crypto-PAn and it is able to provide consistency in the process. Using the same key will guarantee consistent results with different implementations.

2.4.10 AnonTool

AnonTool [TMF, 2015] is an open-source tool developed in C in 2006 that provides easy, flexible and efficient functions that can be used to anonymize live traffic, or packet traces in the 'libpcap' file format.

It supports several formats as IP, TCP/UDP, HTTP, FTP and Netflow. One of the possible applications provides a basic anonymization functionality for the IP/TCP/UDP protocols and other applications can perform anonymizations on NetFlows.

Despite the community support (open-source), the last update that dates from August 2015 was released with the announcement of the end of support and development for this tool.

2.4.11 Open Anonymizer

Open Anonymizer [SourceForge, 2015] is a Java developed tool introduced in 2011 and supported until 2013 that protects sensitive data with generalization. This feature, based on the concept of k-anonymity and l-diversity, allows the creation of data twins that mask the identity of individuals.

The following section presents tool that also implement metrics.

2.5 Tool that also include metrics

The previous section presented tools that implement a specific algorithm or algorithms. This section presents tools that in addition the algorithm's implementation, also implement metrics.

2.5.1 ARX Data Anonymization Tool

ARX Data Anonymization Tool [Prasser and Kohlmayer, 2015] was introduced in 2012 by researchers of the Technical University of Munich and it has an up-to-date support (last from November 2015). This tool is very complete due the wide range of algorithms implemented.

The scalability feature was considered since its early development. Capable of analyzing data utility and re-identification risks, it supports privacy models, such as k-anonymity, l-diversity and t-closeness. Semantic privacy models as differential privacy. Data transformation techniques like generalization, suppression and top / bottom coding as well as global and local recoding.

Being an open source tool it has a great support by the community. Developed in Java, has a graphic interface, it is capable of handling large amounts of data files and it is platform independent, running on Windows, Mac and Linux.

2.5.2 SECRETA

Released in 2014, SECRETA [Poulis et al., 2014] presents itself is a system for evaluating efficiency and effectiveness of anonymization algorithms. With a possibility of choosing which algorithms to evaluate, the analysis is make in an interactive and progressive way. The results are displayed with the attribute statistics and several indicators of the data utility, in a summarized and graphical form.

The system is implemented in C++. Having different modes of operation, this tool invokes one or more instances of the anonymization module with the specified algorithm and parameters. The anonymization results are collected by the Evaluator method and forwarded to the Experimentation module. From there, the results are then forwarded to the Plotting module, for the graphical visualization, and/or to the Data Export module, for data export.

2.5.3 PARAT

As an example, PARAT is being described also for being a commercial solution in the privacy and anonymization area of business. Owned by Privacy Analytics, offers a solution similar to the ones referred before.

Different anonymization algorithms, risk and data utility metrics, but focusing on medical data and, in a professional and commercial point of view.

On the company online page, it is possible to register and have a two-hour worth of demonstration of the tool, through a Windows virtual machine.

2.5.4 sdcMicro

sdcMicro[Matthias et al., 2015] - Statistical Disclosure Control Methods for Anonymization of Microdata and Risk Estimation, is an up-to-date 'R' package used for the generation of anonymized data. In addition, it also includes metrics and estimation processes, which provides a better and more complete analyses of the data.

Fitted with a graphical user interface, there is a wide variety of available techniques to apply in the anonymization process. If compared with μ-Argus, this has all the techniques plus a few more.

There is still the possibility of usage in batch-mode from other software, the flexibility of the functions and there is no need to spend time managing meta-data. In sum, this is also a very complete tool there is available to anyone without cost and with a short learning curve.

There are others tools available, for example: TIAMAT, UTD Anonymization Toolbox, Cornell Anonymization Toolkit or LOBSTER. However, the focus are the ones that are currently being supported or developed and more up-to-date.

The table that follows, Table 2.5, presents a summary of the previously mentioned tools, their main characteristics and indication of the tools still being maintained and supported.

Anonymization Tools	**Metrics**	**Target**	**Latest Version**
Ip2anonIP	No	IP (.csv)	2003
TCPdprive	No	tcpdump	2005
CANINE	No	Netflow	2005
SCRUB-PA	No	P. A. Logs	2005
IPanonymous	No	Netflow	2005
Crypto-PAn	No	IP	2010
Datafly	No	Tables	2012
FLAIM	No	Generic	2014
AnonTool	No	Network traces	2015
μ-Argus	No	Tables	2015
TIAMAT	Yes	Anon. Tecn.	2009
UTD Anonymization Toolbox	Yes	Generic	2012
Cornell Anonymization Toolkit	Yes	Generic	2014
SECRETA	Yes	Anon. Tecn.	2015
PARAT	Yes	Generic	2015
sdcMicro	Yes	Generic	2015
ARX Data Anonymization Tool	Yes	Generic	2015

Table 2.5: Characteristics of the anonymization tools

The most complete and up to date anonymization tools are placed in the bottom of the table. Including a wide range of algorithms and metrics while still being supported and maintained.

For the experimental stage, the DNA-inspired information concealing algorithm will be used. It is not present in Table 2.5 because it is not available any public version of the algorithm. The code being used for this work is a Matlab implementation provided by the authors.

The next chapter, Research Approach (Chapter 3), describes the approach in this context and the methodology.

CHAPTER 3

Research Approach

In this chapter there is a detailed explanation of the approach followed in order to achieve the proposed objectives.

To have a proper experimental environment, valid results and conclusions it is necessary to define and organize each step of the experimental research.

The following sections identify and describe each part of the research experiment: context and objectives, methodology, experiment setup, tools and scripts used and experiment data files.

3.1 Context and Objectives

In this experiment the DNA-inspired concealing algorithm will be used to anonymize certain data sets. An evaluation of how well the data is concealed and kept private will also be made. The performance will be also a point of analyses measuring different factors for that purpose. The result will provide a guideline about the usage and applicability of the algorithm.

It will happen in a controlled environment set up before the experimental stage take place. The same machine for the entire set of experiments and the same conditions will be used.

The objective of this work is to test the applicability of the referred method taking into consideration the privacy and confidentiality protection achieved. Furthermore, performing an evaluation of the performance of the algorithm and its properties are made and conclusions about its applicability in the cloud will be drawn.

As mentioned before, the main goal of this research is to test the applicability of the DNA-inspired information concealing algorithm over different data sets and evaluate it.

Evaluating the performance of an algorithm requires the evaluation of several parameters, collection of data and taking measures. To be able to test the applicability of the DNA-inspired information concealing algorithm it is necessary to collect and measure the details of the concealing process and analyze the data generated. To analyze the data being generated, there are metrics as similarity between original e concealed files, execution times and spatial complexity that need to be determined and measured.

This is the first stage where the metrics referred in the State of the Art will be used. With the support of the metrics described, the similarity of the data concealed by the algorithm will be measured and determined. This will provide concrete numbers and statistics that will be essential for the evaluation and analyses of the results. During the experimental stage there will be changed various parameters of the algorithms. This will allow an analysis of the results based on different scenarios.

Considering the above described, the approach is to run the DNA-inspired information concealing algorithm with different data sets. The second stage of the research is to focus on the performance evaluation. Each execution time is recorded and then an average is determined. This, align with an analysis of the size of the files generated, will allow to determine how fast the algorithm is and its spatial complexity - size of the concealed data.

After collecting and analyzing the results it will be possible to evaluate its performance and how good is the privacy and confidentially protection provided by the algorithm.

With the experimental part finished, an analyses of the data collected will take place. Along with it, it will be possible to present the results of experiments that not have been made yet. Achieving, in this way, new conclusions about the applicability of the DNA-inspired information concealing algorithm in this kind of data. Furthermore, discussing its possible application for data privacy protection in the cloud.

3.2 Methodology

To perform the experiment several runs of each test were executed. For each batch of runs, there was a variation of the parameters used in the input of the algorithm.

Each time the algorithm was executed with a certain set of parameters, a different seed (used on the cut of the blocks) was used. The reason was to provide different outputs in every execution. In total, each set of parameters was executed five times in order to provide consistent results. For a posterior analysis of the results, averages and standard deviations are calculated over each set of executions.

To run the experiments, a bash script was prepared. In that way, the Matlab instances were called (without graphical user interface) automatically. The script was changing the parameters described below, as well as measure the execution times.

Usually, the anonymization and concealing methods only support one file type: text (with different formats). However, the DNA-inspired information concealing algorithm supports text and audio. Thereby, two different analyses will be made.

The methodology for this work has two approaches, one for each file type - text and audio. The following sections describe the approach followed in the experiments with text and audio files.

3.2.1 Approach for Text Files

As it was described before, there are parameters that can be specified to run the algorithm. Besides K (length of sequence to preserve) there is the possibility of choosing the block size - B - (length of blocs to conceal) and the concealing type: weak or strong.

Regarding the values used in the text experiments, below there is a description of the values used in those parameters and how they were modified.

- Block Size - B

  Length of the blocks used by the algorithm. As described earlier, the algorithm cuts the input into blocks and performs the needed operations over those blocks. The B parameter varied between 64, 128, 256, 512, 1024 and 2048. The block size parameter is presented as B64, B128, B256, B512, B1024 and B2048.

  Example: Block size 64 - 'The number of characters or letters in a text like this is 64...' Block size 64 - 64 samples read from the wav file

- Length of sequences to preserve from input - K

  Length of sequences to be preserved from the input file. Characters in case of text, samples in case of audio. It preserves fixed length sequences of the input file. The K parameter varied between 3, 4, 5, 7 and 10. The K parameter is presented as K3, K4, K5, K7 and K10.

  Example of text input: K3 - 'The', ' nu', "mbe', 'r o', 'f c', 'har', 'act', 'ers', ' or', 'let', 'ter', 's i', 'n a', ' te', 'xt ', 'lik', 'e t', 'his', ' is', ' 64', '...'

- Concealing type

  Since the algorithm provides two types of anonymization, the two types were used in the experiments.

  Weak: Applies the transformations and operations to the input without adding dust and overlaps.

  Strong: To provide a stronger concealing and to prevent and difficult even more de-concealing attempts, this type adds dust and overlaps of the information based on the input itself. As well as different characteristics for the cuts and permutations performed.

To make the analysis of the concealed files that were generated, the internal product and length (sum of the squared frequencies) of the original and concealed files are calculated. The calculation allows the determination of the cosine similarity.

The cosine similarity measure was taken in four different ways, for all the files. Below there is a description of those measures of similarity.

For all cases, all the characters were considered. It was assumed that all the characters matter for the analysis. Therefore, there is no striping of spaces, punctuation marks or other characters.

- Fixed length sequences

  One of the characteristics of the English language is its word length. It was shown that the English language has an average length of 5.1 letters per word [Bochkarev et al., 2012]. Considering the average length, one way of analyzing the data is by fixing a word length and verify all the sequences of characters with that specified length.

  To consider the above described, a fixed length of five characters was defined. All the sequences of characters in the original and concealed documents are identified and represent a term. Additionally, for comparison purposes, measures with lengths of 25 and 50 characters were also taken.

- Term Frequency

  The notion of Term Frequency is used to refer all the sequences of characters separated by a space. With the term frequency analysis, all the sequences of characters separated by space are identified and represent a term.

- Two consecutive terms

  Similar to the definition above, with the exception that each term is separated by a space. All the sequences of characters separated by one space, represent one term in this case.

- Three consecutive terms

  Analogously, all the sequences of characters separated by two spaces, represent one term in this case. This case, as well as the previous one, allow the analysis of how different the term construction and precedent is on both of the files being compared.

To measure the cosine similarity in such a way, several Python programs had to be developed (Section 3.3.2 - Tools and developed scripts). The developed programs started by reading both original and concealed files, in the four ways described above. After that, for each one of the four ways of analysis, the cosine similarity was implemented manually.

Later, after all the calculations and file analysis, the results were gathered and treated by the author for validation and respective analysis.

3.2.2 Approach for Audio Files

Regarding the audio files, the parameters had to be different due to the nature and characteristics of the inputs. All the audio files used in the experiments had a sample rate of 44100 samples per second.

The reason to choose this particular value for the sample rate is due to its proliferation. This is the most commonly used sample rate. For instance, CDs use it. By having a 44100 sample rate, a 20 kHz maximum frequency is achieved. Which is generally the highest frequency audible by humans, so it makes sense to use this rate.

If compared with text, that would mean 44100 characters. For this reason, the values of K and Block Size, are going to be higher. To conceal the audio files, the following parameters were used on the algorithm:

- Block Size - B

  The B parameter varied between 2048, 20480, 204800 and 2048000. The block size parameter is presented as B2048, B20480, B204800 and B2048000.

- Length of sequences to preserve from input - K

  The K parameter varied between 10, 100, 1000 and 10000. The K parameter is presented as K10, K100, K1000 and K10000.

- Concealing type

  Both Weak and Strong concealing types were applied in the audio experiments.

The cosine similarity measure was taken in two different ways in this case. Following, there is a description of those measures of similarity: sample frequency and fixed length sequences.

The notion of sample frequency is used to refer all the samples of audio. With the sample frequency analysis, it is considered that each second of the audio file has 44100 samples. Due to the high value, this analysis is made for comparison purposes. It would not be feasible to compare audio files width such a high detail. The scope needs to be reduced by using sequence of samples to compare. For instance, comparing sequences of 10, 100, 1000 or 10000 samples.

Audio and text files change considerable in terms of file characteristics. In a text file, a single word, or sequence of characters has much more meaning than a single sample of the audio signal - each second has 44100 samples. For this reason, fixed length sequences should be used for the audio comparison.

It is fairly easy for a human to identify audio segments with, at least, approximately half a second or less of duration. However, for a computer, the analysis is more complex. Several segments of audio can be compared or mined.

However, due to power limitations of the machine being used for the experiments, as well as time, the audio analysis will follow a different approach.

In a first stage, the concealed files will be compared with the original, using all the samples individually. Additionally, sequences of 10 and 100 samples will also be analyzed, by the computer, verifying the similarity with the original files. Beyond that point, it is not possible, in this work, to continue the analysis with longer sequences.

On a second stage, it will be performed an analysis by the author. By listening the original and concealed files with different values of K, the author will realize when is it possible to start to recognize the concealed original files.

3.3 Experimental Setup

To perform the experiments there was a need to develop certain scripts and to use specific tools. The following sections describe the characteristics of the machine the was used for the experiments, as well as the tools and scripts used and developed. The last section describes the data used in the text and audio experiments.

3.3.1 Machine characteristics

The experiments took place in a personal computer prepared for the tests. The environment was the similar for all the runs of the experiments. Meaning that the resources available remained constant. Installed and running applications remained the equal for the duration of the experiments. The characteristics of the system are the following:

- Brand and model: Apple - MacBook Pro (Mid 2015)
- Operating System: Mac OS X El Capitan, version 10.11.5;
- Processor - 2,2 GHz Intel Core i7 (quad-core);
- Memory: 16 GB, 1600 MHz, DDR3;
- Graphics: Intel Iris Pro 1536 MB;
- Storage: 256 GB Flash - PCIe

3.3.2 Tools and developed scripts

The support the experiments, the following tools were essential. The main tool that was used for the implementation of the algorithm itself was Matlab version R2015a (8.5.0.197613) 64-bit.

For the bash scripts used for running the algorithm it was used the built in GNU bash, version 3.2.57(1)-release (x86 64-apple-darwin15). Regarding the analysis of the results and implementation of the cosine similarity metric, the used tool was Python version 2.7.10. To call and execute the scripts, it was used the built in Terminal - version 2.6.1 (361.1).

Like it is explained in the Work Plan (Appendix A), initially, Matlab was used for the calculations of results. However, with the advancements of the work and the use of cosine similarity, this metric had to be implemented manually.

The author implemented the cosine similarity metric recurring to Python. After developing the necessary scripts, it was also necessary to adapt the metric for the context of the work. Meaning that different Python scripts had to be prepared in order to get measures in terms of sequences of characters with fixed lengths, term frequency or two and three consecutive terms. The loading of text and audio files in accordance with the needed data specifications had to be done.

Additionally, others scripts were developed. Retrieving the values from all the measures of the evaluation. Calculating averages and deviations, counting terms, plotting graphics and others.

Regarding the bash scripts, they had the main function of making Matlab calls of the algorithm for application over the files with different parameters (K or block size). The execution times were also measured recurring to bash functions.

3.3.3 Data

The type of data used in the experiments were text (.txt) and audio (.wav) files. Several text files as well as audio files were used in the experiments. Different characteristics allow a more complete evaluation of the concealing performance. The original characteristics of the files are indicated in tables 3.1 and 3.2, for text and audio, respectively.

It was important to have different text files available for the experiments. The different files allow a more complete analysis in cases were different contents, authors or types of texts are being concealed. In these files there are novels, formal text or simple email exchanges. There are different styles of writing (different authors) and also works from the same authors, in order to analyze the differences in these cases.

Other than what is described above, there were no other specific reasons to choose the files presented in Table 3.1.

File	Number of Terms	File Size
Ulysses	264961	1,5 MB
Dubliners	67555	369 KB
A Portrait of the Artist as a Young Man	84868	464 KB
Hamlet	31997	175 KB
Macbeth	18086	100 KB
All's Well That Ends Well	24233	130 KB
Language an Introduction to the Study of Speech	80834	503 KB
Are Planets Inhabited	43445	254 KB
European Union Treaty	65478	398 KB
Emails (set 1)	4374	27 KB
Emails2 (set 2)	7166	44 KB

Table 3.1: Characteristics of the text files, in English language

Regarding the audio files presented above (Table 3.2), the reasons to choose these files were to have different kinds of audio files. To have files with voice recordings, instrumental or music in order to analyze the results with different data files and types of audio. Analogously, the reason to choose those specific files is not more than what is described above.

File	Duration	File Size
Man's World (James Brown)	02:37 min	13,8 MB
I Got You (James Brown)	03:36 min	19,1 MB
Get Up Offa That Thing (James B.)	04:10 min	22 MB
Let's Dance (David Bowie)	04:16 min	25,5 MB
Starman (David Bowie)	04:02 min	21,3 MB
Life on Mars (David Bowie)	04:18 min	22,7 MB
Interstellar (Hans Zimmer)	05:46 min	30,5 MB
Inception (Hans Zimmer)	04:47 min	25,3 MB
Author self-recording 1 (SR1)	00:16 min	1,4 MB
Author self-recording 2 (SR2)	00:16 min	1,4 MB

Table 3.2: Characteristics of the mono audio files

The next Chapter, Experimental Results (Chapter 4), presents the results obtained with the experiments and the respective analysis.

CHAPTER 4

Experimental Results

After gathering all the results, calculated the averages of the batch of executions and organizing the results in a structured way, the data can be analyzed.

Like described before, two data types were used: text and audio. The file extensions are *txt* and *wav*, respectively.

The text files consist of different documents. Different types of writing, different authors and different types of texts (for instance: novels, scientific or email exchange. The audio files consist of self-recordings of voice, music with vocals or instrumental.

In this chapter there is an analysis the experiments made with text and audio files. Followed by a discussion regarding the results obtained, advantages and disadvantages of the algorithm as well as suggestions for improvements of the algorithm.

4.1 Text Files

For the tests performed with text as input, several files with different characteristics were used. The analysis and results presented over next section contain what was obtained by the application of the DNA-inspired information concealing algorithm on text files.

4.1.1 Similarity of non-concealed files

In order to understand what the obtained measures represent, it is necessary to have a reference. To achieve such a reference, different texts can be used and compared to each other. Texts with similar lengths, but with different contents.

By measuring the cosine similarity of the randomly chosen texts, an indicative value is provided. In this way it is possible to know the expected values of similarity between two documents that are not related.

In order to do the above, several files were randomly chosen. In Table 4.1 there is description of the values obtained by measuring the Cosine Similarity between the novels of Sue Leather (The Amsterdam Connection) and Richard MacAndrew (The University Murders). As well as the comparison values between different types of writing: an Interview with William Gibson by Giuseppe Salza and Wireless Transmission of Power Resonating Planet Earth by Toby Grotz.

Type of Similarity	**Novels**	**Scientific vs novel**
Fixed Length - 3 characters	0,914	0,851
Fixed Length - 4 characters	0,834	0,734
Fixed Length - 5 characters	0,745	0,590
Fixed Length - 6 characters	0,579	0,336
Fixed Length - 7 characters	0,462	0,206
Average of Fixed Length Sequences	0,707 ± 0,185	0,543 ± 0,270
Term Frequency	0,847	0,846
Two Consecutive Terms	0,534	0,263
Three Consecutive Terms	0,083	0,005

Table 4.1: Reference values with between novels and different types of writing

Since the English language has an average word length of five characters, the experiments will be, mainly, done with five characters - for sequences of characters. However, for some cases like table 4.1, a fixed length of 3,4,6 and 7 is also applied. The goal is to observe the tendency of similarity variations in the cases of lower and higher sequences of characters.

The measures presented in Table 4.1 show that with a fixed length of 6 and 7, there is a notorious decrease of similarity (considering the length of 5 characters as the main value). When measuring the similarity with lengths of 3 and 4 characters, the opposite happens. There is also a high increase of similarity. The reason is that shorter terms are more frequent in texts (for instance: "the" or "you").

Usually, the term frequency values are the highest. As it is possible to observe in Table 4.1 (and in the results presented in the following sections) that the values are higher than $0,80$, which indicates a high similarity when comparing the terms separated by space. This is understandable. Words used in a novel or a scientific text are mainly the same, with a small group of different terms or expressions.

On the other hand, in case of an analysis of pairs of words (i.e. two consecutive words) or even triplets - three consecutive words - there is a higher difference. In this case it is possible to differentiate the files based on the values of similarity.

For instance, comparing novels, the similarity values of two and three consecutive terms are 0,534 and 0,083, respectively. However, if it is a novel against a scientific paper, the values reduce to 0,262 and 0,005, respectively. This is due to the type of writing. Texts of different nature, or even authors, tend to have a different way of word construction. That is way the values of similarity are usually lower. Being even lower with texts of different origins or characteristics.

In order to better understand how different works from the same author are related, Tables 4.2 and 4.3 present the measures of similarity between the novels of William Shakespeare and James Joyce, respectively.

The similarity is measured with the Cosine Similarity applied with a fixed length of five characters, term frequency, two and three consecutive characters.

