Data Privacy Protection - Concealing Text and Audio with a DNA-inspired Algorithm

Paulo Silva[1], Lukas Kencl[2], and Edmundo Monteiro[1]

[1] Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal
{pmgsilva,edmundo}@dei.uc.pt
[2] R&D Centre for Mobile Applications (RDC), Czech Technical University in Prague, Czech Republic
lukas.kencl@rdc.cz

Abstract. Nowadays, with an increasing amount of personal and confidential data being transmitted and stored online, entities who store and manage data need to assure certain guarantees of data privacy protection. As such, we start by presenting a state of the art review of anonymization and concealing techniques. Their characteristics and capabilities are described, as well as metrics and tools to implement and evaluate data anonymization and concealing. Then, an evaluation of the applicability of the DNA-inspired information concealing algorithm is made. Usually, various metrics are used to measure aspects like disclosure risk or utility of the anonymized data. In this work, we use the Cosine Similarity metric to measure the similarity between the original data and respective versions after application of the algorithm. The evaluation is made by analyzing the output of the algorithm as well as the performance of the algorithm itself. With the final results and analysis, it is possible to determine its overall applicability with text and audio files. There is a discussion with advantages and disadvantages of this and other algorithms, as well as an identification of problems and respective suggestions for improvements on data privacy protection methods.

Keywords: Data Concealing · Data Privacy

1 Introduction

Over time different technologies and solutions are offered to fill the needs and requirements of privacy protection in IT. Solutions that range from privacy policies to security measures, authentication methods to anonymization techniques, or even laws and regulations. All play their role in the process of providing data privacy protection.

Twenty years ago, when those algorithms started to emerge, different approaches have been presented in order to provide the necessary anonymization and concealing of data. Nowadays, the scope of the methods is similar. Varies from making a complete modification of the data, making slight changes or just concealing according to certain rules. However, one aspect remains since the beginning: the more anonymized or concealed the data is, the less utility it could

provide. The opposite also applies. Ultimately, the goal of the privacy protection research field is to give data privacy protection and provide data with useful retrievable information.

This article presents the experiments performed with the DNA-inspired information concealing algorithm [1]. Most anonymization techniques and algorithms work based structured data inputs. Like network traces, logs or tables which is not the case of the DNA-inspired information concealing algorithm that conceals structured and 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 remaining methods. Therefore, the experimental work also aims to provide an analysis and evaluation of the method regarding its applicability and performance. The metric used to evaluate the results, the Cosine Similarity, shows its value not only in information retrieval or text mining applications but also in the analysis of concealed and anonymized files. Several text and audio files with different characteristics were used as an input of the algorithm. Accordingly, an analysis of the main findings and characteristics of the concealing process is presented.

Our motivation is not only to present methods and metrics but also to present actual data anonymization results and analysis. Therefore, our main objective is the evaluation of the applicability of the referred algorithm over certain data types: such as various text and audio files. Furthermore, aligned with the experimental analysis we perform an identification of open issues and problems in the field. Moreover, suggestions and possible solutions (aligned with future work) are also addressed.

The article continues with the following sections: Anonymization and Concealing Solutions (Section 2), Experimental Work (Section 3), Discussion (Section 4) and Conclusion (Section 5).

2 Anonymization and Concealing Solutions

In this section, existent anonymization and concealing algorithms are presented. Anonymization metrics and tools are equally described.

2.1 Anonymization Algorithms and Techniques

The term *anonymization* is commonly used to describe anonymization and concealing processes. There are several anonymization algorithms and techniques currently available. There is also a variation of their specifications, performance, data inputs or capabilities. Considering what was mentioned before, there are algorithms inherently more suitable than others to certain data types or applications (e.g. structured data, unstructured data, offline application, real-time, reversible or non-reversible).

The Black Marker technique [2] is a strong anonymization algorithm as a result of the replacement of fields by *NULL* or 0. Thus, it is an effective way of

hiding sensitive information. However, it does not provide good usability levels. Methods like Suppression [3] and Time Unit Annihilation [2] operate similarly.

A Permutation [3] 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 datasets, without preserving the information of the values themselves.

Hashing functions [2] can be very useful for anonymization of both text and binary data. What a hash function does is the mapping of 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 for more collisions.

Generalization [3] 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.

The K-anonymity concept [4] 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. To answer that problem, K-anonymity provides data privacy by ensuring that the sensitive attributes are repeated k times, with K being always greater than one, in order to provide confidentiality and make more difficult the identification of individual values. To achieve k-anonymity, this algorithm relies on the combination of both generalization and suppression. The MinGen algorithm [5] was developed having as a default condition the adhesion to K-Anonymity with a minimal generalization.