File	**Fixed Length (5 characters)**	**Term Frequency**	**Two Consecutive Terms**	**Three Consecutive Terms**
All's Well ... / Hamlet	0,764	0,925	0,480	0,037
All's Well ... / Macbeth	0,724	0,902	0,409	0,021
Hamlet / Macbeth	0,720	0,921	0,378	0,022
Average	0,736 ± 0,024	0,916 ± 0,012	0,422 ± 0,052	0,027 ± 0,009

Table 4.2: Cosine Similarity between William Shakespeare's novels

File	**Fixed Length (5 characters)**	**Term Frequency**	**Two Consecutive Terms**	**Three Consecutive Terms**
A Portrait of ... / Ulysses	0,957	0,979	0,889	0,157
A Portrait of ... / Dubliners	0,938	0,972	0,857	0,181
Ulysses / Dubliners	0,929	0,964	0,831	0,146
Average	0,941 ± 0,014	0,972 ± 0,008	0,859 ± 0,030	0,161 ± 0,018

Table 4.3: Cosine Similarity between James Joyce's novels

File	**Fixed Length (5 characters)**	**Term Frequency**	**Two Consecutive Terms**	**Three Consecutive Terms**
Ulysses / Hamlet	0,718	0,862	0,454	0,027
Dubliners / All's Well ...	0,652	0,781	0,346	0,016
A Portrait of ... / Macbeth	0,688	0,844	0,315	0,013
Average	0,686 ± 0,033	0,829 ± 0,043	0,372 ± 0,073	0,019 ± 0,007

Table 4.4: Cosine Similarity between works of different authors - James Joyce and William Shakespeare

It is possible to observe a great value of term frequency (words). Meaning that even if it is a different work, the fact of being created by the same author, results in a high similarity in terms of words being used. The tendency of having lower values of similarity with two and three consecutive terms is also verified. Although, with James Joyce, the results indicate that the author connects more pairs and triples of words more frequently.

Focusing now in table 4.4, for a comparison of works from different authors, is it noticeable a decrease of similarity. It is natural since not only the content is different, but the origin is also different. Differentiating it even more.

The similarity between authors and different types of work is important, but it is not the only. It is important also to understand how the size of the input can after the similarity of the files. The next section presents an analysis of the size influence of the texts.

4.1.2 Length influence

The above section demonstrates how similar are two files that are not related in content but related with authorship or type of text.

To verify how the size of the file (i.e. length of the text - amount of terms) influences the similarity of the documents, several tests were performed in order to analyze files with different lengths. The original files were cut into smaller files. Starting with 1250 terms, the comparison is made with files of the same and different lengths. The reason for being 1250 terms, is to be significantly less than the original file size and to be able to notice a difference in similarity according to size of the files.

The following tables (4.5, 4.6, 4.7 and 4.8) contain with the results from the analysis with fixed length with five characters, term frequency, two and three consecutive terms, respectively.

Terms	**Ulysses vs Dubliners**	**Hamlet vs Macbeth**	**Ulysses vs Hamlet**
1250 / 1250	0,487	0,351	0,357
1250 / 2500	0,559	0,428	0,430
1250 / 5000	0,593	0,468	0,454
1250 / 10000	0,605	0,484	0,446
1250 / 20000	0,624	-	0,474
1250 / 40000	0,643	-	-
2500 / 2500	0,624	0,525	0,491
2500 / 5000	0,662	0,574	0,520
2500 / 10000	0,677	0,594	0,509
2500 / 20000	0,694	-	0,543
2500 / 40000	0,714	-	-
5000 / 5000	0,682	0,601	0,524
5000 / 10000	0,700	0,624	0,514
5000 / 20000	0,713	-	0,547
5000 / 40000	0,733	-	-
10000 / 10000	0,750	0,623	0,552
10000 / 20000	0,765	-	0,585
10000 / 40000	0,786	-	-
20000 / 20000	0,833	-	0,673
20000 / 40000	0,852	-	-
40000 / 40000	0,884	-	-
Complete file	0,929	0,720	0,718

Table 4.5: Fixed Length (5 characters) - Cosine Similarity with variation of the file length (number of terms)

It is possible to observe that in the case of a five character fixed length (Table 4.5) the value of similarity of considerably smaller versions of the files is close to 50% less, when compared to the full length version. It is also noticeable that as length of the file grows, the similarity values growth accordingly. Both for files from the same author as different authors.

Looking at the fourth column, there is yet another confirmation of what was previously observed. Comparing different authors, the similarity is lower.

Terms	**Ulysses vs Dubliners**	**Hamlet vs Macbeth**	**Ulysses vs Hamlet**
1250 / 1250	0,744	0,708	0,698
1250 / 2500	0,803	0,781	0,732
1250 / 5000	0,824	0,798	0,741
1250 / 10000	0,821	0,807	0,720
1250 / 20000	0,855	-	0,745
1250 / 40000	0,875	-	-
2500 / 2500	0,852	0,855	0,781
2500 / 5000	0,873	0,857	0,795
2500 / 10000	0,867	0,868	0,773
2500 / 20000	0,888	-	0,800
2500 / 40000	0,904	-	-
5000 / 5000	0,887	0,876	0,806
5000 / 10000	0,886	0,885	0,784
5000 / 20000	0,904	-	0,812
5000 / 40000	0,918	-	-
10000 / 10000	0,894	0,876	0,798
10000 / 20000	0,918	-	0,825
10000 / 40000	0,932	-	-
20000 / 20000	0,928	-	0,837
20000 / 40000	0,941	-	-
40000 / 40000	0,948	-	-
Complete file	0,964	0,921	0,862

Table 4.6: Term Frequency - Cosine Similarity with variation of the file length (number of terms)

Analyzing the term frequency, there are different characteristics. The values presented in Table 4.6 show that instead of 50% less similarity - like in the fixed length case - there is, on average, 25% less. Once again verifying that analyzing term frequency, the values of similarity will be higher most of the times and the growth comes along with the increasing of the length of the files.

Since in this case the analysis is made of terms and not sequences of characters, one possible application for this type of comparison, could be for example language identification.

Comparing with files with different alphabets (languages), could be possible to identify (if the similarity is high) the language of the files being analyzed, without disclosing the content.

Terms	**Ulysses vs Dubliners**	**Hamlet vs Macbeth**	**Ulysses vs Hamlet**
1250 / 1250	0,089	0,044	0,006
1250 / 2500	0,120	0,079	0,080
1250 / 5000	0,153	0,103	0,091
1250 / 10000	0,179	0,130	0,095
1250 / 20000	0,199	-	0,121
1250 / 40000	0,222	-	-
2500 / 2500	0,197	0,096	0,112
2500 / 5000	0,248	0,126	0,137
2500 / 10000	0,280	0,169	0,142
2500 / 20000	0,309	-	0,181
2500 / 40000	0,332	-	-
5000 / 5000	0,305	0,152	0,166
5000 / 10000	0,350	0,209	0,175
5000 / 20000	0,385	-	0,221
5000 / 40000	0,413	-	-
10000 / 10000	0,434	0,234	0,222
10000 / 20000	0,484	-	0,278
10000 / 40000	0,514	-	-
20000 / 20000	0,579	-	0,338
20000 / 40000	0,612	-	-
40000 / 40000	0,679	-	-
Complete file	0,831	0,378	0,454

Table 4.7: Two Consecutive Terms - Cosine Similarity with variation of the file length (number of terms)

In the case of two and three consecutive words, the results are different. Since the sequence of words matters, the longer the text, the higher the similarity. It is comprehensible that for instance the complete files have a value of 0,831 in similarity and 0,089 when comparing the smaller versions (Table 4.7). Close to 90% less.

Despite the bigger gap in similarity values, the general characteristics of having increased similarity with increased lengths, is maintained.

Terms	**Ulysses vs Dubliners**	**Hamlet vs Macbeth**	**Ulysses vs Hamlet**
1250 / 1250	0,0007	0,0016	0,0008
1250 / 2500	0,0027	0,0022	0,0006
1250 / 5000	0,0030	0,0020	0,0008
1250 / 10000	0,0072	0,0022	0,0011
1250 / 20000	0,0092	-	0,0022
1250 / 40000	0,0123	-	-
2500 / 2500	0,0039	0,0027	0,0004
2500 / 5000	0,0062	0,0022	0,0006
2500 / 10000	0,0114	0,0035	0,0019
2500 / 20000	0,0127	-	0,0039
2500 / 40000	0,0184	-	-
5000 / 5000	0,0089	0,0039	0,0008
5000 / 10000	0,0143	0,0062	0,0024
5000 / 20000	0,0175	-	0,0044
5000 / 40000	0,0281	-	-
10000 / 10000	0,0151	0,0084	0,0040
10000 / 20000	0,0203	-	0,0060
10000 / 40000	0,0305	-	-
20000 / 20000	0,0219	-	0,0071
20000 / 40000	0,0318	-	-
40000 / 40000	0,0493	-	-
Complete file	0,1463	0,0222	0,0268

Table 4.8: Three Consecutive Terms - Cosine Similarity with variation of the file length (number of terms)

For last, the analysis of three consecutive terms (Table 4.8). The output is similar to the previous. However, the values are significantly lower. Due to the lack of a high amount of three term sequences present in both files, the values of similarity will be generally low in this kind of analysis.

The execution of the experiments presented above, demonstrated two facts. The first is that the size of the file matters. Since a smaller file has a smaller distribution, the impact of the comparison of smaller files is higher. The second is that, the larger a file is, the higher are the chances of having more common elements, increasing the similarities.

With the reference values, it is possible to make the analysis of the concealing algorithm itself. The following section presents results of the application of the DNA-inspired information concealing algorithm over three different types of text input.

4.1.3 Performance evaluation

For the experiments conducted in this work, eleven text files were used (described in Chapter 3, Section 3.3.3). However, for simplicity and similarity of results, in this section three of them will be presented: Dubliners, Hamlet and the Email log (set1). Additional experiments and results can be found in the Appendices (Appendix B).

All the graphics presented below contain the values of similarities obtained by comparing the concealed files with the original (private) file. In each graphic, on the left (Y axis) is the cosine similarity; below (X axis) the values of the K parameter (length of sequence preserved) and in each bar, the different block sizes (B) - from 64 to 2048.

In the following sections, Term Frequency, Fixed Length, Two and Three Consecutive Terms, it is possible to observe the results obtained from the concealing of three files: Dubliners, Hamlet and the Emails log (set 1). In each section is presented a comparison between the three files, regarding the weak and strong concealing method.

4.1.3.1 Term Frequency

From Figure 4.1 to Figure 4.6, it is possible to observe not only how the K parameters and block size influence the concealing process but also the differences between the weak and strong concealing over the three files mentioned above.

To recall the reference values taken from the previous analysis, for works of the same author, the similarities were in average between 0,84 and 0,91 (excluding James Joyce - 0,97 - with high values in all tests due to the high values of similarity of his works). For works from different authors, the similarity values are, in average, between 0,82 and 0,84.

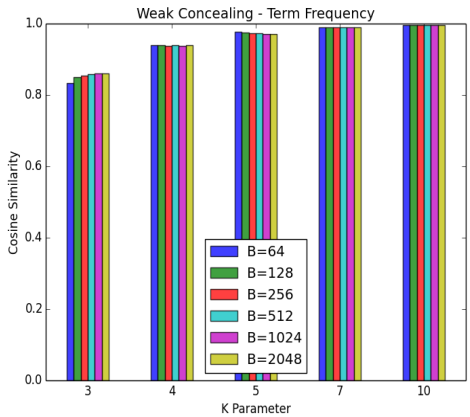


Figure 4.1: Weak Concealing, Term Frequency - Cosine Similarity (Dubliners)

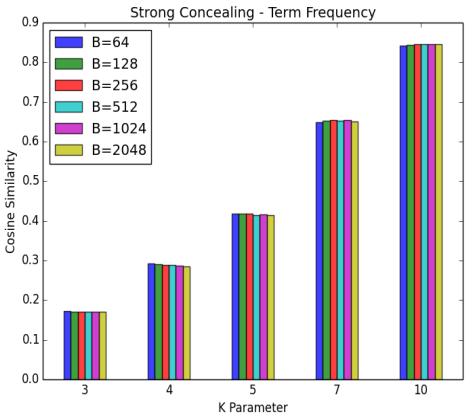


Figure 4.2: Strong Concealing, Term Frequency - Cosine Similarity (Dubliners)

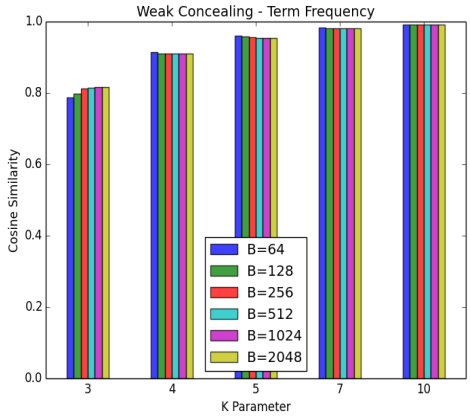


Figure 4.3: Weak Concealing, Term Frequency - Cosine Similarity (Hamlet)

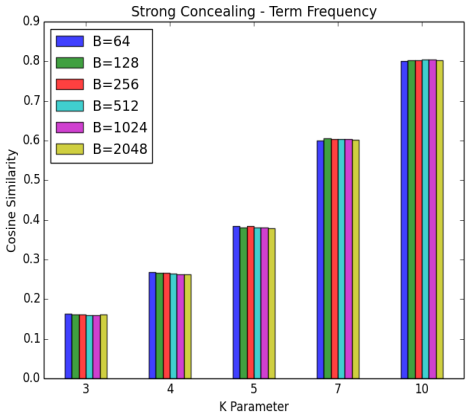


Figure 4.4: Strong Concealing, Term Frequency - Cosine Similarity (Hamlet)

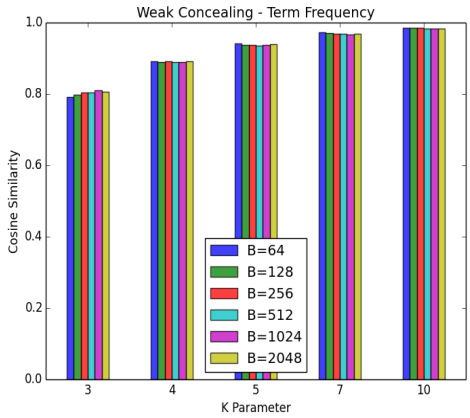


Figure 4.5: Weak Concealing, Term Frequency - Cosine Similarity (Emails)

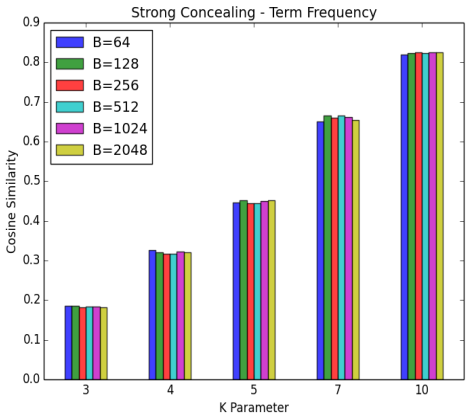


Figure 4.6: Strong Concealing, Term Frequency - Cosine Similarity (Emails)

The analysis of the graphics shows that when the weak concealing method is being used, setting the K parameter to three characters (K3), provides values of similarity are within the range of the reference values. The remaining values used in the weak concealing show a higher value of similarity with the original file.

On the other hand, if the strong concealing method is applied, it possible to preserve sequences up to ten characters. Conserving three, four, five, seven or even ten characters will provide values of similarity that are still within the reference values.

The fact of being within the reference values, represents an advantage. Consider a situation where two files are being analyzed and they have similarity values within the reference values. In these circumstances, they might look like randomly compared files, but in reality they could be the original and concealed files.

4.1.3.2 Fixed Length (5 characters)

From Figure 4.7 to Figure 4.12, it is possible to observe how the K parameters and block size influence the concealing process and the differences between the weak and strong concealing over the same three original files.

To recall the reference values taken from the previous analysis, for works of the same author, the similarities were in average between 0,70 and 0,73 (excluding James Joyce - 0,94 - with high values in all tests due to the high values of similarity of his works). For works from different authors, the similarity values are, in average, between 0,54 and 0,68.

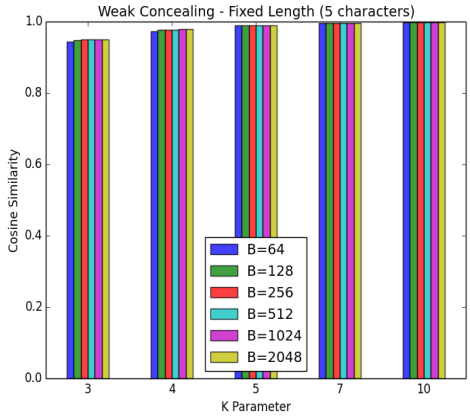


Figure 4.7: Weak Concealing, Fixed Length - Cosine Similarity (Dubliners)

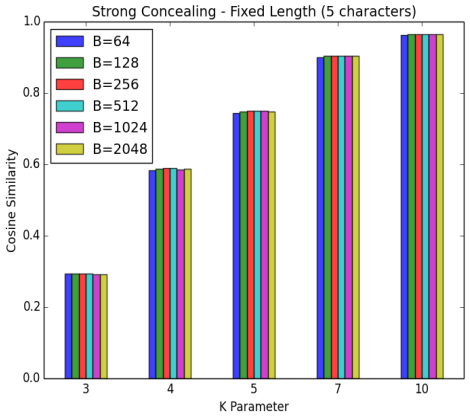


Figure 4.8: Strong Concealing, Fixed Length - Cosine Similarity (Dubliners)

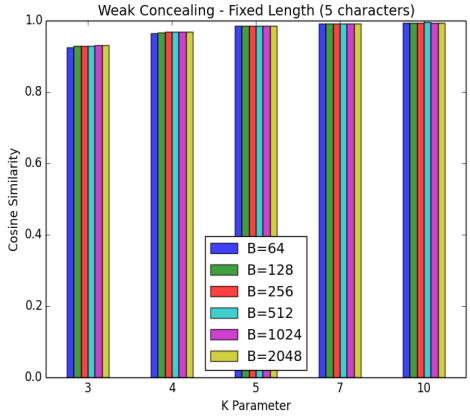


Figure 4.9: Weak Concealing, Fixed Length - Cosine Similarity (Hamlet)

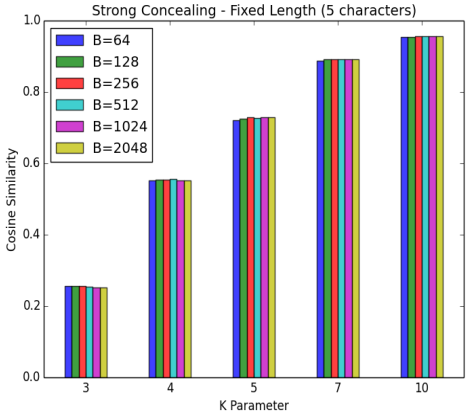


Figure 4.10: Strong Concealing, Fixed Length - Cosine Similarity (Hamlet)

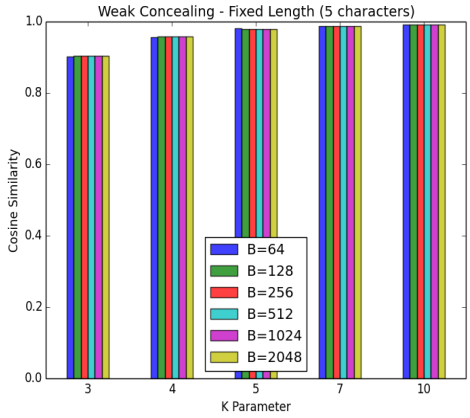


Figure 4.11: Weak Concealing, Fixed Length - Cosine Similarity (Emails)

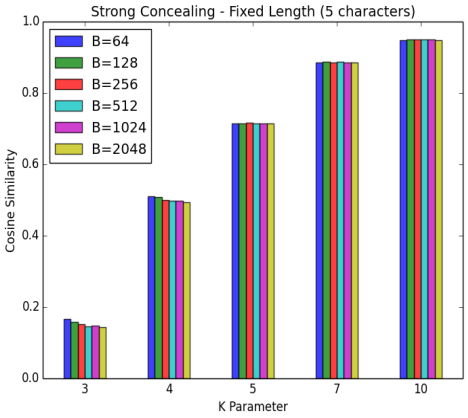


Figure 4.12: Strong Concealing, Fixed Length - Cosine Similarity (Emails)

In the case of the analysis with sequences of five characters - the average word of an English word - the results for the weak concealing method are different than the previous. In this case, the weak concealing method cannot generate a file which could resemble with another random file. It is possible to observe (Figures 4.7, 4.9 and 4.11) that the values of similarity are above 0.9, which indicates a high similarity with the originals files.

The strong concealing method, presents a better performance. It allows the concealing of up to five characters (K5) and still have values of similarity like a non-related file.

4.1.3.3 Two Consecutive Terms

From Figure 4.13 to Figure 4.18, it is possible to observe how the K parameters and block size influence the concealing process and the differences between the weak and strong concealing over the same three original files.

To recall the reference values taken from the previous analysis, for works of the same author, the similarities were in average between 0,42 and 0,53 (excluding James Joyce - 0,85 - with high values in all tests due to the high values of similarity of his works). For works from different authors, the similarity values are, in average, between 0,26 and 0,37.

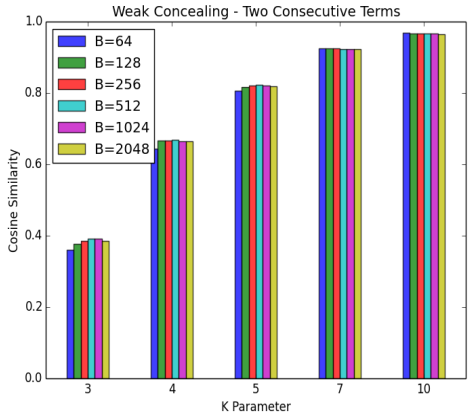


Figure 4.13: Weak Concealing, Two Consecutive Terms - Cosine Similarity (Dubliners)

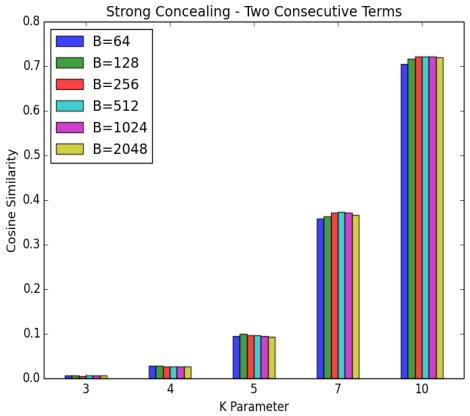


Figure 4.14: Strong Concealing, Two Consecutive Terms - Cosine Similarity (Dubliners)

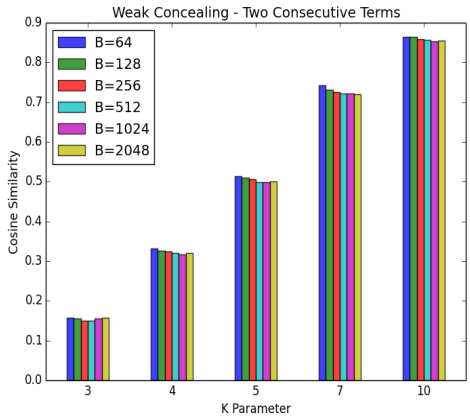


Figure 4.15: Weak Concealing, Two Consecutive Terms - Cosine Similarity (Hamlet)

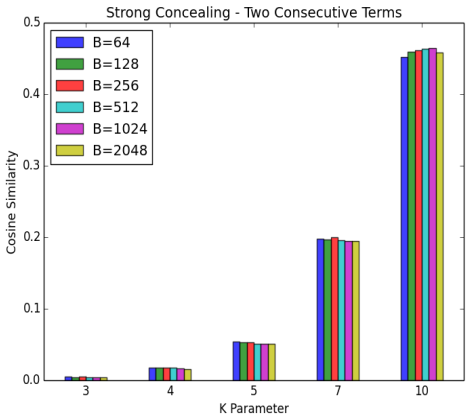


Figure 4.16: Strong Concealing, Two Consecutive Terms - Cosine Similarity (Hamlet)

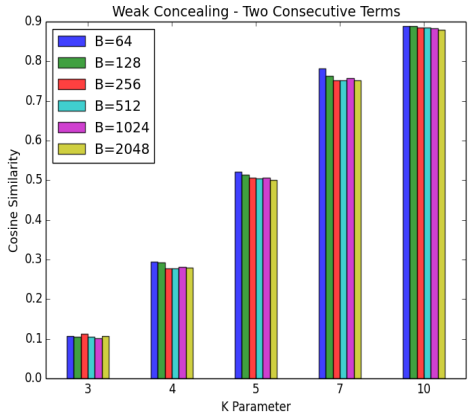


Figure 4.17: Weak Concealing, Two Consecutive Terms - Cosine Similarity (Emails)

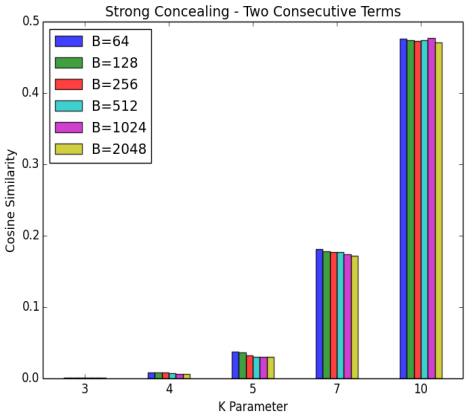


Figure 4.18: Strong Concealing, Two Consecutive Terms - Cosine Similarity (Emails)

Like it was described before (Section 4.1.2), an analysis of two consecutive terms in a text, increases the possibility of identification of the content. Duo to the connection between terms, the distribution of the content to analyze it is thereby more specific.

In this case, both methods provide relatively good results, due the lower reference values. The weak concealing method is capable of concealing up to five characters and up to three in the case of the Dubliners file. Still presenting similarity values below the reference values.

The strong concealing method has an identical behavior. However, in this case, the length of characters to preserve can go up to ten (Figure 4.14 and 4.16) and seven (Figure 4.18).

4.1.3.4 Three Consecutive Terms

From Figure 4.19 to Figure 4.24, it is possible to observe how the K parameters and block size influence the concealing process and the differences between the weak and strong concealing over the same three original files.

To recall the reference values taken from the previous analysis, for works of the same author, the similarities were in average between 0,02 and 0,08 (excluding James Joyce - 0,16 - with high values in all tests due to the high values of similarity of his works). For works from different authors, the similarity values are, in average, between 0,005 and 0,019.

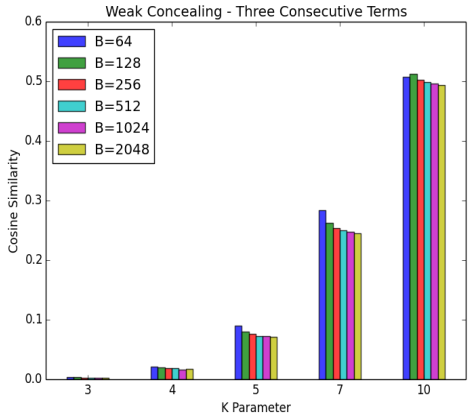


Figure 4.19: Weak Concealing, Three Consecutive Terms - Cosine Similarity (Dubliners)

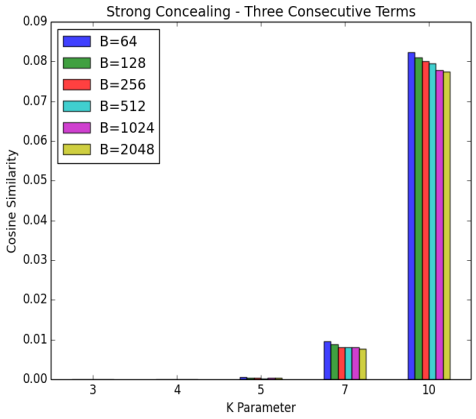


Figure 4.20: Strong Concealing, Three Consecutive Terms - Cosine Similarity (Dubliners)

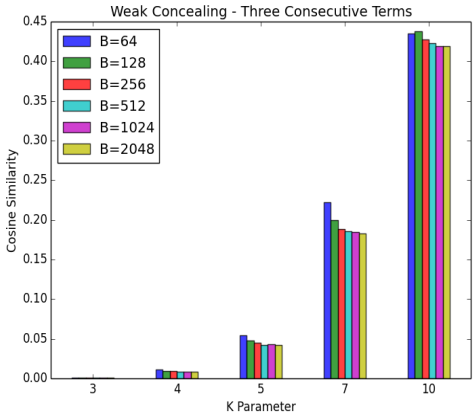


Figure 4.21: Weak Concealing, Three Consecutive Terms - Cosine Similarity (Hamlet)

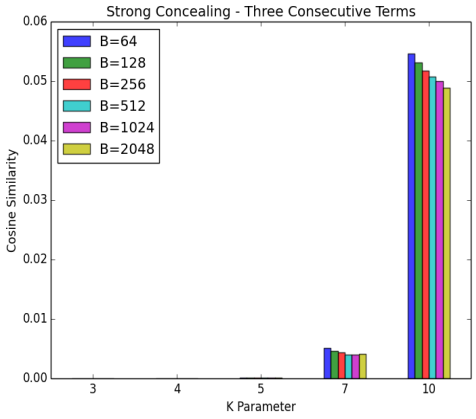


Figure 4.22: Strong Concealing, Three Consecutive Terms - Cosine Similarity (Hamlet)

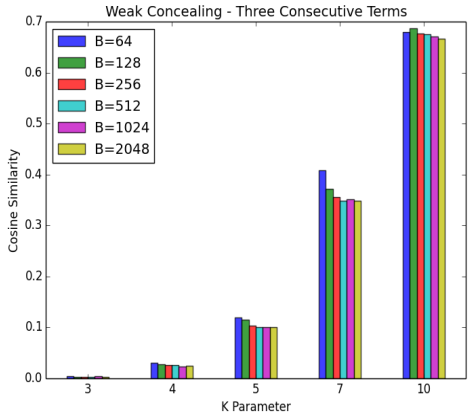


Figure 4.23: Weak Concealing, Three Consecutive Terms - Cosine Similarity (Emails)

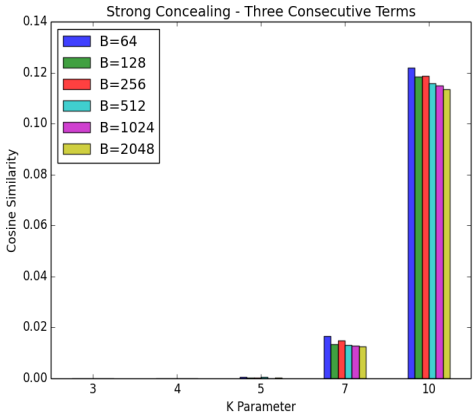


Figure 4.24: Strong Concealing, Three Consecutive Terms - Cosine Similarity (Emails)

In the last case, three consecutive terms, the K parameter has a slightly lower impact due to general values of similarity derived for the characteristics of this analysis. Having sequences of three terms, both in the original and concealed file, has a low probability of happening.

Considering the above, the weak concealing method can still preserve sequences of three and five characters (K3 and K5) and provide files with similarity values in accordance with the reference values.

For the strong concealing method, the ability is enhanced. It is possible to preserve sequences of seven and ten characters (K7 and K10) and be in accordance with the reference values.

4.1.3.5 Long Sequences

To demonstrating the idea of preserving sequences of a certain length, when an analysis of longer sequences is made, then almost no similarity exists. This is due to the characteristic of preserving sequences of length K. For instance, with file Dubliners, from Figure 4.25 to Figure 4.28 it is possible to observe analysis of sequences of twenty-five and fifty characters, respectively.

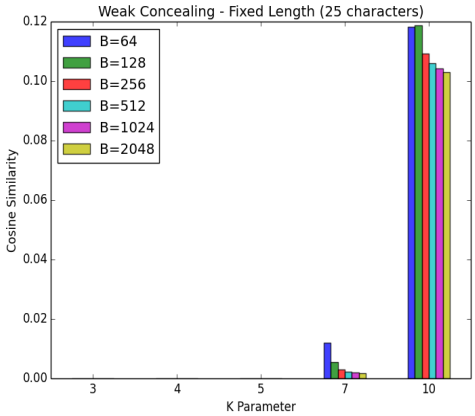


Figure 4.25: Weak Concealing, Fixed Length (25 characters) - Cosine Similarity (Dubliners)

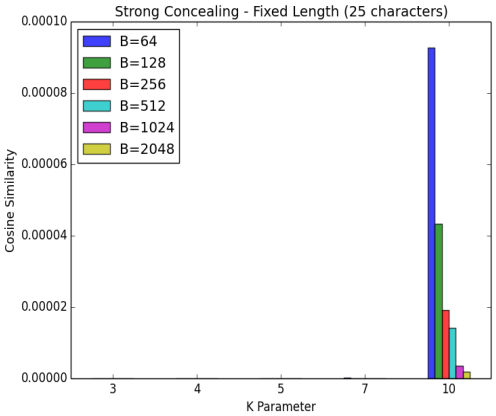


Figure 4.26: Strong Concealing, Fixed Length (25 characters) - Cosine Similarity (Dubliners)

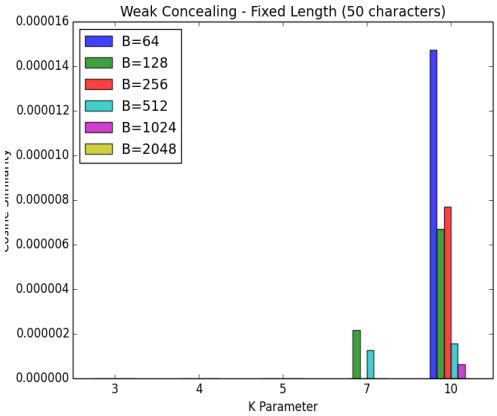


Figure 4.27: Weak Concealing, Fixed Length - Cosine Similarity (Dubliners)

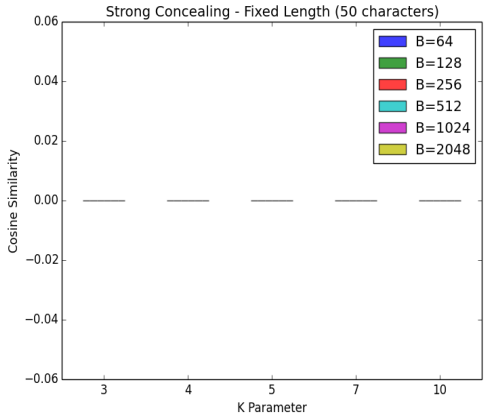


Figure 4.28: Strong Concealing, Fixed Length (50 characters) - Cosine Similarity (Dubliners)

As it is possible to observe in the graphics above (Figures 4.25 to 4.28), the values of similarity are very low. Since the length of sequences of characters to preserve is limited to ten (K10), when a comparison with sequences of twenty-five and fifty characters is made, it is expected a low similarity with the original files.

The only case where were the similarity is not close to zero, is with the weak concealing when sequences of twenty-five characters are being compared. Since the weak concealing does not add dust to the blocks, it is possible that some sequences have some resemblances with others of twenty-five characters due to the overlaps of the blocks. Other than that, the strong concealing methods already breaks the local information with the addition of dust its properties of concealing.

For the cases of fifty characters' sequences, with both the weak and strong concealing methods, the similarities are close to zero.

4.1.3.6 File size and execution time

With the analysis of the concealing itself evaluated in the previous section, it is important to measure other relevant characteristics like execution time and size of the files generated.

In order to analyze these performance metrics, there are two measures to consider. The average time spent to conceal each file with the different parameters and the size of the files generated for each configuration. As well as the size of the files generated in relation with the choice of the K parameter and the block size - B. For both there is a differentiation between the concealing type: weak or strong.

The following tables (4.9, 4.11, 4.10 and 4.12) present the average results for the time spent and the size of the concealing files produced with the weak and strong concealing method, respectively. In this case, the file being concealed was an email exchange with an original file size of 44 KB.

K \ Block	**64**	**128**	**256**	**512**	**1024**	**2048**
3	5	5	5	4	4	4
4	5	5	4	4	4	4
5	5	5	4	4	4	4
7	5	4	4	4	4	4
10	5	4	4	4	4	4

Table 4.9: Average execution time (in seconds) of the weak concealing algorithm according to K and B

K \ Block	**64**	**128**	**256**	**512**	**1024**	**2048**
3	123	121	120	120	120	120
4	130	128	127	126	126	126
5	132	130	128	127	127	127
7	138	134	132	131	130	130
10	149	140	137	135	134	133

Table 4.10: Average file size (in KB) after application of the weak concealing algorithm, according to K and B

Starting with an analysis of the weak concealing, in Table 4.9, it is possible to observe that with a longer K, there is a slight decrease of execution time. Shorter values for K, mean that there will be more - shorter - blocks to conceal, thereby, more execution time is necessary. The average execution time for this file is 4,3 $\pm$ 0,46 seconds.

Concerning the size of the files, Table 4.10 shows that, in average, the size of the files generated by the weak concealing method has approximately three times the size of the original. The overlaps of information in each block explain the increase of size. The essence of the concealing method is precisely concealing by repeats, based on the DNA families of repeats.

K \ Block	64	128	256	512	1024	2048
3	14	13	11	12	13	17
4	11	10	10	8	9	10
5	12	10	8	9	9	10
7	10	8	8	7	7	7
10	9	8	6	6	6	7

Table 4.11: Average execution time (in seconds) of the strong concealing algorithm according to K and B

K \ Block	64	128	256	512	1024	2048
3	1,2	1,2	1,2	1,2	1,2	1,2
4	1,1	1	1	1	1	1
5	1,2	1,2	1,2	1,2	1,2	1,2
7	1,3	1,2	1,2	1,2	1,2	1,2
10	1,2	1,2	1,1	1,1	1,1	1,1

Table 4.12: Average file size (in MB) after application of the strong concealing algorithm, according to K and B

In the case of the strong concealing method, Table 4.11 shows the same phenomena as in the weak concealing. Having longer execution times with shorter sequences to preserve (K). The average execution time is 9,5 $\pm$ 2,58 seconds. Representing, in average, twice as much time as the weak concealing method.

As it is possible to observe in table 4.12, the size of the generated files, is approximately twenty-six times bigger than the original one. Due to the addition of overlaps of the content and addition of dust from the content itself, along with the characteristics of the method, it is understandable the drastic increase of size.

After analyzing the size of all the concealed files, it is possible to assert that the growth of the concealed files is constant in comparison with the original unconcealed file. For all the experiences performed, with both concealing methods, weak and strong, a similar file size ratio is verified. Being approximately 2,87 times bigger in the case of the weak concealing and 25,62 times bigger for the strong concealing.

Regarding the execution time, supported by additional tests, another conclusion was drawn. In average, the strong concealing method takes twice - 2,08 times more - as much time as the weak concealing method.

4.2 Audio Files

As mentioned before, the audio files used for the experiments had the wave format (.wav extension). Other characteristic is the fact of being mono files - one audio channel - due to the algorithm's restrictions.

The starting point, like in the experiments of text files, is to compare audio files with each other. By doing so, it is possible to have reference values for the similarity measure between the files.

4.2.1 Similarity of non-concealed files

Following the approach of having reference values, the first step is to calculate the similarity values for related and non-related files. In Table 4.13 there is a description of the values obtained by measuring similarity values for related and non-related files.

Files	**Cosine Similarity (All the samples)**
Man's World / I Got You	0,86
Man's World / Get Up ...	0,96
I Got You / Get Up ...	0,94
Let's Dance / Starman	0,86
Let's Dance / Life of Mars	0,40
Starman / Life on Mars	0,75
Interstellar / Inception	0,61
SR1 / SR2	0,99
Average	0.796 ± 0.202

Table 4.13: Cosine Similarity between two audio files from the same artist

The table above (4.13), holds a comparison between files from the same artist, as well as the same type of audio. For instance, Interstellar versus Inception, is the same artist, and the same type, instrumental.

The next table (4.14), on the other hand, presents the comparison between different artists, as well as types. For instance, Inception versus SR1 (self-recording 1), instrumental and voice recording, from different artists. Other than the reasons just described, the selection of the files did not have other criteria.

Files	**Cosine Similarity (All the samples)**
Man's World / Let's Dance	0,21
Man's World / Interstellar.	0,61
Get Up ... / Inception	0,69
Get Up ... / Life of Mars	0,89
Man's World / SR1	0,45
Get Up / SR1	0,53
Life on Mars / SR1	0,66
Starman / SR1	0,82
Inception / SR1	0,86
Average	0.635 ± 0.218

Table 4.14: Cosine Similarity between two audio files from different artists

The reader may notice that the values obtained for the comparison with files from the same artist (Table 4.13), generally have similarity values above 0,5. The average similarity is 0,796 ± 0,202, which is expectable given that all the audio samples are being compared.

Regarding the comparison of audio files from different artists (Table 4.14), the similarity values are lower, accordingly. The average similarity of files from different artists is 0,635 ± 0,218. However, there are cases like Starman versus SR1 (self-recording with voice), or Inception versus SR1, that indicate high similarity values.

To better understand why that is happening, a more specific analysis has to be made. Going deeper into those cases where, theoretically, the similarity values should be lower, it is important to take the following into consideration: duration and characteristics of the audio.

Duration of the files: SR1 - 0:16 min; Life on Mars - 04:02 min; Starman - 04:18 min and Inception - 05:46 min. Characteristics of the files: SR1 - voice recording; Life on Mars - music; Starman - music; Inception - instrumental.

The following audio signals (Figure 4.29) represent and compare the four files: SR1, Life on Mars, Starman and Inception, respectively.

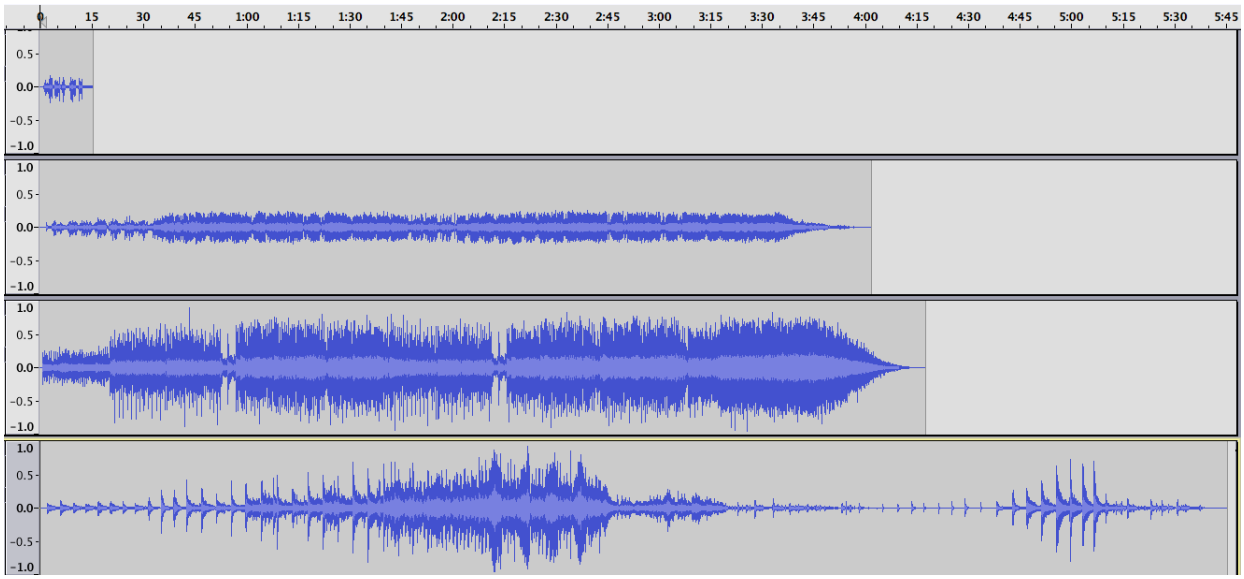


Figure 4.29: Audio Signals of SR1, Life on Mars, Starman and Inception, in this order

Visually, becomes clearer why does the similarity of such a small voice recording - SR1 - is high when compared to Inception. Moments of silence or no voice, match some moments of silence or no instruments from Inception. That is why the similarity with this file is higher. Even if the terms of audio content, or duration, the files are different, their signals share common characteristics.

The signal from Life on Mars - second from the top - has considerably different characteristics from Inception. This is also translated in the lower similarity values. Life on Mars vs SR1: 0,66, and Inception vs SR1: 0,86.

4.2.2 Duration influence

To better understand how different parts of an audio file can influence the similarity values, the files were cut. The files were cut according to their size. Reducing the duration to twenty-five, fifty and seventy-five percent of the original duration. Table 4.15, below, holds the comparison between the different times of each file. For instance, the first row compares the first twenty-five percent of "Man's World" with the first twenty-five percent of "I Got You".