Differential Privacy [3] aims to provide means to maximize the accuracy of queries from a given dataset while minimizing the chances of identifying its records. There are similarities with the noise addition. In this case, anonymization is assured by the addition of Laplace noise to the queries performed on the dataset. Therefore, it is not possible to distinguish if a certain value was modified or not.

The DNA-inspired information concealing algorithm [1] (used in the experimental work (Section 3) is able to conceal information based on the introduction and maintenance of families of repeats, as DNA itself is able, at a different level. 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. To provide a solution to this problem, it was

proposed an algorithm composed of 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 at 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 and its length is related to the length of the sequences to preserve (K).

2.2 Anonymization Metrics

To assess how well the data is anonymized or concealed, there is a need to use certain metrics. Sometimes more than one at a time. The methods that can be used in this area range from common descriptive statistics for more advanced clustering algorithms. The utilization of descriptive statistics is a general but also an 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, covariance, dispersion or others, are some of the values that can be measured to quantify the distortion between anonymized and original data.

With the Classification Error Metric [3] 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 Shannon's Entropy [6] 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.

When considering Mutual Information (MI) [7] 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.

The Pearson's Correlation Coefficient [3], or correlation metric, measures the level of a linear correlation between two datasets, the original dataset 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.

Like the Euclidean Distance, the Davies Bouldin Index [8] is used to evaluate how good the clustering of data is. There are 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 the cluster which can be useful for further analysis (e.g. Euclidean Distance).

The Cosine Similarity [9] is the inner product of two vectors, divided by the product of their lengths. Also, the angle between the two vectors is represented by θ. This generates a normalized value between zero and one. The files f_1 and f_2 (or vectors) being compared have the same information if the Cosine Similarity is one. On the other hand, the files are completely different if the Cosine Similarity is zero. This is the metric used in the experimental work and can be defined as:

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

2.3 Anonymization Tools

There is quite a choice of available software and tools to make data anonymization. They work on different data types, data formats and provide different end solutions. Thereby, in this section, there will be presented solutions that offer a wider range of anonymization options and metrics as well.

ARX Data Anonymization Tool [10] is very complete due to 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.

sdcMicro [11] or 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 analysis 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 other tools as μ-Argus (which does not provide metrics), this has all the techniques plus a few more.

There are other tools available, for example, TIAMAT, UTD Anonymization Toolbox, the Cornell Anonymization Toolkit or SECRETA [12].

3 Experimental Work

The DNA-inspired information concealing algorithm was used to perform experiments over data. The objective was to conceal text and audio using a series of repetitions and permutations with the rules of the algorithm.

3.1 Methodology, Tools and Data

To perform the experiments 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. Matlab and Python scripts were used to support the experiments and data analysis.

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

Text Files To make the analysis of the generated files containing concealed information, we used the cosine similarity metric (described in section 2.1). The cosine similarity measure was taken in four different ways, for all the files. Following 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 was 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 [13]. Considering the language average length, one way of analyzing the data was by fixing a word length and verify all the sequences of characters with that specified length. Considering what was described, a fixed length of 5 characters was defined. All the sequences of characters in the original and generated 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 to 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 previous definition, 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, allows the analysis of how different the term construction and precedence is in both files being compared.

To measure the cosine similarity in such a way, several Python programs had to be developed. The developed programs started by reading both original and generated files, in the four ways previously described. After that, for each one of the four ways of analysis, the cosine similarity was manually implemented. Later, after all the calculations and file analysis, the results were gathered and treated for validation and respective analysis.

There were parameters that could be specified when running the algorithm. Besides K (length of the sequence to preserve), it was possible to choose the block size (B), which is the length of blocs to conceal and the concealing type: weak or strong. Some internal parameters of the algorithm (such as overlap, lower bound, and upper bound) depend on each other. Nonetheless, the concealing method, block size, and sequence length are independent. 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 in this case (text files).

Block Size (B) - Length of the blocks used by the algorithm. As described before (Section 2.1), 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 reasoning behind these values are variations in average email lengths (character count). 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...'

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 (section 3.1. 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 - During the experiments there were two types of concealing used. The *Weak* concealing type applies the transformations and operations to the input without adding dust and overlaps. On the other hand, the *Strong* concealing type besides providing a stronger concealing, it also prevents 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.

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 value for the sample rate is due to its proliferation. This is the most commonly used sample rate. For instance, CDs use this sample rate. 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 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 - The *Weak* and *Strong* concealing types were used 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 with such a high level of detail. The scope needs to be reduced by using sequences of samples to compare. For instance, comparing sequences of 10, 100, 1000 or 10000 samples.