Duration (%)	**Man's World vs I Got You**	**Let's Dance vs Starman**	**Man's World vs Let's Dance**
25 / 25	0,8812	0,56	0,25
25 / 50	0,8973	0,41	0,38
25 / 75	0,8968	0,34	0,48
25 / 100	0,8818	0,60	0,20
50 / 50	0,8746	0,53	0,34
50 / 75	0,8746	0,47	0,45
50 / 100	0,8676	0,67	0,18
75 / 75	0,8618	0,58	0,44
75 / 100	0,8543	0,72	0,18
Complete file	0,86	0,86	0,21

Table 4.15: Cosine Similarity with variation of the file duration (time)

The results gathered above were surprising in the case of James Brown's music. The values of similarity of the music "Man's World" and "I Got You" are placed between 0,85 and 0,89. Being 0,86 the value of similarity between the two complete files. In order to understand what is the phenomena that is happening in this case, further analysis has to be done.

Following (Figure 4.30) there is a graphical visualization of the audio signals from the files being compared in figure tb:durationsvariationaudio.

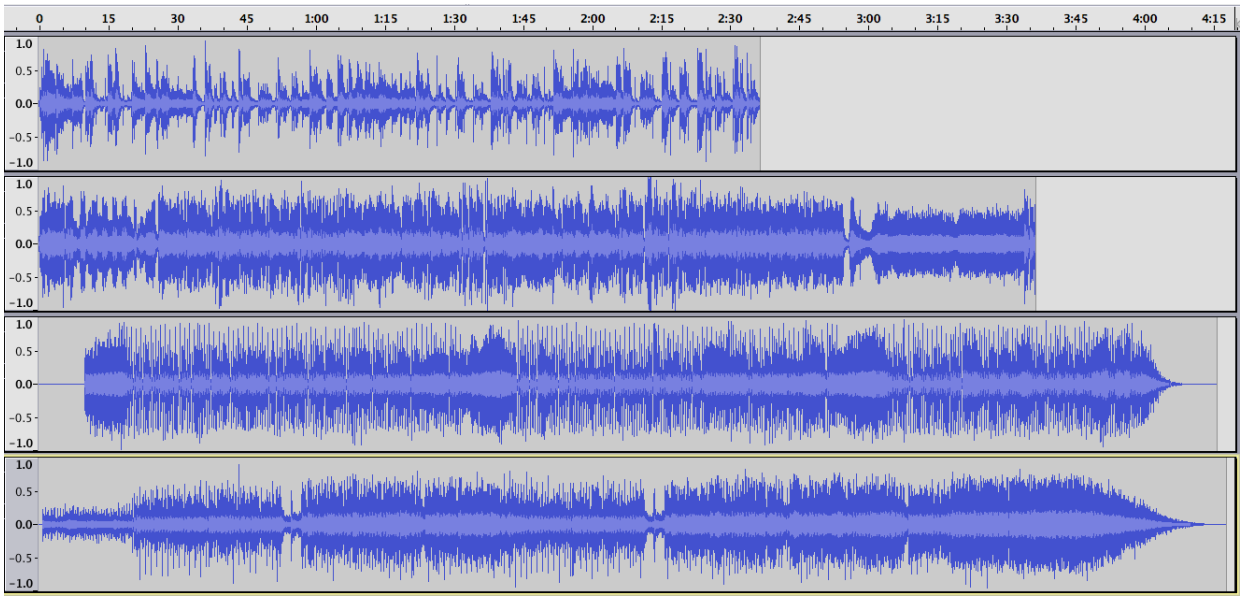


Figure 4.30: Audio Signals of Man's World, I Got You, Let's Dance and Starman, in this order

In this case, (Figure 4.30), it is not possible, in a visual way, to point out differences that would allow an understanding of the values of similarity obtained. To provide a thorough analysis, each file - "Man's World" and "I Got You" - was divided in four parts.

The first part contains the audio from the beginning to the time that corresponds to first quarter of the length. The second part, from the first quarter to the half. Third part, from the half to the third quarter. Finishing with a fourth part containing the audio from the third quarter till the end.

The similarity measures taken with the previously described parts are presented in Table 4.16.

File Sections (%)	**Man's World**	**I Got You**
0 - 25 / 25 - 50	0,9913	0,9911
0 - 25 / 50 - 75	0,9867	0,9909
0 - 25 / 75 - 100	0,9910	0,9746
25 - 50 / 50 - 75	0,9945	0,9922
25 - 50 / 75 - 100	0,9938	0,9651
50 - 75 / 75 - 100	0,9937	0,9630

Table 4.16: Cosine Similarity with variation of the file duration (time)

With the additional data provided by table 4.16, is now possible to better understand the similarity values between "Man's World" and "I Got You". The comparison between parts with different length is around 0,88 due to the similarity of the parts of the music itself.

The previous table (4.16) indicates that the four parts of the songs are very similar between themselves. Consequently, when comparing for example the first quarter or the first three quarters, it is natural that a high similarity is the result.

This is, however, a particular case. As it is possible to observe in table 4.15, "Let's Dance" versus "Starman" and "Man's World" versus "Let's Dance" have lower similarity values, which expected and natural.

4.2.3 Performance evaluation

The performance evaluation of the audio files is made recurring to computer analysis for short sequences and an audible analysis for longer sequences. Not only by the impossibility of performing such large computations for longer sequences (for instance K10000), but also to assess how easily can a human - the author in this case - identify information or recognize samples from the concealed audio files.

In the following sections, Sample Frequency and Human analysis it is possible to observe the results obtained from the weak and strong concealing of SR1. Additionally, the weak concealing results are also presented for "Let's Dance", "Man's World" and "Inception" files.

4.2.3.1 Sample frequency

As it will be presented below, the similarity of the concealed files, when considering all the samples, is very high. The reason behind this, is the higher volumes of data to conceal and a bigger alphabet. When text is considered, for example Ulysses, from James Joyce, it has close to 250000 terms. Which means around one and a half million characters and it is already a big file to conceal. However, considering audio, since each second has 44100 samples, a short audio clip of sixty seconds has around 2.6 million samples to conceal.

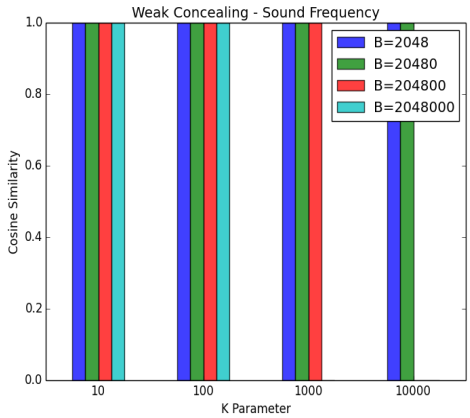


Figure 4.31: Weak Concealing, Sound Frequency - Cosine Similarity (SR1)

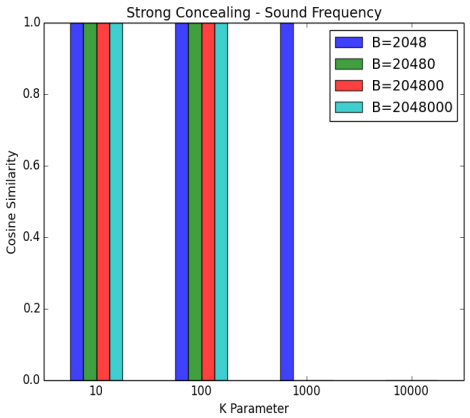


Figure 4.32: Strong Concealing, Sound Frequency - Cosine Similarity (SR1)

Observing Figure 4.31 and Figure 4.32, the reader can confirm what was described before. When an analysis based on all the samples is made, the similarity will be very high. In both cases, weak and strong concealing, the similarity values are 0,99.

In both figures referred above, there are values that are not present for K1000 and K10000. This is due to computation limitations and the enormous amount of memory needed to conceal such files with the indicated parameters.

This sort of evaluation with be made in a future work, with the support of a more powerful infrastructure to perform the experiments. However, based on previous results and conclusions already drawn, it is predictable that with longer sequences, the files will present lower similarity values.

The experiments with other files, support what is above described. The cosine similarity between the files is very high. Figures 4.33 and 4.34 confirm that the values of similarity are also close to one.

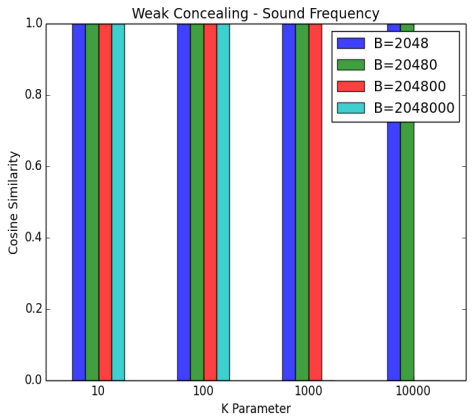


Figure 4.33: Weak Concealing, Sound Frequency - Cosine Similarity (Inception)

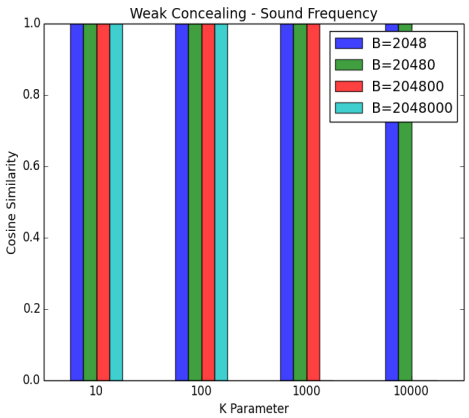


Figure 4.34: Weak Concealing, Sound Frequency - Cosine Similarity (David Bowie - Let's Dance)

When longer sequences are being compared between the original and concealed files, for instance 10 or 100 samples, the results remain high in terms of similarity. Like mentioned earlier, if it would be possible, computationally, to compare sequences of 1000, 10000 or even 100000 samples, then the similarity values would start to be lower. However, the computer analysis for those cases is left for future work with more computer power.

For now, is possible to use the human audition, and analyze how perceptible are the concealed files with longer sequences of K. For instance, K10000, represents the preservation of sequences of 10000 samples, which is may be possible for the human ear to identify some content in the sound, like artist, genre or content. To find out, the next section contains the results of the experiments to identify concealed files by listening.

4.2.3.2 Human analysis

Previously, it was shown that similarity values of the comparison of all the samples is very high. However, it was explained that with longer sequences the similarity values would decrease. In this point, it would be interesting to discover after which values of K it is possible for a person to understand, or recognize parts of the concealed audio.

To achieve the above, the author listened to all the concealed audio files and evaluated when was possible to identify some sound characteristic. The files used were the ones concealed with K10, K100, K1000 and K10000, with different block sizes - 2048, 20480, 204800 and 2048000.

There were two concealing types to differentiate, the weak and the strong.

- Strong concealing

  For the strong concealing method, for K10 and K100, the output is audible noise. Smoother with bigger block sizes. Other than that is imperceptible any kind of voice or music recognition. However, with K1000, in some parts of the files, the output has some resemblance with computerized voice, with a very fast pace. Files with K10000 have not been tested with the strong concealing type due to the limitations mentioned before.

- Weak concealing

  With the weak concealing method, the case is however, different. For values of K10 and K100 the evaluation is the same as the strong concealing - noise and without possibility of identification or recognition of the file. Although, for K1000 and K10000 the evaluation is different.

  For instance, with K1000 and smaller block sizes, it was already possible to differentiate the audio between music, voice or instruments. With bigger blocks sizes the different is that the audio looks smoother and faster.

  Finally, with K10000, substantial differences were observed. In this case it was possible to identify music, singers, and speech (words). The findings make sense since 10000 samples represent roughly a quarter of a second, which is enough time for a person to identify certain characteristics of the audio.

The findings are particularly interesting due to the possible applications it could have. The audio could be concealed (i.e. without disclosing all information) and still be useful for audio analysis, for instance, mood detectors based on tones or words detected in the file.

The possible applications are discussed later in the Cloud Applicability (Chapter 5) and Conclusion (Chapter 6). Following there is an analysis of the file size and execution times with audio files.

4.2.3.3 File size and execution time

Below are presented the measurements possible to obtain with the audio experiments. Not as detailed as the one possible to obtain with text, but still relevant and significant. Additional measurements for the strong concealing method would be needed for a more thorough evaluation.

Original File Size: 1,4 MB - SR1 (Self Recording 1)

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	3,20	4,18	4,20	4,20
Strong	33,23	34,16	39,01	-

Table 4.17: Average file size (in MB) after application of the weak and strong concealing algorithm on SR1 file

The output of the files generated, in the case of the audio concealing, is identical to text, in terms of size ratio. Which means that concealed audio files with the weak method are, in average, 2,81 times the size of the original. With the strong concealing, the files are, in average, 24,33 times the size of the original. This is another evidence that the DNA-inspired information concealing method is consistent with the generation of the concealed files.

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	8	4	4	4
Strong	1462	147	599	-

Table 4.18: Average execution time (in seconds) of the application of the weak and strong concealing algorithm on SR1 file

Regarding the execution times, derived from the impossibility of gathering considerable results from the execution of the strong concealing method due to the limitations mentioned before, it is not possible to provide concrete estimates of execution times in the case of the audio.

Nevertheless, there are other relevant and interesting findings. The following section presents a summary of the previous experiments and highlights the main findings.

4.3 Discussion

Until now, this chapter presented the results and findings regarding the experiments with text and audio files for the DNA-inspired information concealing algorithm.

It was shown that the similarity between works or texts from the same author tend to have higher similarities than others from different authors. It was equally shown that when analyzing text with pairs of words, or even triplets, the results are more specific. The usage of this type of measure provides for instance a way of comparison of files with specific authors of file originators.

The size of the files showed importance in the evaluation. The shorter the texts, higher the different between them. However, increasing the text length, shows an increase of similarity, in all the cases due to a higher frequency of terms.

Nevertheless, the evaluation of the DNA-inspired information concealing algorithm revealed new findings. In addition to the capabilities described by the authors, it was possible to show values for K which would lead to values of similarity in line with those obtained in the comparison of random texts.

For instance, in the four types of analysis - term frequency, fixed length, two and three consecutive terms - *the weak concealing method can preserve sequences up to five characters and still presents similarity results like random files.*

In the same situation, the *strong concealing method could present similar results and preserve sequences up to seven or even ten characters* in some cases.

The strong concealing execution time showed to be, in average, twice as much the time as the measured in the weak concealing method. Both of them taking longer execution times with small values for the K parameter.

Concerning the block size, it was found that, in the case of text, with the values used, there were not significant changes in the similarity values obtained. The higher variations occurred when the similarity values were too low, evidencing the - small - differences. Analogously, the block size in the case of the audio experiments, did not reveal substantial differences. Although it was possible to notice audible differences when the K value was higher in the audio, further testes will be made in future work in order to understand those differences in terms of similarity (refer to Future Work - Section 6.2).

One of the findings that are not as positive as expected, is the file size. The *weak concealing method produces files three times bigger* than the original file - which is acceptable. However, the strong concealing type, due to its inherent concealing characteristics, generates files almost *twenty-six times the size of the original files* - representing a big con in terms of storage and memory needed for instance.

Regarding the audio files, the experiments allowed to confirm that file size ratio of the generated files is identical to text. Despite being a different file type, since the treatment given by the algorithm is identical, the ratio remains at three and twenty-six times bigger files, for the weak and strong concealing methods, respectively.

Execution times for the audio files it was one aspect that could not be confirmed whether or not the characteristics were identical to the ones obtained in with text files.

It was also shown that the cosine similarity measures taken between audio files from the same artist or different artists are higher with files from the same artist. Additionally, the experiments performed with different file lengths, showed - as in the text experiments - that the duration of the file as significance when verifying how similar two files are. It was also showed that, in terms of values for the audio signal, one of the audio files - Man's World - had its four split parts similar to each other, resulting in high similarity values when comparing different sizes of the file.

Considering the similarity between concealed and original files, it was showed that when comparing all the samples, and due to the amount of samples, the values are close to 1, both with weak and strong concealing. Being expected lower similarity values of long sequences of samples how be compared, for instance 1000 or 10000 samples - to be experimented in future work with the support of a more powerful equipment for analysis purposes.

When the audio analysis was made by hearing and trying to identify some characteristic of the concealed files, the values of K1000 and K10000 showed that after these values start to be possible to recognize and differentiate characteristics. For instance, differentiate weather it is a music or a person speaking.

Another interesting insight provided by this work, was for example, the possibility of using the Term Frequency analysis to identify languages, due to the alphabet of the files. Comparing the concealed files with certain files would provide higher or lower values of similarities if the languages match, or not.

Following, a description of the advantages and disadvantages of this algorithm.

4.3.1 Advantages and Disadvantages

Considering what was described and analyzed before, there are some points to highlight regarding the usage of this algorithm.

Advantages:

- Accepting unstructured data as input, unlike many other algorithms and techniques which demand structured data like CSV or XML with organized contents
- It is not necessary to define specific attributes conceal. For instance, other methods require a preparation and definition of attributes such as name or address to prepare the anonymization process.
- Easy and simple to deploy. Choosing the concealing type (weak or strong), the value for K and Block size and the algorithm does the rest.
- Ability to conceal audio files

- Conceal the files using local data only. Not adding new symbols/samples to the file alphabet.

Disadvantages:

- Generating files larger than the original ones. Which could be an obstacle if storage is limited.
- Accepting only mono audio files, when nowadays the majority of the audio files are at least dual channel.

The execution time it is not placed in either the above groups. The reason is that it may, or may not, be a disadvantage. For some cases where performance is a must, then could represent a disadvantage. However, the majority of the final applications for this processes does not need immediate availability. So, the execution time would not be necessarily classified as a disadvantage.

The last section of this chapter, which follows, contains suggestions that could improve the performance and capabilities of the algorithm.

4.3.2 Suggested Improvements to the Algorithm

After the experiments and analysis to the algorithm, is possible to suggest improvements. As it was refereed, the code is implemented in Matlab, in an experimental version. According to this, and code analysis, if the algorithm is implemented in such a way that could take advantage of multiple cores or processors, the execution times would be reduced, representing a performance increase.

In the same line of the previous, if the algorithm would be implemented in another platform, let's say, C language, taking advantage of parallel work, then would be expected a noticeable improvement.

In terms of files supported, an addition of some file extensions would represent a great advantage in comparison with other methods. Currently, the method is accepting *txt* and *wav* file formats. Supporting files formats like CSV or XML, or even PDF files (with certain content restrictions such as text only). Regarding the audio, considering the support of dual channel audio files or additional formats could also improve the algorithm's capabilities.

The following chapter, Applicability to the Cloud (Chapter 5), contains the analysis of the method's applicability for the cloud.

CHAPTER 5

Applicability to the Cloud

This chapter provides an overview about the cloud and the respective applicability of the DNA-inspired concealing algorithm. Describes what changes with the cloud computing paradigm, what does it mean and the main players involved.

The next section starts by providing an overview of the cloud. The following sections refer methods of data privacy protection and its applicability.

5.1 Overview

Cloud computing is not an entirely new concept. While the concept only started to become widely known in the last decade, its origins come from far behind. Cloud computing, in its essence, means relying on a set, or a network of computers, instead of only one machine for storage and processing of data.

In the past, when a single computer was the size of an office, a user would access a terminal, for a limited time, and use the available computing power. Nowadays, the previous model is similar. However, instead a single computer occupying an entire office, was is available is a whole network of computers at global scale.

This paradigm is changing the way computational power is provided, or even how storage and applications are being used. For instance, a traditional application has a fairly predictable load, a specific set of devices to be used in and a fixed cost.

On the other hand, a cloud application does not have a predictable load. The infrastructure needs to be elastic and provide the same quality service for a high number of end users.

It is used on a wide variety of devices and, according to the demand, the maintenance costs can be variable.

Nowadays, the need of saving all the documents on a hard drive or burning a CD is a thing of the past. There fast internet connections, cheap and sometimes even free online storage services available. This lead users and business to save their files online.

With such a high demand, major companies already offer the most variate pack of services. Computing power, storage, applications or services are examples of what companies offer. Companies like Google, Amazon, Microsoft, IBM, Dropbox, Apple, Mega, and many more, offer those kind of services. From commercial services, to enterprise or academic. However, when data is uploaded, sometimes it not clear who is the proprietary: the user - who upload the data - or the companies which hold user data in their servers and sometimes in more than one across the world.

Since every company has its own terms of service and regulations, there is, naturally, a raise of security and privacy concerns. Ownership or control of the data itself are just two points that can raise several questions. Dealing with these concerns and assuring privacy to the data should be a duty performed by whoever uploads data to the cloud.

Besides getting to know well all the assurances and guaranties provided by the companies, the data uploader can take additional measures. This is where anonymization and concealing methods could be applied. According to different user needs, and different data privacy requirements, there are some cases where the application of this additional measures could be useful.

5.2 Data Privacy Protection Requirements

Considering what was described before, there are several ways of anonymizing and concealing data. However, it is not the only way of providing data privacy. With different approaches, different requirements are met.

Other typical security mechanisms like password protection or encryption also provide privacy, at a different level. Provided that an attacker does not have access to password or the encryption cannot be broken, the data is kept safe. However, with some security flaw, private data can be exposed and then, there is no privacy assurances, at all.

Providers of cloud services offer different privacy assurances and protection against data disclosure, from securing communications and infrastructures, to data itself. However, even this has its risks. To avoid unnecessary risks, the user should take its own measures to protects sensitive data.

In the case of taking additional protection measures over the data itself, other the ones that its own infrastructure offers, it is possible two differentiate between two groups: reversible or non-reversible methods. Which vary according to each user needs or and specific data purposes.

- Reversible

  The case of reversible methods to ensure data privacy, is a category where for example encryption or password protection methods can be placed. These methods help protecting from unauthorized access to the data. However, it is not fail safe. Governmental authorities could, for instance, order companies to reveal information from a certain user or entity, thereby disclosing private information.

- Non-reversible

  On the other hand, there are non-reversible methods. In this category, can be placed for example the DNA-inspired information concealing algorithm or Black Marker. What happens it that any modification made over the data is irreversible. Thereby, the content and (or) the originator are protected.

Below (Table 5.1) there are some examples of methods that can be applied to the cloud and provide data privacy.

Method	**Reversibility**	**Data Usability**	**Performance Impact**
Encryption	Yes	None	Low
Password Protection	Yes	None	None
Tokenization	Application Dependent	Partial	None
DNA-inspired algorithm	No	Partial	Low
K - Anonymity	No	Partial	None

Table 5.1: Examples of methods of data privacy protection applicable to the cloud

It is possible to observe that there are mainly, two principal differentiating factors in the table: reversibility and data usability. Of course the type of method used will be dependent on each user needs.

The next section describes the applicability of anonymization and concealing techniques in the cloud. Including the DNA-inspired information concealing algorithm.

5.3 Applicability

There are, of course, different user necessities of data privacy protection. If the data holder is the type of user that needs to restore data back to its original state, then, the algorithm being evaluated in this work is not suitable due to the non-reversibility of the concealing method.

Conversely, there are many cases where data does not need to be restored to its original state. Thereby, applying the DNA inspired information concealing algorithm would provide the needed privacy protection. The algorithm can provide privacy protection to the content of the data being concealed. This is, the originator is not necessarily hidden when the application of the algorithm in enforced.

Since the application of the algorithm can provide content privacy, there several cases where this application could be useful.

When the data would be uploaded to some cloud storage service, the application of the algorithm with the desired parameters (length of the sequences to preserve) would ensure that once the data is saved on the online servers, the sensitive content is already concealed. Meaning that, the same data could be protected against intruders, data breaches or other non-authorized accesses.

The downside of above example, is the additional storage space needed to hold the concealed data. Although, nowadays, storage capacity is rapidly increasing. In this way the concealed data stored online, could then be accessed by anyone, and still be protected. This could provide an additional level of privacy for research.

Consider a research department or security company working on spam filters and analysis. The concealing of email contents would allow additional input for those kinds of systems due to the search for specific keywords. Another example could be for instance location records.

Consider a mood detector project of application for text or audio. With the data stored online and the application of the algorithm, anyone could access the data without compromising privacy. Meaning that the data available could be more than just the specific data treated and anonymized for the effect.

For instance, network data traces, accounting logs. Those kinds of logs could equally be available online, but not with the application of this algorithm, since they could need to be rolled back to the original state. That would allow researchers, or anyone, to find for example it their website, or they company is present in the logs by searching for the specific keywords.

To summarize, that possible applications of the method in the cloud context, would directed to the academic or enterprise dimensions. The single and typical commercial user, perhaps would not find advantages in making used of this type of protection.

Other than the previously described, the execution times would not be necessarily considered a problem for the type of application described. It is a one-time only action, with no further need for maintenance or supervision. Could be performed either by the uploader or by the online storage provider.

With a description of the possible applications of this algorithm in the cloud and the evaluation of the algorithm itself, it is possible to present the Conclusion of this work (Chapter 6) in the following chapter.

CHAPTER 6

Conclusion

The previous chapters presented the work made for this dissertation. The following sections contain the final considerations of the work as well as the future work.

6.1 Final Considerations

The focus of this work was data privacy protection and its possible applications for the cloud. To do that, algorithms, tools and metrics were presented. Then, a specific algorithm was evaluated and its performance and cloud applications were discussed.

This work began by providing an overview of methods and algorithms available to provide data anonymization and concealing. Along with the algorithms, there were equally showed tools that perform the necessary data transformations. As well as metrics used to measure the anonymization and concealing performances. This was followed by an evaluation of the DNA-inspired information concealing algorithm over text and audio. Which also revealed additional capabilities of the algorithm.

It was showed that the average file size ratio for files concealed with the weak and strong concealing methods are, respectively, three and twenty-six times the size of the original. It was equally shown that the in average the strong concealing method takes twice as much time as the weak method. It is important to note that, the execution times could be reduced, if an optimized version of the algorithm is developed. Additionally, it was demonstrated that, under certain parameters, the concealed files could provide similarity values in line with those presented by randomly compared files.

As a capability not present in many algorithms, the ability to conceal unstructured data, demonstrated its usefulness. Not only to conceal a variety of text files, but also audio files.

The concealing of audio files showed that after certain high values of K (size of sequence to preserve) there were parts of the audio that could be recognizable for a human and still, the content could not be disclosed.

Other than the findings directly related with the algorithm, it was possible to understand how this and other algorithms work and provide the necessary data privacy. To complement what was previously described, there was a discussion about the cloud anonymization and concealing requirements and how applicable to the cloud this and other methods could be. The respective implications were equally discussed. For instance, the performance impact and its pros and cons.

Regarding the author's contributions for this work, it is now possible to confirm that the expectations were fulfilled. Considering the above described and the work presented in the previous chapters, everything that was proposed to be done in the beginning was successfully achieved. Even though there were obstacles in the way, persistence and perseverance prevailed and led to the work that was presented.

In the appendices there is a Work Plan section (Appendix A) with all the information about the management of this work. As well as the obstacles found along the way, how the situations were managed and the restructuring and redefining of work plan.

The research nature of this work, gave the author the opportunity to acquire new researching capabilities, discover new interests and improve several aspects both personal and academically. Moreover, it was possible to develop and present new findings and conclusions in the field of data privacy protection.

The conclusion and presentation of this report, does not mean that the work is already finished. There is future work to be done. The next section describes what is yet to be done about this work.

6.2 Future Work

The work developed in this Master Dissertation was able to provide new results and conclusions about a method for cloud data protection. As well as its applicability. However, due to certain limitations, the audio experiments could not be executed like initially intended. Due to the above, and to provide a thorough analysis and a complete set of results, the author will perform the additional experiments.

After guarantying the necessary conditions for the execution of the experiments, for instance, cloud computing processing power, the audio experiments will be performed. Furthermore, with the conclusion of the audio experiments, the results will be added to the paper currently being prepared about this work.

With the addition of the remaining audio experiments to the paper, and its respective completion, the paper will be submitted in October 2016, to the 2017 edition of the IEEE International Conference on Communications.

Provisional paper reference: Silva, P., Kencl, L., Monteiro, E. (2016). Data Privacy Protection for the Cloud. IEEE International Conference on Communications.

References

[Bochkarev et al., 2012] Bochkarev, V. V., Shevlyakova, A. V., and Solovyev, V. D. (2012). Average word length dynamics as indicator of cultural changes in society. *arXiv preprint arXiv:1208.6109*.

[Boschi and Trammell, 2011] Boschi, E. and Trammell, B. (2011). Ip flow anonymization support. *Internet Engineering Steering Group (IESG)*.

[Cragin et al., 2007] Cragin, M., Heidorn, B., Palmer, C., and Smith, L. (2007). An educational program on data curation. *ALA Science & Technology Section Conference*.

[Daintith, 2016] Daintith, J. (19-06-2016). "anonymization." a dictionary of computing. 2004. - http://www.encyclopedia.com/doc/1o11-anonymization.html.

[Dalenius and Reiss, 1982] Dalenius, T. and Reiss, S. P. (1982). Data-swapping: A technique for disclosure control. *Journal of Statistical Planning and Inference*, 6:73–85.

[Davies and Bouldin, 1979] Davies, D. and Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 224–227.

[Fan et al., 2004] Fan, J., Xu, J., Ammar, M. H., and Moon, S. (2004). Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-base scheme. *Computer Networks*, pages 253–272.

[Hundepool and Willenborg, 1996] Hundepool, A. J. and Willenborg, L. C. R. J. (1996). mu- and tau-argus: Software for statistical disclosure control. *Third Internation Seminar on Statistical Confidentiality*.

[Kencl and Loebl, 2010] Kencl, L. and Loebl, M. (2010). Dna-inspired information concealing: A survey. *Computer Science Review*, pages 251–262.

[Kraskov et al., 2003] Kraskov, A., Stögbauer, H., Andrzejak, R., and Grassberger, P. (2003). Hierarchical clustering based on mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Li et al., 2005] Li, Y., Slagell, A., Luo, K., and Yurcik, W. (2005). Canine: A combined conversion and anonymization tool for processing netflows for security. *13th Intl. Conf. on 13th Intl. Conf. on Telecommunications Systems*.

[Luo et al., 2005] Luo, K., Li, Y., Ermopoulos, C., Yurcik, W., and Slagell, A. (2005). Scrub-pa: A multi-level multi-dimensional anonymization tool for process accounting. *13th Intl. Conf. on 13th Intl. Conf. on Telecomunications Systems*.

[Matthias et al., 2015] Matthias, T., Alexander, K., and Bernhard, M. (2015). Statistical disclosure control for micro-data using the R package sdcMicro. *Journal of Statistical Software*, 67(4):1–36.

[Microsoft, 2014] Microsoft (2014). Protecting data and privacy in the cloud - http://download.microsoft.com/download/2/0/a/20a1529e-65cb-4266-8651-1b57b0e42daa/protecting-data-and-privacy-in-the-cloud.pdf.

[Mivule and Anderson, 2015] Mivule, K. and Anderson, B. (2015). A study of usability-aware network trace anonymization. *Science and Information Conference*, pages 1293–1304.

[Petitcolas et al., 1999] Petitcolas, F. A., Anderson, R. J., and Kuhn, M. G. (1999). Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078.

[Poulis et al., 2014] Poulis, G., Gkoulalas-Divanis, A., Loukides, G., Skiadopoulos, S., and Tryfonopoulos, C. (2014). Secreta: A system for evaluating and comparing relational and transaction anonymization algorithms. *17th International Conference on Extending Database Technology (EDBT 2014)*, pages 620–623.

[Prasser and Kohlmayer, 2015] Prasser, F. and Kohlmayer, F. (2015). Putting statistical disclosure control into practice: The arx data anonymization tool. *Gkoulalas-Divanis, Aris, Loukides, Grigorios (Eds.): Medical Data Privacy Handbook*.

[Shannon, 2001] Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

[Singhal, 2001] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull. 24.4*, pages 35–43.

[Slagell et al., 2006] Slagell, A., Lakkaraju, K., and Luo, K. (2006). Flaim: A multi-level anonymization framework for computer and network logs. *Large Installation System Administration Conference (LISA '06)*.

[Solove, 2009] Solove, D. (2009). *Understanding Privacy*. Harvard University Press.

[SourceForge, 2015] SourceForge (20-12-2015). Open anonymizer - http://sourceforge.net/projects/openanonymizer/.

[Sweeney, 1997] Sweeney, L. (1997). Datafly: A system for providing anonymity in medical data. *Eleventh International Conference on Database Security*.

[Sweeney, 2000] Sweeney, L. (2000). Simple demographics often identify people uniquely. *LIDAP-WP-4*.

[Sweeney, 2002a] Sweeney, L. (2002a). Achieving k-anonimity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzzines and Knowledge-based Systems*, pages 571–588.

[Sweeney, 2002b] Sweeney, L. (2002b). K-anonimity: A model for protecting privacy. *International Journal on Uncertainty, Fuzzines and Knowledge-based Systems*, pages 557–570.

[TMF, 2015] TMF (10-12-2015). Open anonymizer - http://www.tmf-ev.de/home.aspx.

[Wiles et al., 2008] Wiles, R., Crow, G., Heath, S., and Charles, V. (2008). The management of confidentiality and anonymity in social research. *International Journal of Social Research Methodology*, 11(5):417–428.

[Yurcik et al., 2007] Yurcik, W., Woolam, C., Hellings, G., Khan, L., and Thuraisingham, B. (2007). Scrub-tcpdump: A multi-level packet anonymizer demonstrating privacy/analysis tradeoffs. *3rd IEEE International Workshop on the Value of Security through Collaboration (SECOVAL)*.

[Yurcik et al., 2008] Yurcik, W., Woolam, C., Hellings, G., Khan, L., and Thuraisingham, B. (2008). Measuring anonymization privacy/analysis tradeoffs inherent to sharing network data. *Network Operations and Management Symposium*.

Appendices

APPENDIX A

Work Plan

This Appendix contains the detailed work plan and the respective management.

The present Master Dissertation was an international project divided in two semesters, two countries and different universities. The first semester, working in Portugal, at the Informatics Engineering Department of University of Coimbra with a curricular weight of 12 ECTS. The second semester, moving to The Czech Republic and work at the Department of Telecommunications Engineering of Czech Technical University in Prague with a curricular weight of 30 ECTS.

In the following sections there is a detailed description of the work plan for each semester as well as each deadline associated. Table A.1 presents the official deadlines that were followed throughout the year.

Description	Date
Official Start of Dissertation	14th of September, 2015
First Meeting	15th of September, 2015
Delivery of the Intermediate Report	22nd of January, 2016
First Public Defense	3rd of February, 2016
Delivery of the Final Report	1st of July, 2016
Final Public Defense	11th of July, 2016

Table A.1: Master Dissertation Official Calendar

First Semester

During the first semester, after the first meeting held on 15th of September, it was agreed between the three parts (student and both supervisors) that it would be held a meeting every week till the conclusion of the work. That provided a closer follow up of the work being developed. It was also defined the elaboration of meeting minutes after each meeting.

Considering the work done during the first semester, it was divided in five sections, as follows:

- State of the Art analysis (Anonymization and Concealing chapter)

  The first part of the work was to study the background of the work being developed in the area. For that, it was necessary to read relevant documents as papers, surveys, news and sections of books.

  Following a temporal line, researching the methods that were presented in the first place. Techniques and tools available, their functioning, as well as their evolution over time. The research made in this stage is described in Chapter 2.

- Study of the applicability of the method to provide privacy and confidentiality protection

  After the research for the State of the Art, it was possible to understand and start the analysis of one of the algorithms found - DNA-inspired information concealing algorithm. Verifying its advantages and disadvantages. Understanding how and over what they could be used, the type of data, the privacy granted to the data.

  In this Master Dissertation, the privatization of data and enforcement of confidentiality will be applied on different data sets. The concealing will be applied recurring to the DNA-inspired concealing algorithm (Chapter 3). It will allow the author to make an assessment about the behavior and performance of the algorithm.

- Definition of approach and validation strategy

  For the Definition of approach and validation strategy there is a detailed description in Chapter 3. The problem is defined in the referred chapter. How the experiments will be performed and how to validate the results generated by the experiments is also described.

- Definition of the detailed Work Plan for the second semester

  To define the work plan for the second semester it was necessary to decide what would be the approach to follow. The next section has a description of the work plan for the second semester, including the deviations for the initial approach planned in the first semester as well as the reasons and explanations for that.

- Writing of the Intermediate Report

  The last stage of the work plan for the first semester was the writing of the report. The supervisors proposed the 8th of January as the deadline for the delivery of the first draft of the intermediate report. Doing so, there were two weeks until the 22nd of January to make the editing of the document taking into account given recommendations or corrections.

Below is presented a Gantt Chart (Figure A.1) with the tasks assigned and scheduled for the first semester.

First Semester: Task / Weeks	September			October				November				December					January				Feb.
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21
Set up work environment in the laboratory and laptop																					
Reading suggested bibliography																					
Own reasearching and work																					
PowerPoint Presentation																					
Writing of Report																					
Preparation of First Draft																					
Editing of Report																					
Set up and prepare Presensation / Defense																					
Intermediate Defense																					

Figure A.1: Gantt chart referent to the First Semester

All the tasks and milestones scheduled for the work first semester were completed and followed without any deviations.

Following there is a section with the description of the work developed during the second semester.

Second Semester

Starting on February 8th of 2016, after the intermediate defense and moving to Prague, in The Czech Republic, the second part of the Master Dissertation was divided in three four parts:

- Implementation of the proposed approach

  The Research Approach, as proposed in Chapter 3 takes place. An experimental setup for the experiments is built. The DNA-inspired information concealing algorithm is applied on the data, results are afterwards collected and saved for posterior validation.

- Validation of the proposed approach

  In a stage where the results were gathered from the experiments, they needed to be validated. The validation of the results passed from analyzing the results obtained and verify if there were any outliers, treating, cleaning and formatting the data. It was also verified the spatial complexity and the performance of the algorithm.

- Writing the final dissertation

  The final stage of the Master Dissertation was to document all the results and findings, in order to finish and complete the final report. The final report contains all the essential and relevant information that was used in the process, as the description of the processes, the experiments, the algorithm and analysis of the results. The Final Report was delivered on the 1st of July. After that, there was one week to prepare and present the final public defense, held on 11th of July.

- Preparation of a paper

  Following the final stage of the Master Dissertation, there was the preparation of a paper gathering the results obtained in this work. The results obtained, and additional experiments with audio files will be presented there. The prepared paper will be submitted for the 2017 edition of the IEEE International Conference on Communications by Paulo Silva, Lukas Kencl and Edmundo Monteiro.

Initial plan for the Second Semester

After describing the work plan for the next stage of the Master Dissertation, it was necessary to establish a calendar to meet all the necessary deadlines. To do that, there were milestones and deliverables to follow and plan the work ahead of time. The deliverables and milestones presented below, are the ones initially scheduled during the first semester:

Deliverables:

- Prepare laboratory and experimental set up
- Apply algorithm on different data samples
- Apply on specific data - HTTP Proxy data trace
- Apply other anonymization algorithms on the data
- First draft of Final Report
- Final Report

Milestones:

- Preparation of environment - 29th Fev
- Preliminary experiments - 20th Mar
- Advanced experiments - 30th Apr
- Validation of results - 31st May
- Delivery of Draft of Final Report - 19th Jun
- Delivery of Final Report - 1st July

However, the above could not be strictly followed due to deviations that occurred during the development of the work. Due to the research nature of this work, first of the kind for the algorithm in question, new research questions and problems arose during the process.

The next section describes in detail the reasons for those deviations.

Changes and deviations from initial plan

Initially the plan was to make preliminary test with text and audio, passing later for a specific data like a HTTP Proxy data trace which is structured data. However, to make proper tests with the algorithm, which is actually capable of concealing not only structured but unstructured data, it was essential to perform tests with unstructured data like text and audio.

Inputs

Since the majority of the algorithms and techniques presented in the State of the Art are focused on the anonymization of structured data, the author concluded it was necessary to make experiments and a proper analysis of the concealing of unstructured data.

By doing that, the referred algorithms could not be compared with. One of the reasons, was the type of file accepted by the algorithms. The initially indicated ARX receives as input file types as CSV, XLS, XLSX or Database (JDBC) in a structured way.

The second tool indicated, R, accepted CSV, XLS or txt. However, the files needed to be structured (i.e. fields), so again, it would not be possible to make this kind of comparison with unstructured data as text (i.e. email contents). Even though the HTTP Proxy data trace could be treated is this way, a whole analysis of the behavior with unstructured files would be missing. As well as the audio files that could not be analyzed with the other techniques. At this point, the plan was to make the preliminary experiments and then proceed for more specific content.

Metrics

The next stage, after the concealing process, was to measure and analyze the process. In order to do that, The Shannon Entropy and Mutual Information were the metrics to use. The first obstacle in this stage was the different sizes of the files being generated by the concealing algorithm.

Since the calculation of the Mutual Information is based on calculations entropy and joint entropy, the file size was a problem. The joint entropy asks for same size vectors for the calculations and the concealed files have a bigger size than the original file.

To overcome the size difference, the smaller vector (original file) was zero padded to match the size of the concealed files. This, however, does not provide accurate results. On one hand due to the modification of the file itself, but on the other have due to the sequential properties of the Mutual Information. This meant that, for instance, shifting the start of the content on original or the concealed files, the values were different, even though the information was the same, but ordered differently.

Analogously, with the audio files, simple the sampling is higher than with text, the vectors are enormous. That turned out to be computationally hard to compute joint entropy over such long vectors, meaning long calculations ad demanding in memory. Even with the conversion of the float values, to short integer, the situation was similar.

With the above described, the author needed to apply a metric that would be able to measure the similarity of the files in a different way. By being computationally less complex, handling the ordering of the information and handling files with different sizes. For that purpose, a widely used technique in information retrieval, the Cosine Similarity was chosen. Providing a similarity metric based on the angular distance between two vectors, this approach was fit for both text and audio analysis.

Although this metric could be used, there was not available an implementation for the effect needed. The author needed to allocate time and resources to develop Python scripts that would the application of that metric on the data being tested and with the different measures (i.e. term frequency or fixed length). The implication was development of scripts, from scratch, for the cosine similarity measure using are a criterion the term frequency, fixed length of characters (or samples, in the case of audio files) or the two and three consecutive terms. Also took a considerable amount of time to develop and test the scripts.

Language of text files

Initially it was intended by the author to make tests using different languages. The tests would allow a study of language influence and differences and in concealing process and how similar the files could be in those circumstances.

The English language was used, even though the initial idea was to use also Portuguese and Czech languages. This was not possible to achieve due to encoding problems. Problems encountered with the system encoding and Matlab encoding didn't allow a correct evaluation of the output of the files generated. There were several special characters there were not being properly encoded. Consequently, not being recognized for the needed evaluation.

Several combinations of settings were experimented in order to solve the problems. However, the special characters of the Portuguese and Czech language remained a problem. Due to that situation, and the time already spent trying to fix these and the above described issues, the focus was the English language.

Algorithm's code

The experiments needed to be executed several times, with the same input file and configurations. For this work, five times. This allowed a calculation of average values and standard deviation the terms of similarity and execution time for example.

Initially, when the execution of the experiments was being made, in the validation of the generated files, it was noticed that the values were equal to all previous experiments. To understand why that was happening a thorough analysis of the algorithm's code had to be done in order to solve the situation.

The solution was to change from a fixed seed generator, to random seed generator each run of the experiments. By applying that modification, all the experiments were having different results, like they were supposed and the averages and standard deviations could be taken.

There was also research about other methods that could be used. For instance, Needleman Wunsch or Smith Waterman. However, it was not practical to implement them under the present circumstances.

Considering the above described, it was made the decision of focus the experiments over a wider range of text in English language. As well as audio experiments. Both supported by the cosine similarity metric.

Figure A.2 represents the initially defined plan. However, with the deviations suffered along the semester, there were some changes in the calendar. Figure A.3 represents the actual followed plan with updated schedule and tasks. The spaces filled with red color represent the changes in comparison with the first semester.

There is also the addition of a task referent to the article. The article is currently in development. After the delivery and defense of the Master Dissertation it will take place the conclusion and submission of an article referent to this work.