Audio and text files change considerably 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.

In a first stage, the concealed files are compared with the original, using all the samples individually. Additionally, sequences of 10 and 100 samples are also analyzed by the computer, verifying the similarity with the original files. On a second stage, a perceptive analysis is made. By listening to the original and generated files with different values of K, it is determined if it is possible to recognize certain tracks of the original files in the generated files.

3.2 Results

It was found that the similarity between texts from the same author (Table 1) tends to have higher similarities than others from different authors (Table 2). It was equally found that when analyzing text with pairs of words, or even triplets, the results are more specific. The usage of this type of measure provides a way of comparison of files with specific authors or file creators. The file size was meaningful in the evaluation. The shorter the texts, the higher the difference between

Table 1. Cosine Similarity between William Shakespeare's novels

File	F. Length (5)	Term Freq.	Two Cons.	Three Cons.
A[1] vs Hamlet	0,764	0,925	0,480	0,037
A[1] vs Macbeth	0,724	0,902	0,409	0,021
Hamlet vs Macbeth	0,720	0,921	0,378	0,022
Average	0,736	0,916	0,422	0,027

Table 2. Cosine Similarity between works of different authors - James Joyce and William Shakespeare

File	F. Length (5)	Term Freq.	Two Cons.	Three Cons.
Ulysses vs Hamlet	0,718	0,862	0,454	0,027
Dubliners vs A[1]	0,652	0,781	0,346	0,016
A[2] vs Macbeth	0,688	0,844	0,315	0,013
Average	0,686	0,829	0,372	0,019

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 interesting findings. It was possible to show values of 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 Figure 1 it is possible to observe that K3, K4 and K5 present similarity values between original and concealed files not greater than the reference values taken from the comparison between not concealed 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.

Concerning the block size, as it is possible to observe in Figures 2 and 3, it was found that there were no 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.

The execution time of the strong concealing method showed to be, in average, twice as much the time as the measured in the weak concealing method. Both taking longer execution times with small values for the K parameter. One of the findings that it is not as positive as expected, is the file size. The weak concealing method produces files three times the size of the original file (which can be acceptable). However, the strong concealing type due to its inherent concealing characteristics generates files almost twenty-four times the size of

[1] All's Well That Ends Well
[2] A Portrait of the Artist as a Young Man

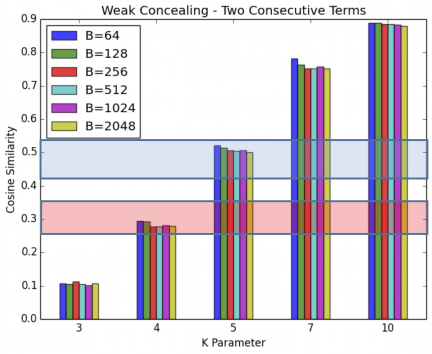

Fig. 1. Example of weak concealing method over the file Emails - Two Consecutive Terms. In the blue and pink areas, are the original (unconcealed) similarity values for files of the same author and different authors, respectively.

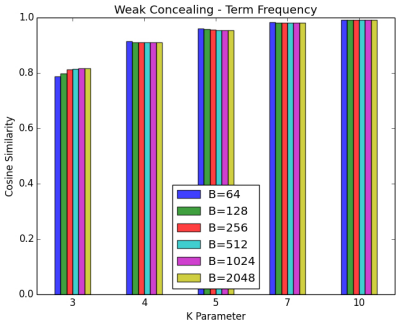

Fig. 2. Weak Concealing, Term Frequency - Cosine Similarity (Hamlet)

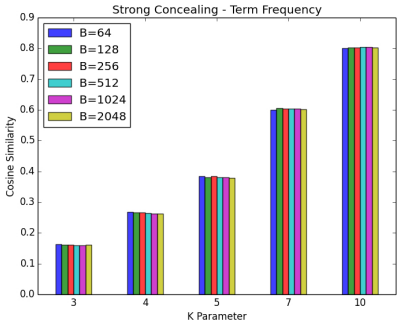

Fig. 3. Strong Concealing, Term Frequency - Cosine Similarity (Hamlet)

the original files - representing a considerable setback in terms of storage and memory demand.

Regarding the audio files, the experiments could 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-four times bigger files, for the weak and strong concealing methods, respectively.

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. On Table 3 it is possible to observe the similarity values between audio files from the same artist (considering all the samples). While on Table 4 it is possible to observe the similarity values between audio files from different artists.