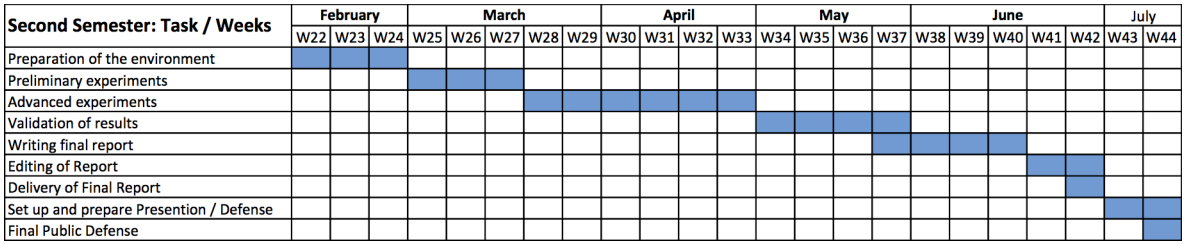


Figure A.2: Gantt chart referent to the Second Semester

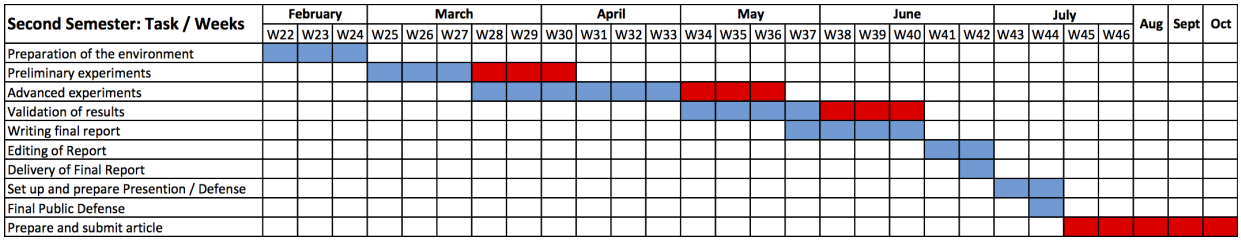


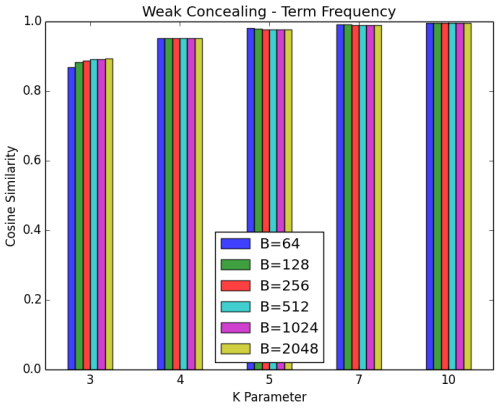
Figure A.3: Updated Gantt chart referent to the Second Semester

APPENDIX B

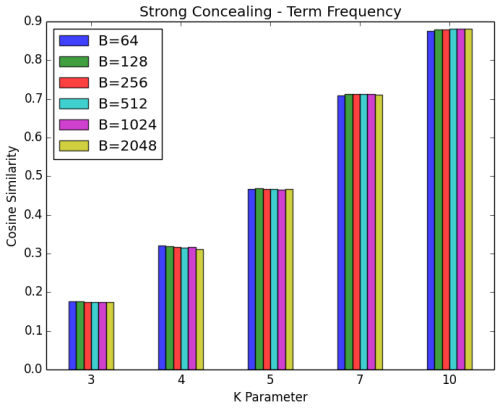
Additional Text Results

This Appendix contains additional results gathered from the experiments with text files.

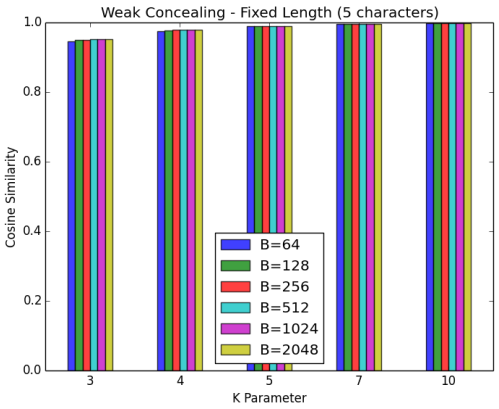
File: "Language - An Introduction to the Study of Speech" (book)



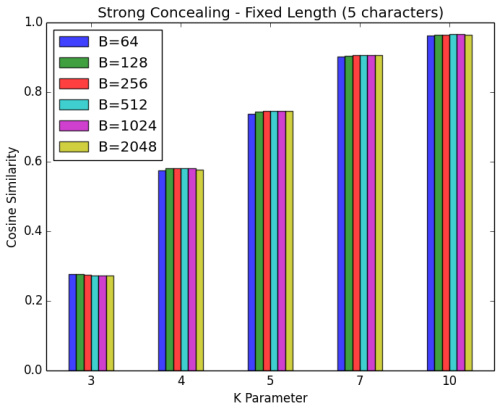
(a) Weak Concealing, Term Frequency



(b) Strong Concealing, Term Frequency

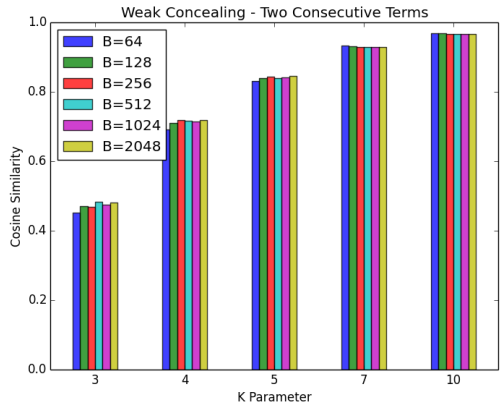


(c) Weak Concealing, Fixed Length (5 characters)

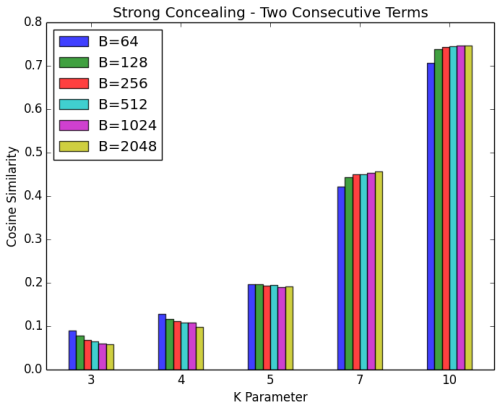


(d) Strong Concealing, Fixed Length (5 characters)

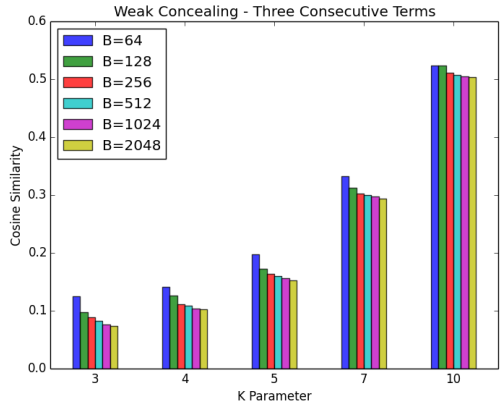
Figure B.1: Results of term frequency and fixed length - File: "Language - An Introduction to the Study of Speech" (book) - English



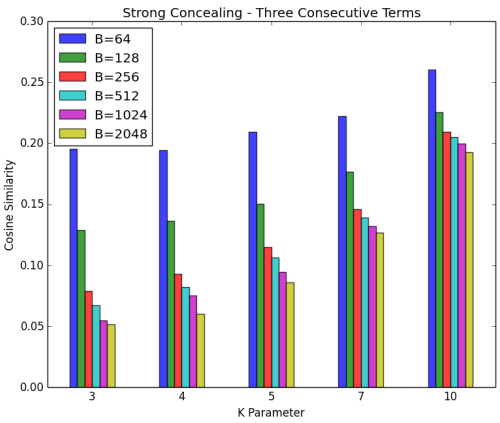
(a) Weak Concealing, Two Consecutive Terms



(b) Strong Concealing, Two Consecutive Terms



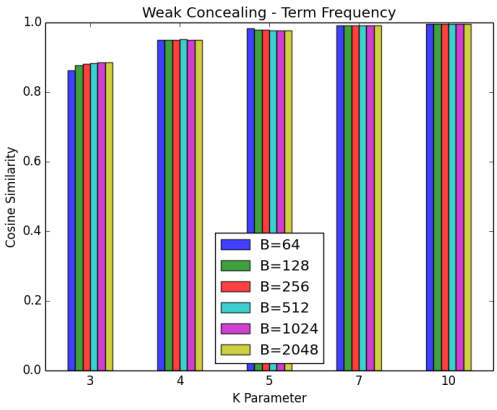
(c) Weak Concealing, Three Consecutive Terms



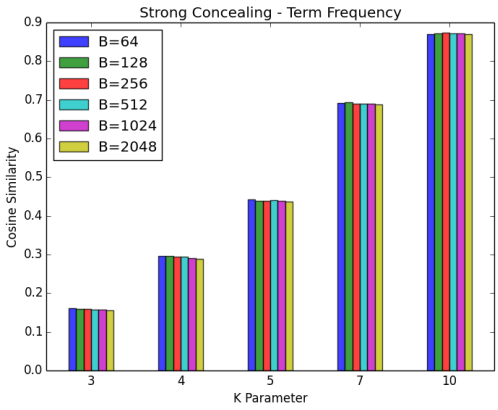
(d) Strong Concealing, Three Consecutive Terms

Figure B.2: Results of two and three consecutive terms - File: "Language - An Introduction to the Study of Speech" (book) - English

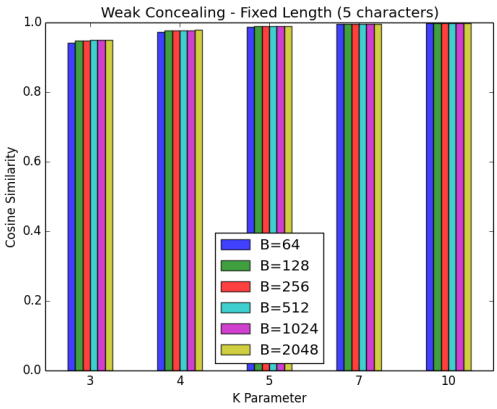
File: "Are the Planets Inhabited" (book)



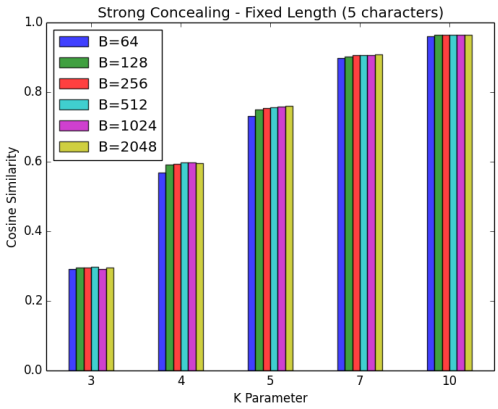
(a) Weak Concealing, Term Frequency



(b) Strong Concealing, Term Frequency

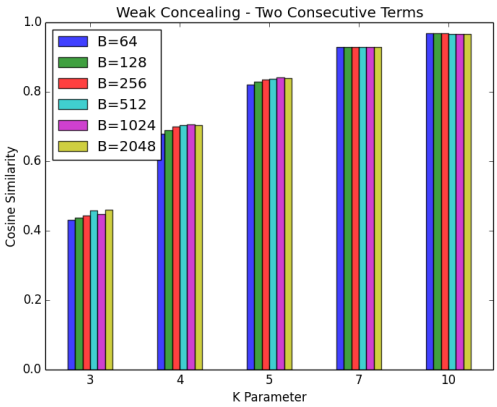


(c) Weak Concealing, Fixed Length 5 characters)

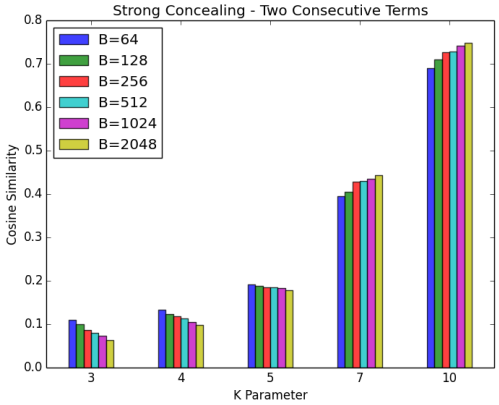


(d) Strong Concealing, Fixed Length (5 characters)

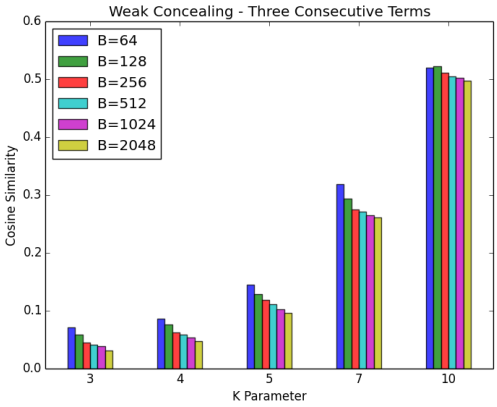
Figure B.3: Results of term frequency and fixed length - File: "Are the Planets Inhabited" (book) - English



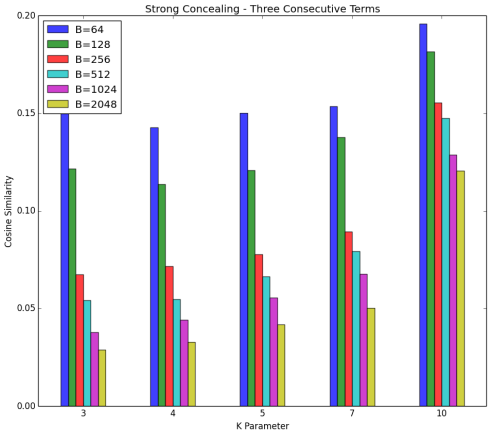
(a) Weak Concealing, Two Consecutive Terms



(b) Strong Concealing, Two Consecutive Terms



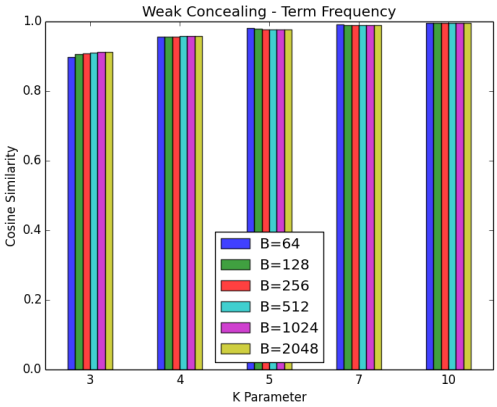
(c) Weak Concealing, Three Consecutive Terms



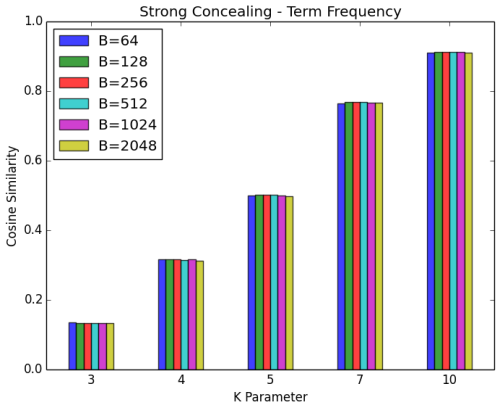
(d) Strong Concealing, Three Consecutive Terms

Figure B.4: Results of two and three consecutive terms - File: "Are the Planets Inhabited" (book) - English

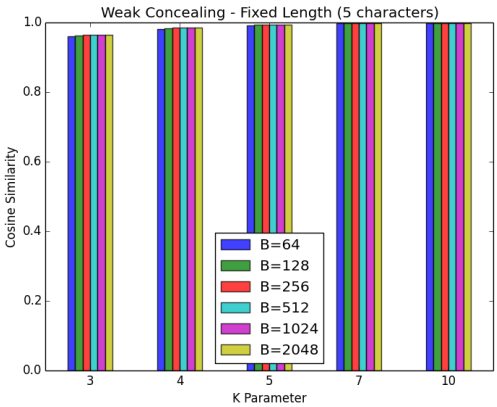
File: EU Treaty



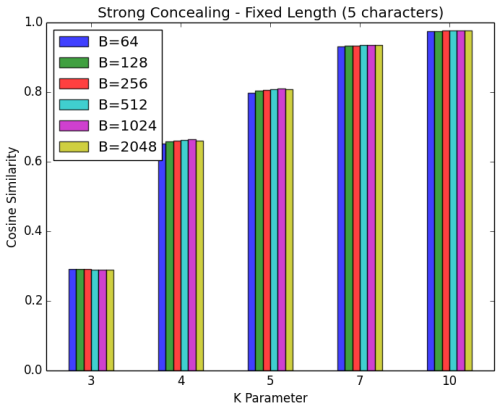
(a) Weak Concealing, Term Frequency



(b) Strong Concealing, Term Frequency

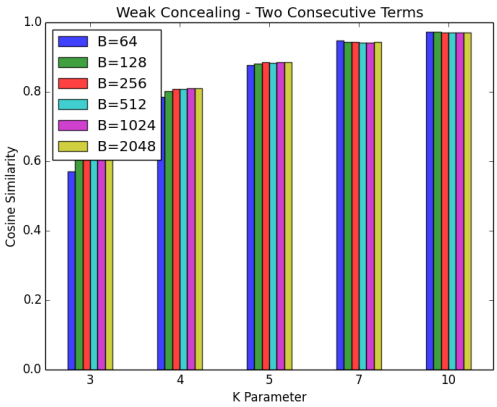


(c) Weak Concealing, Fixed Length (5 characters)

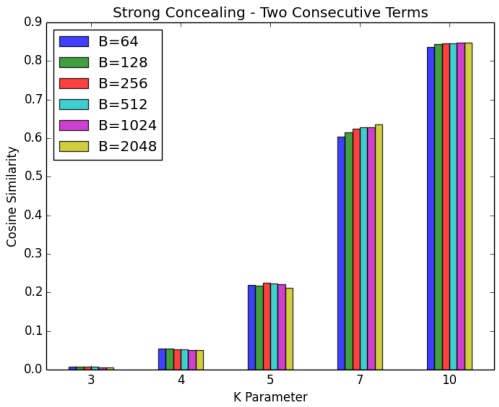


(d) Strong Concealing, Fixed Length (5 characters)

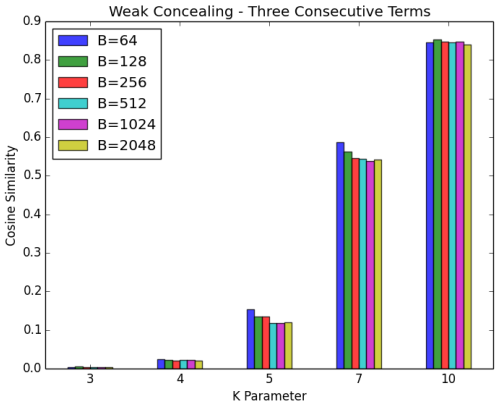
Figure B.5: Results of term frequency and fixed length - File: EU Treaty - English



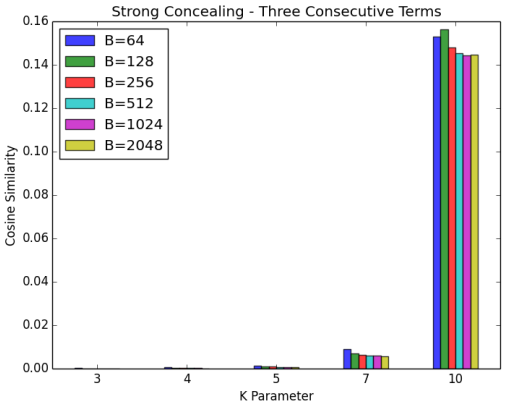
(a) Weak Concealing, Two Consecutive Terms



(b) Strong Concealing, Two Consecutive Terms



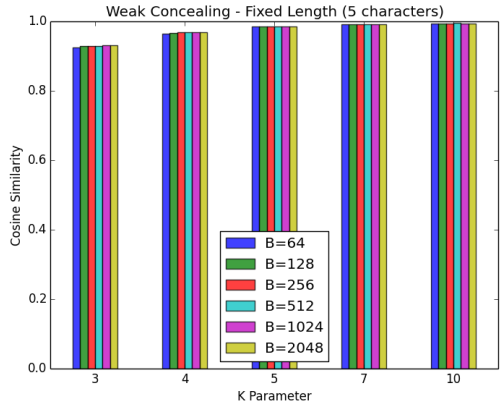
(c) Weak Concealing, Three Consecutive Terms



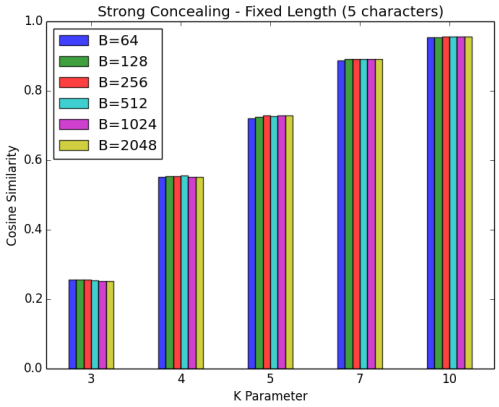
(d) Strong Concealing, Three Consecutive Terms

Figure B.6: Results of two and three consecutive terms - File: EU Treaty - English

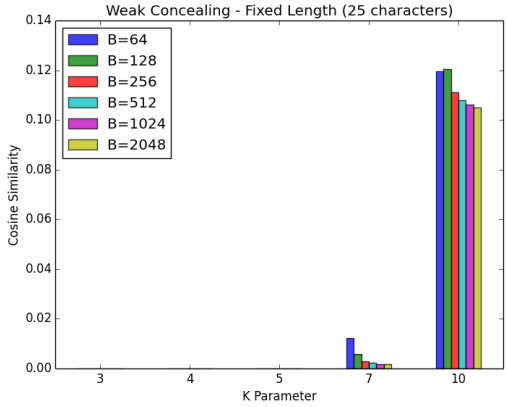
File: Hamlet



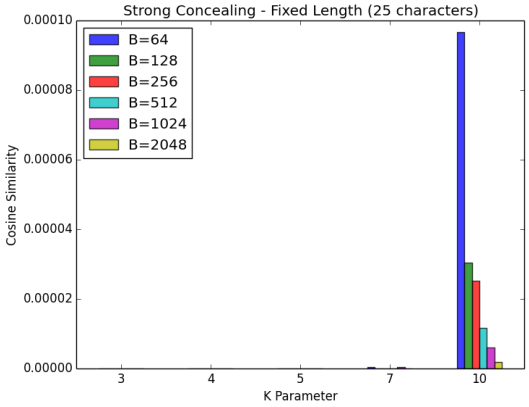
(a) Weak Concealing, Fixed Length (5 characters)



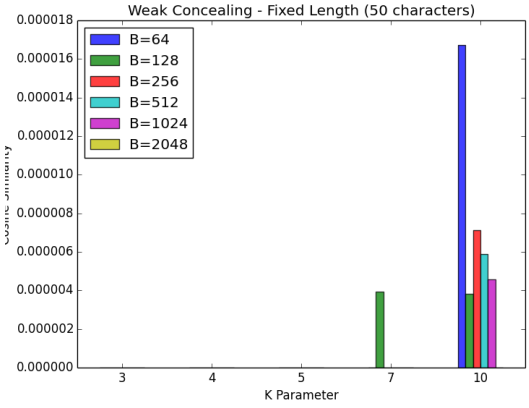
(b) Strong Concealing, Fixed Length (5 characters)



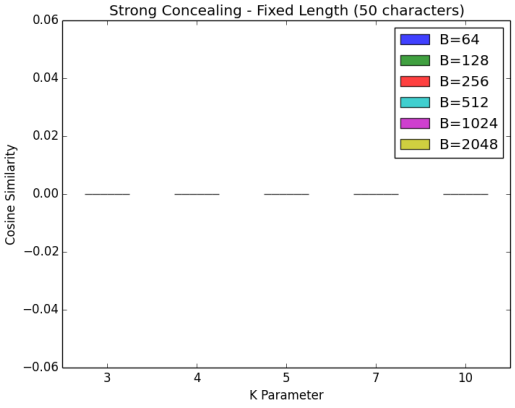
(c) Weak Concealing, Fixed Length (25 characters)



(d) Strong Concealing, Fixed Length (25 characters)

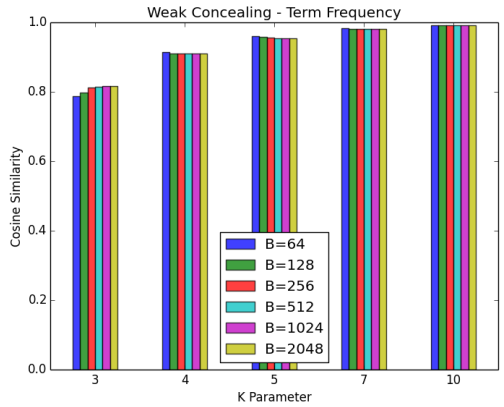


(e) Weak Concealing, Fixed Length (50 characters)

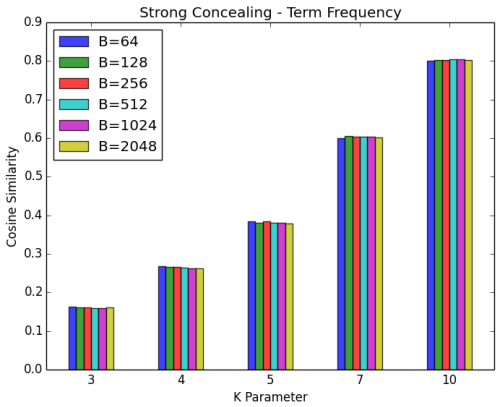


(f) Strong Concealing, Fixed Length (50 characters)

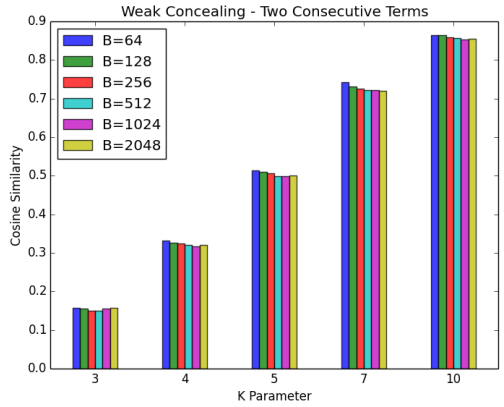
Figure B.7: Results for fixed lengths: 5, 25 and 50 - File: Hamlet - English



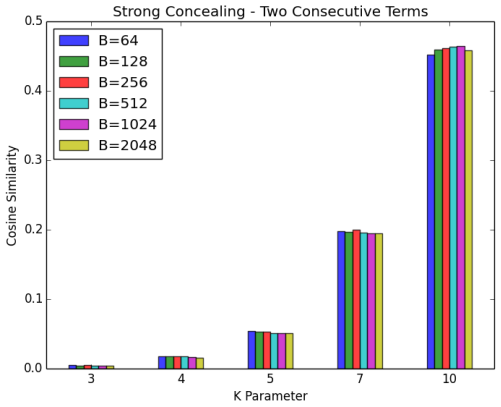
(a) Weak Concealing, Term Frequency



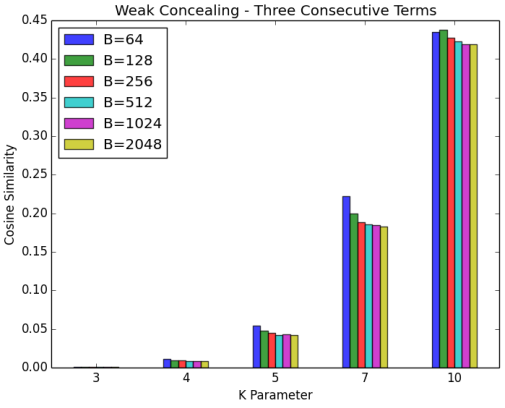
(b) Strong Concealing, Term Frequency



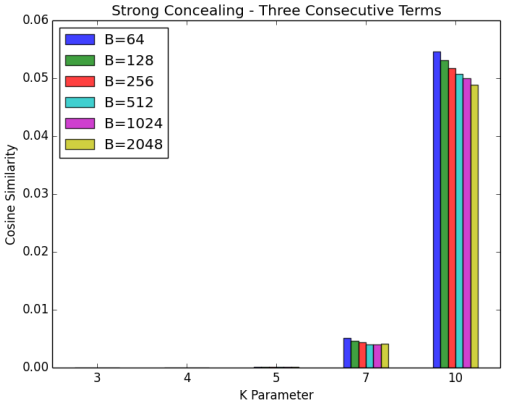
(c) Weak Concealing, Two Consecutive Terms



(d) Strong Concealing, Two Consecutive Terms



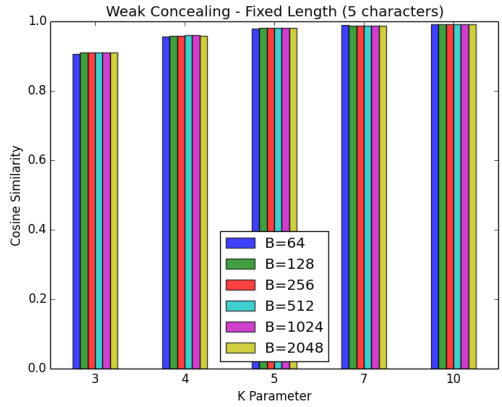
(e) Weak Concealing, Three Consecutive Terms



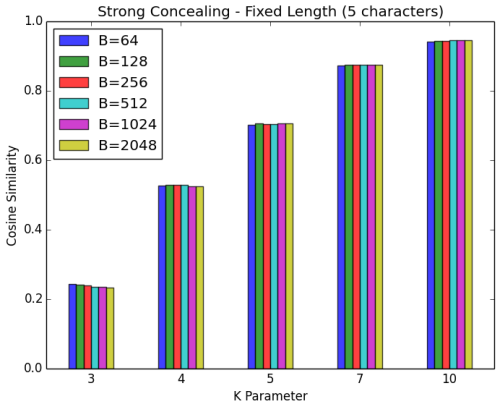
(f) Strong Concealing, Three Consecutive Terms

Figure B.8: Results of term frequency, two and three consecutive terms - File: Hamlet - English

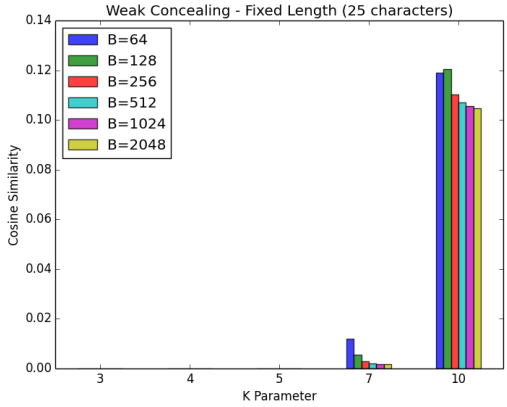
File: Macbeth



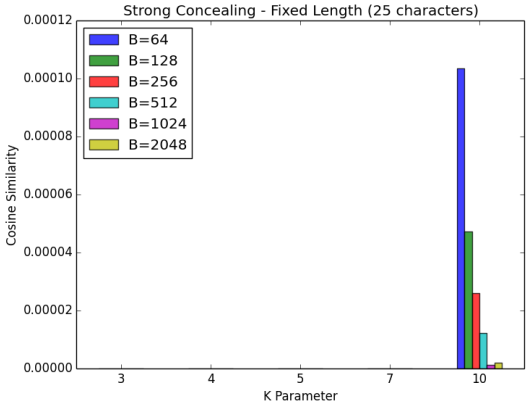
(a) Weak Concealing, Fixed Length (5 characters)



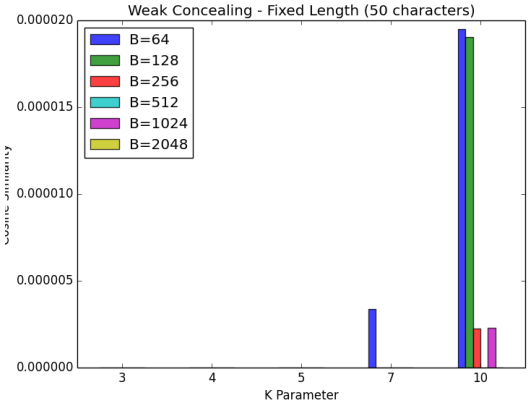
(b) Strong Concealing, Fixed Length (5 characters)



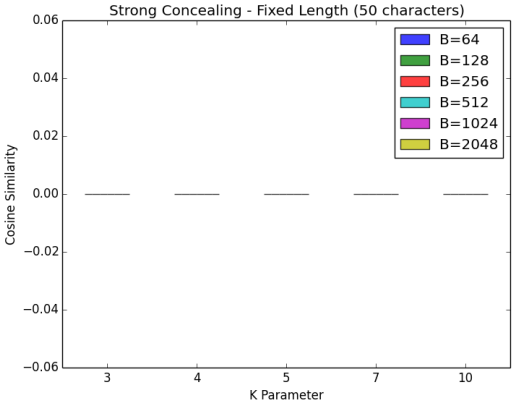
(c) Weak Concealing, Fixed Length (25 characters)



(d) Strong Concealing, Fixed Length (25 characters)

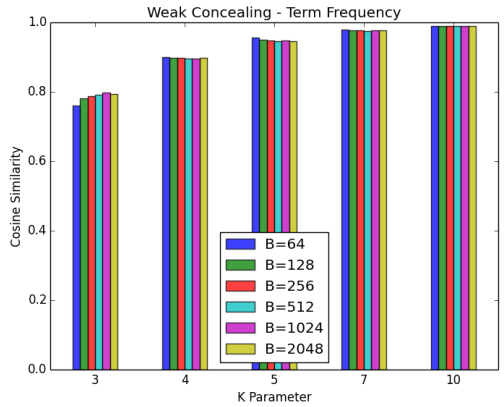


(e) Weak Concealing, Fixed Length (50 characters)

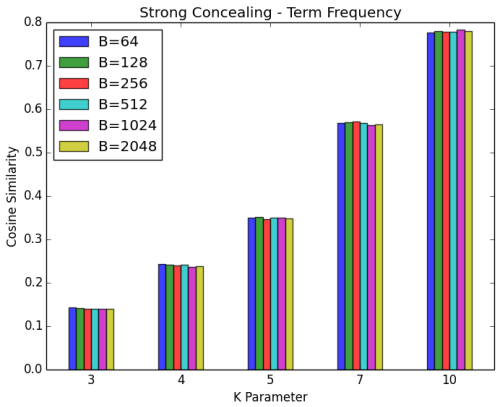


(f) Strong Concealing, Fixed Length (50 characters)

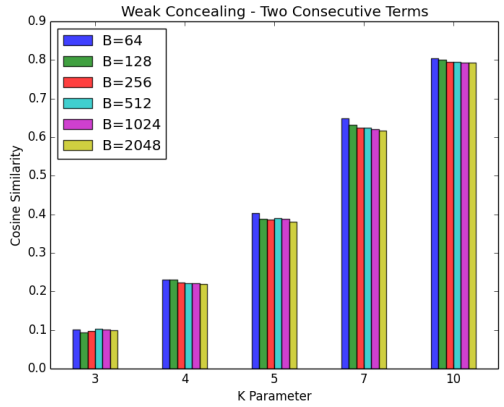
Figure B.9: Results for fixed lengths: 5, 25 and 50 - File: Macbeth - English



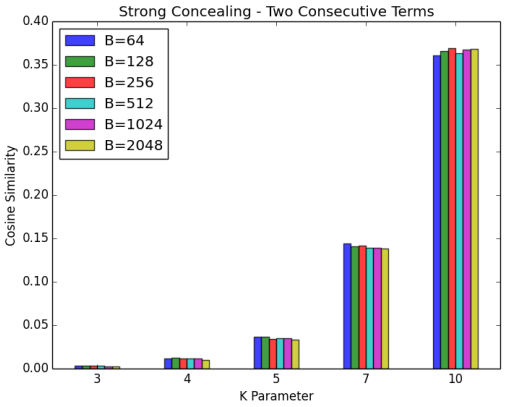
(a) Weak Concealing, Term Frequency



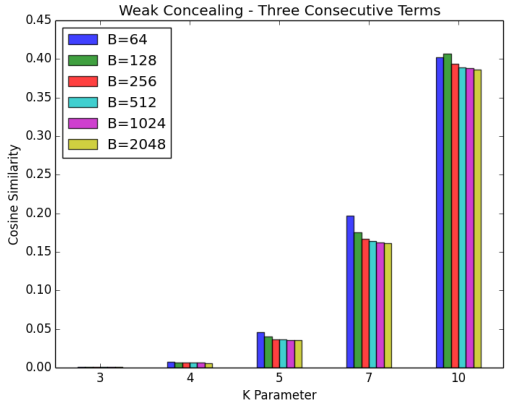
(b) Strong Concealing, Term Frequency



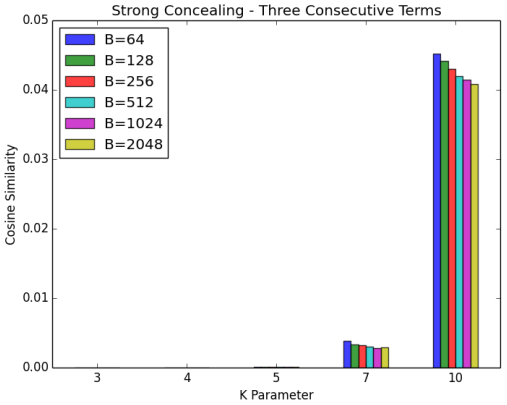
(c) Weak Concealing, Two Consecutive Terms



(d) Strong Concealing, Two Consecutive Terms



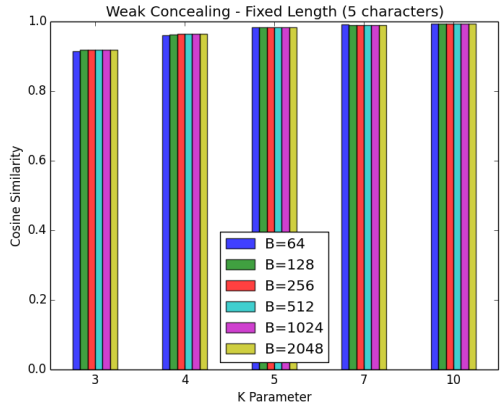
(e) Weak Concealing, Three Consecutive Terms



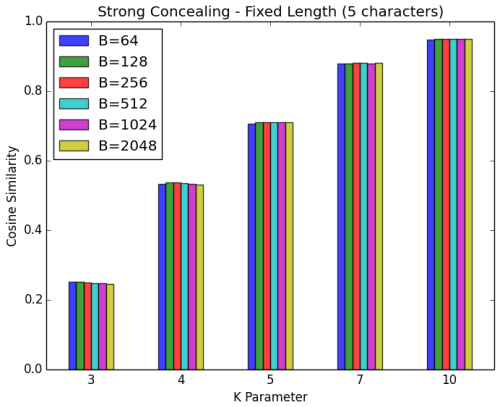
(f) Strong Concealing, Three Consecutive Terms

Figure B.10: Results of term frequency, two and three consecutive terms - File: Macbeth - English

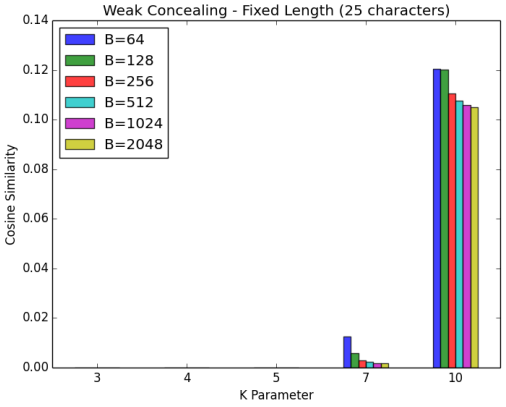
File: All's Well That Ends Well



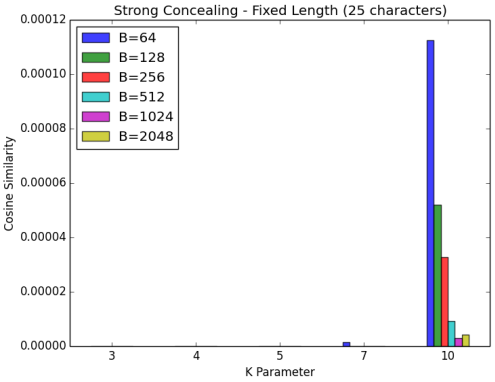
(a) Weak Conc., Fixed Length (5 characters)



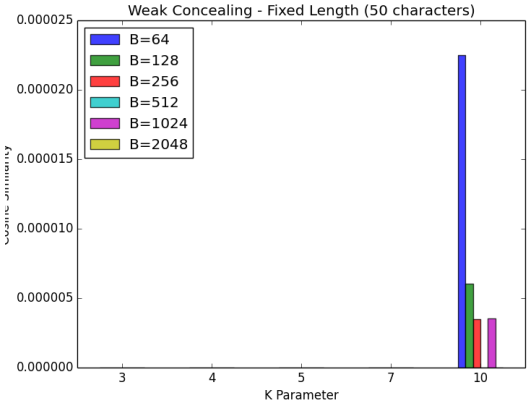
(b) Strong Conc., Fixed Length (5 characters)



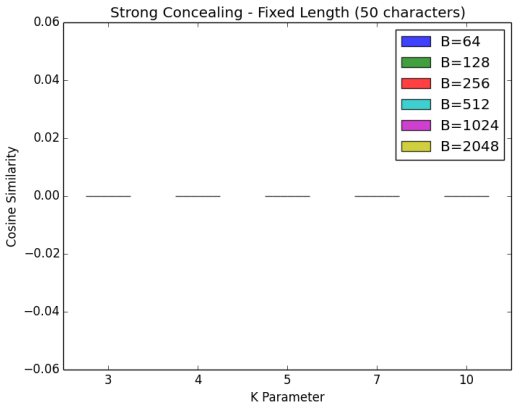
(c) Weak Conc., Fixed Length (25 characters)



(d) Strong Conc., Fixed Length (25 characters)

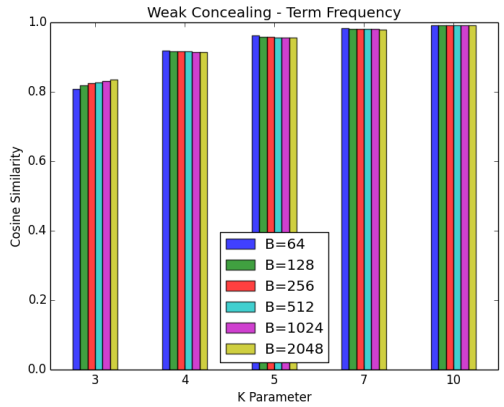


(e) Weak Conc., Fixed Length (50 characters)

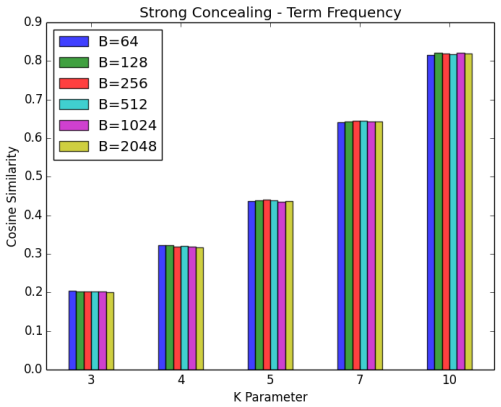


(f) Strong Conc., Fixed Length (50 characters)

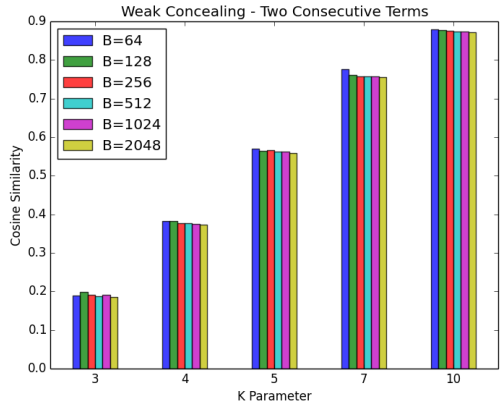
Figure B.11: Results for fixed lengths: 5, 25 and 50 - File: All's Well That Ends Well - English



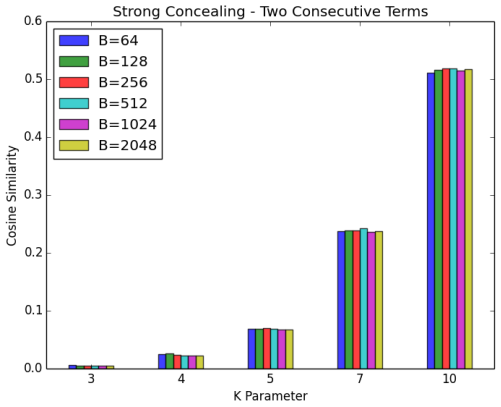
(a) Weak Concealing, Term Frequency



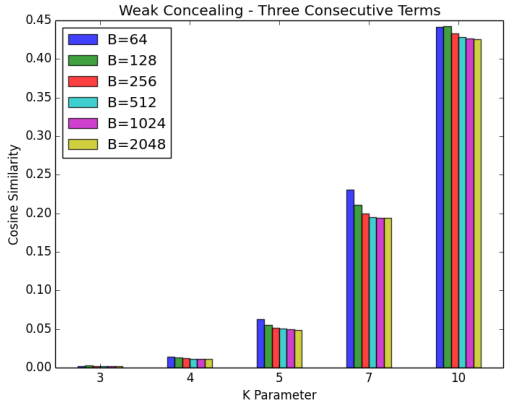
(b) Strong Concealing, Term Frequency



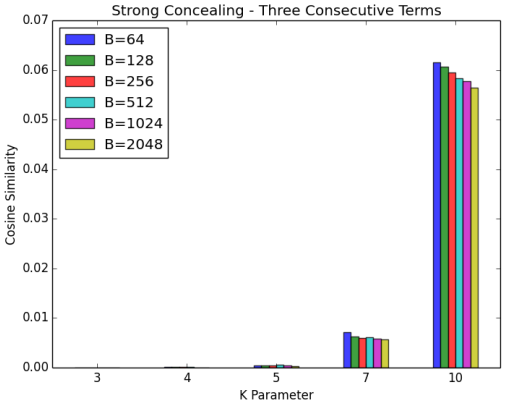
(c) Weak Concealing, Two Consecutive Terms



(d) Strong Concealing, Two Consecutive Terms



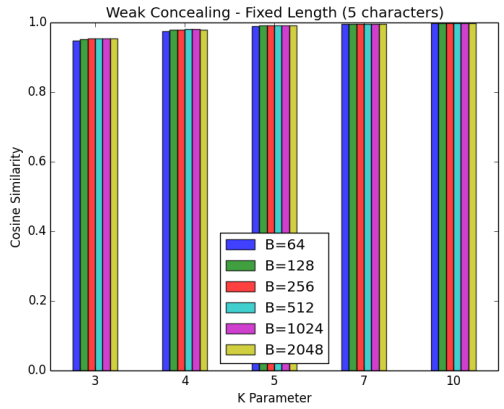
(e) Weak Concealing, Three Consecutive Terms



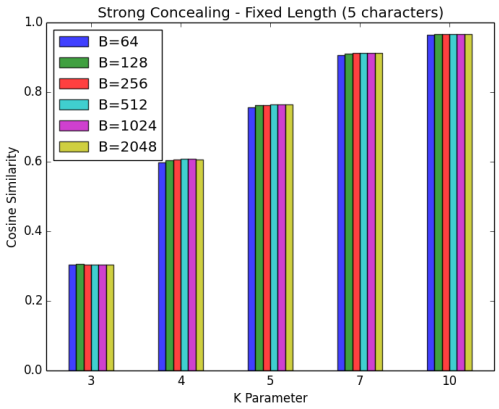
(f) Strong Concealing, Three Consecutive Terms

Figure B.12: Results of term frequency, two and three consecutive terms - File: All's Well That Ends Well - English

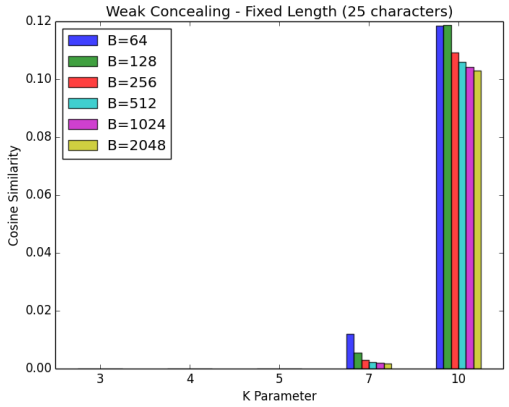
File: A Portrait of the Artist as a Young Man



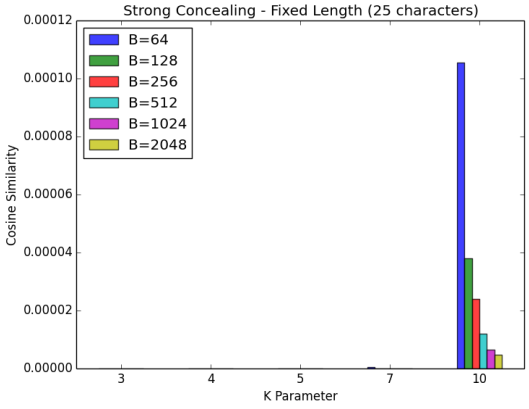
(a) Weak Conc., Fixed Length (5 characters)



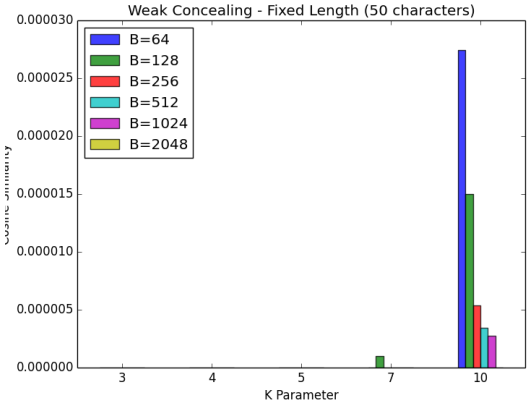
(b) Strong Conc., Fixed Length (5 characters)



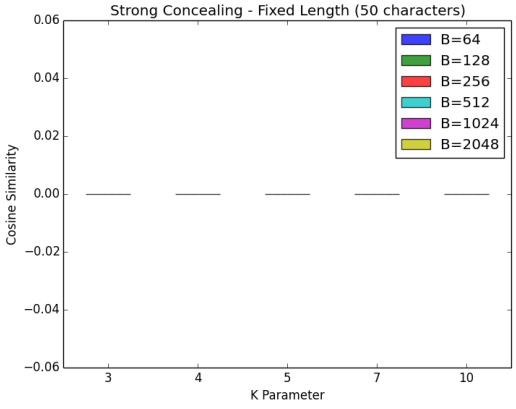
(c) Weak Conc., Fixed Length (25 characters)



(d) Strong Conc., Fixed Length (25 characters)

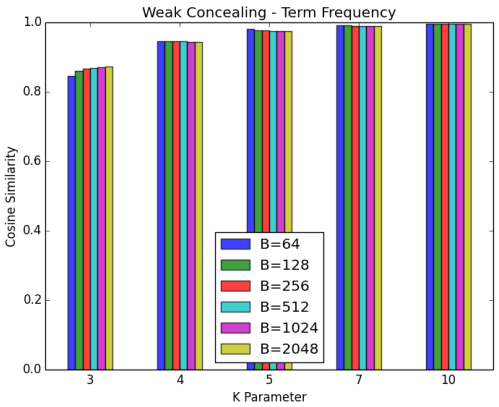


(e) Weak Conc., Fixed Length (50 characters)

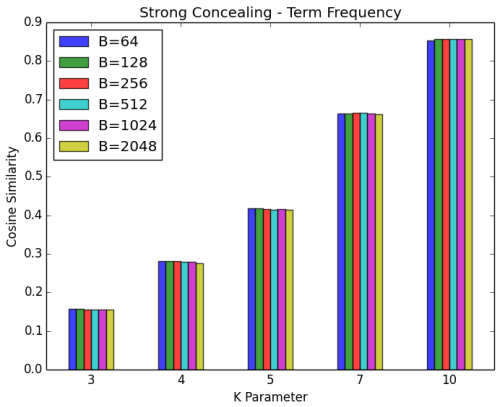


(f) Strong Conc., Fixed Length (50 characters)

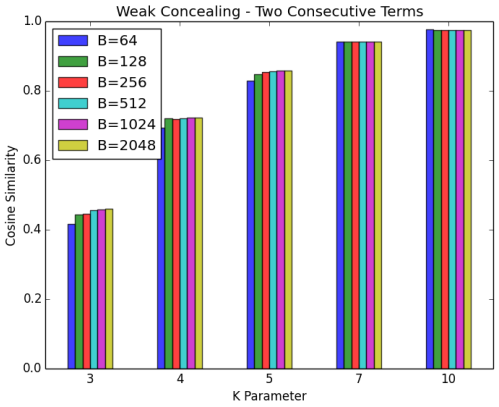
Figure B.13: Results for fixed lengths: 5, 25 and 50 - File: A Portrait of the Artist as a Young Man - English



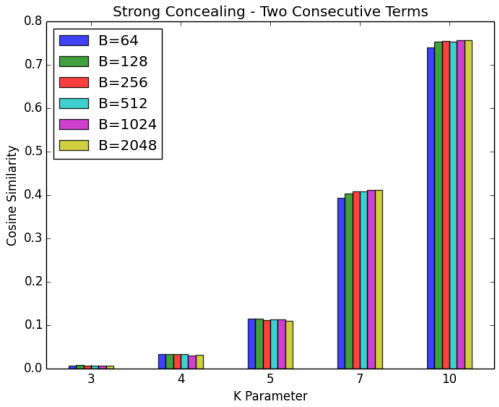
(a) Weak Concealing, Term Frequency



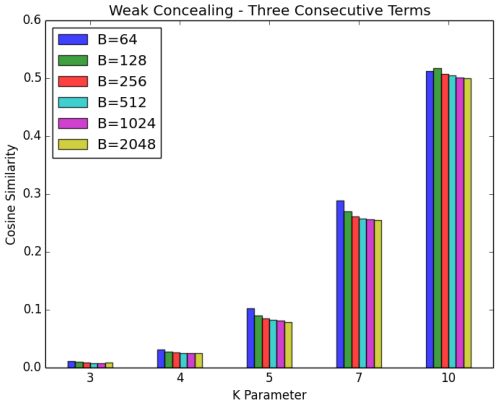
(b) Strong Concealing, Term Frequency



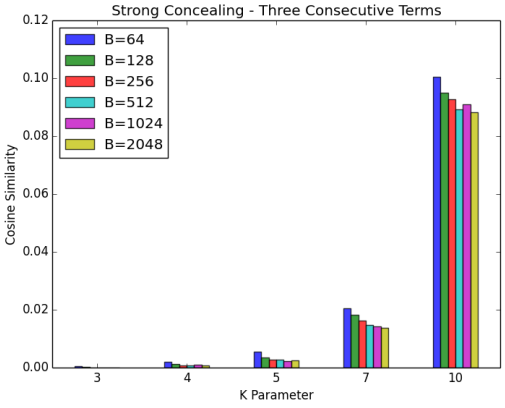
(c) Weak Concealing, Two Consecutive Terms



(d) Strong Concealing, Two Consecutive Terms



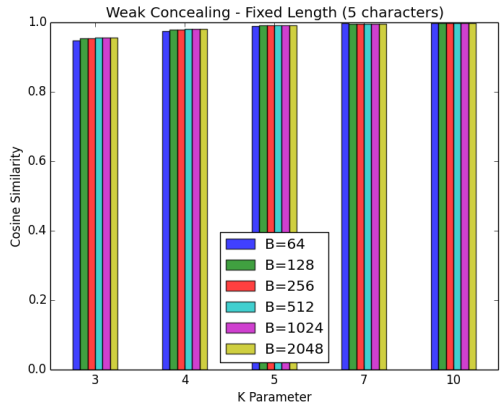
(e) Weak Concealing, Three Consecutive Terms



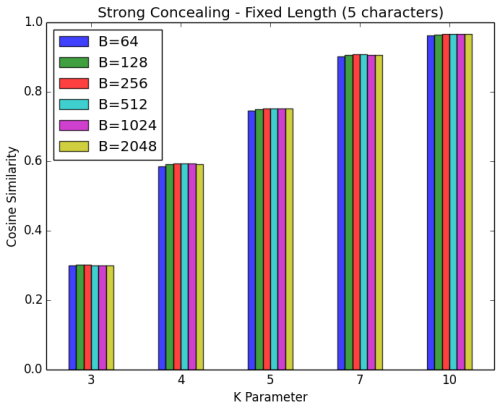
(f) Strong Concealing, Three Consecutive Terms

Figure B.14: Results of term frequency, two and three consecutive terms - File: A Portrait of the Artist as a Young Man - English

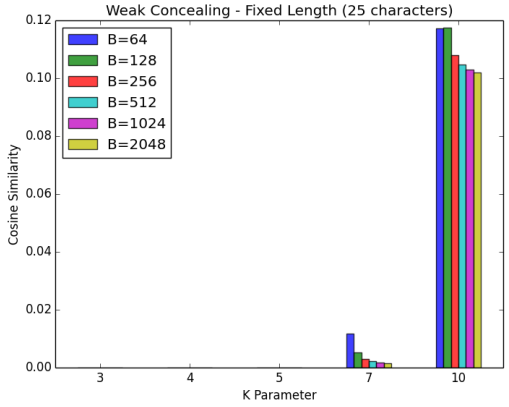
File: Ulysses



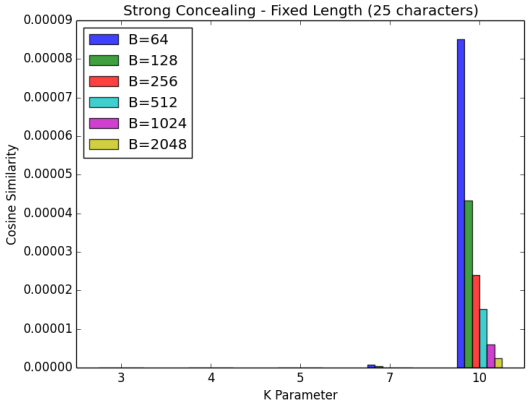
(a) Weak Concealing, ixed Length (5 characters)



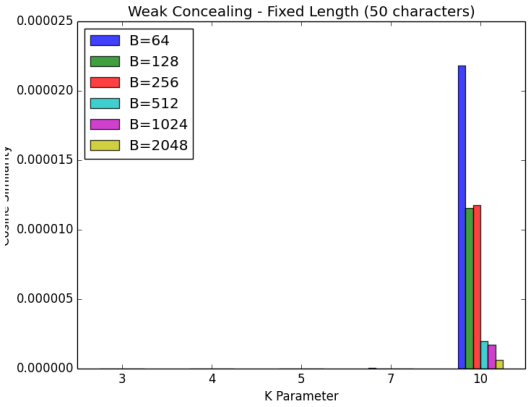
(b) Strong Concealing, Fixed Length (5 characters)



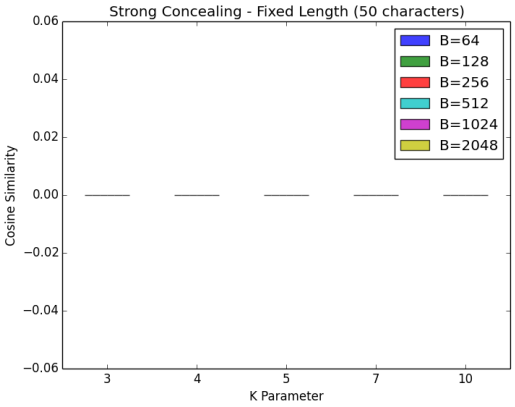
(c) Weak Concealing, Fixed Length (25 characters)



(d) Strong Concealing, Fixed Length (25 characters)

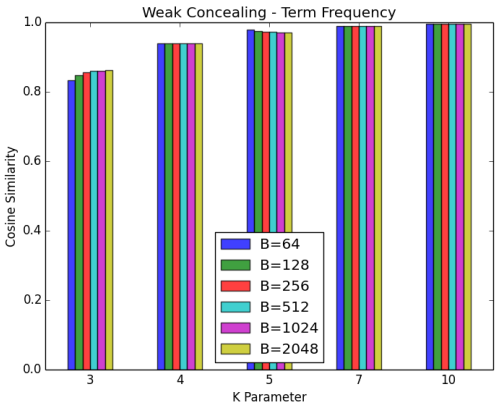


(e) Weak Concealing, Fixed Length (50 characters)

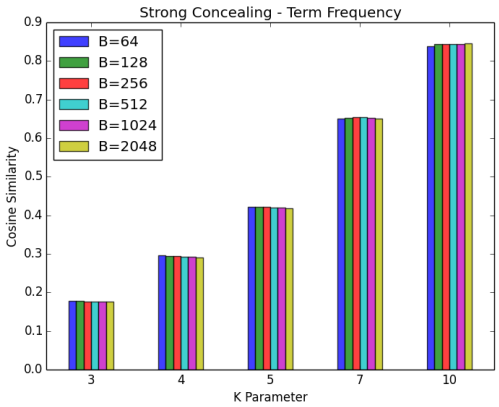


(f) Strong Concealing, Fixed Length (50 characters)

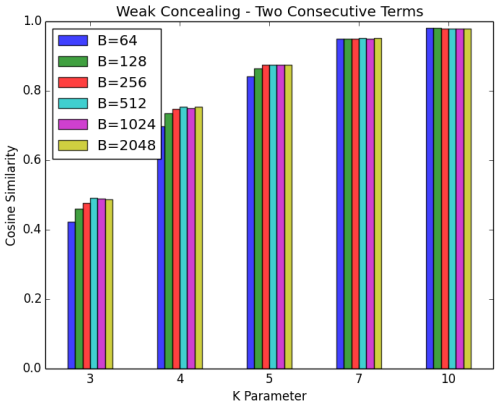
Figure B.15: Results for fixed lengths: 5, 25 and 50 - File: Ulysses - English



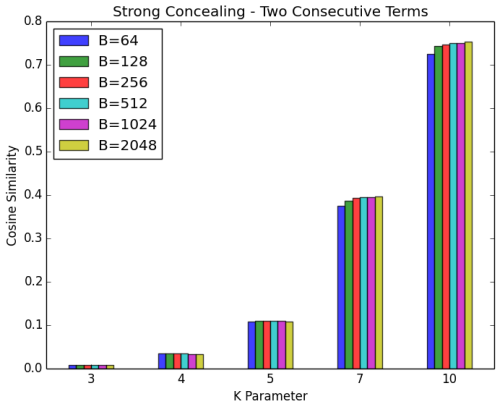
(a) Weak Concealing, Term Frequency



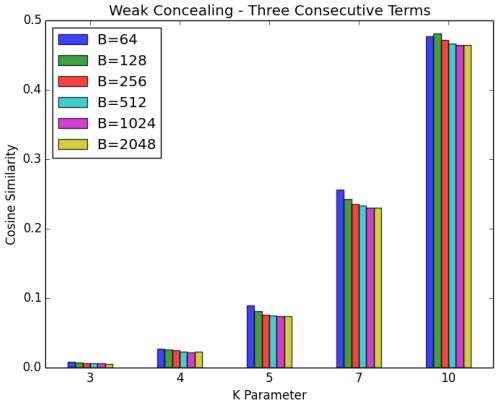
(b) Strong Concealing, Term Frequency



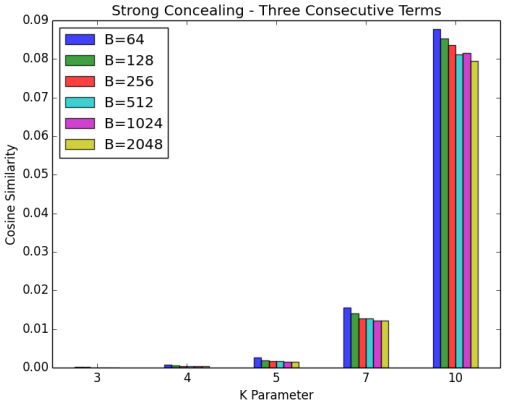
(c) Weak Concealing, Two Consecutive Terms



(d) Strong Concealing, Two Consecutive Terms



(e) Weak Concealing, Three Consecutive Terms



(f) Strong Concealing, Three Consecutive Terms

Figure B.16: Results of term frequency, two and three consecutive terms - File: Ulysses - English

APPENDIX C

Additional Audio Results

This Appendix contains additional results gathered from the experiments with audio files.

James Brown - Sound Frequency (all samples)

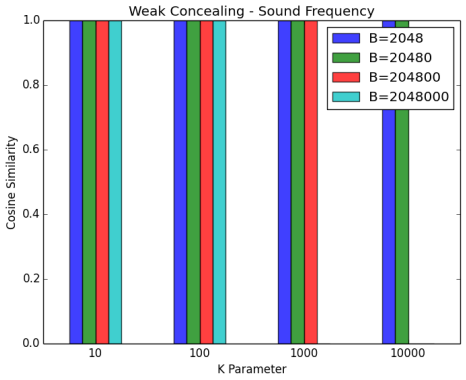


Figure C.1: Results of Weak Concealing, Sound Frequency, File: James Brown - A Mans's World

Hans Zimmer - Sound Frequency (all samples)

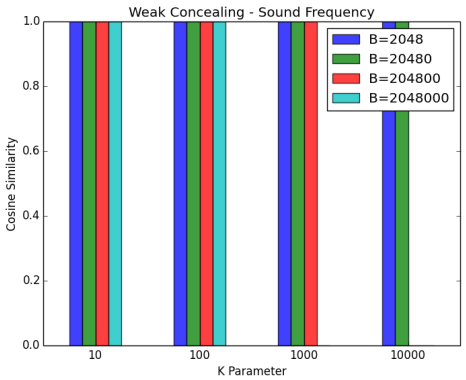


Figure C.2: Results of Weak Concealing, Sound Frequency, File: Hans Zimmer - Inception

Self Recording 1 - Sound Frequency (all samples)

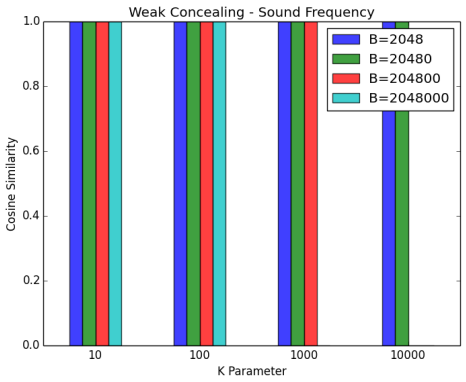


Figure C.3: Results of Weak Concealing, Sound Frequency, File: Author's Self Recording

Performance

Original File Size: 22,5 MB - David Bowie (Let's Dance)

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	65,53	68,05	68,53	68,95
Strong	540	-	-	-

Table C.1: Average file size (in MB) after application of the weak and strong concealing algorithm

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	51	12	6	5,5
Strong	1546	-	-	-

Table C.2: Average execution time (in seconds) of the application of the weak and strong concealing algorithm

Original File Size: 13,8 MB - James Brown (A Man's World)

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	39,92	41,68	41,98	42,20
Strong	331,81	-	-	-

Table C.3: Average file size (in MB) after application of the weak and strong concealing algorithm

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	32	9	5	4,5
Strong	799	-	-	-

Table C.4: Average execution time (in seconds) of the application of the weak and strong concealing algorithm

Original File Size: 30,5 MB - Hans Zimmer (Inception)

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	88,11	92	92,63	93,22
Strong	732	-	-	-

Table C.5: Average file size (in MB) after application of the weak and strong concealing algorithm

Concealing	**K10**	**K100**	**K1000**	**K10000**
Weak	73	15	7	6,5
Strong	1838	-	-	-

Table C.6: Average execution time (in seconds) of the application of the weak and strong concealing algorithm