Additionally, the experiments performed with different file lengths showed that the duration of the file has significance when verifying how similar two files

File	Cos. Sim.
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

Table 3. Cosine Similarity (all samples) between audio files from the same artist

File	Cos. Sim.
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

Table 4. Cosine Similarity (all samples) between audio files from different artists

are. As in the text experiments. Also, *Man's World* had its four sections similar to each other, resulting in high similarity values when comparing different sizes of the file. Considering the similarity between original and generated files, due to the larger number of samples, the values are close to 1. This happens with weak and strong concealing.

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

4 Discussion

Considering what was described and analyzed in the previous sections, there are some points to consider regarding the usage of this algorithm. In terms of advantages, the following can be considered: 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 and conceal the files using local data only. Not adding new symbols or samples to the file alphabet. However, there are disadvantages as well. For instance, generating files 3 times (using the weak concealing method) and twenty-four times (using the strong concealing method) larger than the original. This could be an obstacle is storage is limited for instance. Another disadvantage is the fact of accepting only mono audio files when nowadays most audio files are at least dual channel.

The execution time, two times longer in the strong concealing, it is not placed in either the previous groups. The reason is that it may be a disadvantage in some cases, and may not, in some other cases. For scenarios where performance is a must-have requirement (e.g. cloud environments), then it could represent a disadvantage. However, many of the final applications of these processes do not demand immediate data availability (i.e. real-time anonymized data). Therefore, the execution time would not be necessarily classified as a disadvantage.

Another interesting insight provided by this work is the potential usage of term frequency analysis to identify languages, due to the alphabet of the files. Although this was not experimented in this work, comparing the generated files with certain files would provide higher or lower values of similarities if the languages match, or not. Thereby, suggesting language differences or not.

In this article, we presented and discussed results of literature datasets like Shakespeare or James Joyce novels. Moreover, we obtained similar results with datasets such as email correspondence and scientific publications. As such, we consider that the undertaken experiments are representative enough.

4.1 Open Issues

Although there are several methods available, one should not search for a "one fits all" solution. The particular method we analyzed has its pros and cons (as most methods). Nevertheless, there are other aspects that should be considered when this or other methods are enforced. One aspect equally important is ensuring proper access control management before, during and after applying data anonymization methods on the private information. There are many specific situations that can be thought of. Nonetheless, there are two points of view that generally apply: the user's side and the entity that keeps the user's private data.

On the user's side, taking web navigation as an example, one can always try to use safer methods like Virtual Private Network (VPN) services, anonymous browsing or not providing sensitive information at all. However, there are cases whether the user needs to provide sensitive information, or it cannot control the process (e.g. medical records or voter's information). This part is where entities treating the data could act and provide privacy to data. Which is something that does not happen every time. Based on the aforementioned cases, there can be scenarios where either the infrastructure is secure, and the data is not treated in terms of anonymization processes. It could happen that in a possible point of failure of the infrastructure, data can be compromised.

A situation that can also happen, is an incorrect anonymization process. Although certain tables or files could be anonymized by an anonymization or concealing method, if an attacker can cross information from different sources (or files), it could be possible to identify sensitive fields. The process of anonymization or concealing data, based on the tools presented before, needs preparation. Certain tools can only anonymize data with specific formats or specific file types. This is a problem if data is obtained from different sources or has a different type than the ones supported for instance.

4.2 Suggestions and Possible Solutions

It could be beneficial if additional file types and formats would be supported. Not only for the algorithm analyzed in this work but for others mentioned before. Although performance evaluations were not done on other algorithms, as the trend with many services might be the migration to cloud computing services, the faster the better. An ideal scenario in these cases would be a near real-time anonymization or concealing process.

Prevention of data crossing is a point approach, but difficult to tackle. Cross-referencing data has the potential to look at several sources and data repositories with the attempt of discovering confidential information or unveiling identities for instance. Other than trying to guarantee that all the fields and information available are protected according to the amount of information publicly available, there is not much a person could do in order to avoid this situation. This is assuming that one or more fields would always be left partially disclosed (e.g. researching purposes). Reversibility is a feature that could be useful, depending on the use cases. However, is it not of as great importance as other possible solutions.

On the fly anonymization and concealing would be a very complete scenario. Suppose a solution merging both authentication systems and privacy requirements (such as anonymization or/and concealing) is offered. A user, data curator or system administrator could use its credentials to authenticate in a certain system and based on those credentials, have access to certain types of information from different confidential levels. The system (ideally) would then execute the necessary data privacy protection procedures automatically and generate data with the requested levels of privacy or confidentiality.

This type of methods has advantages when compared with encryption or denying access to data. By limiting or blocking the access to data or encrypting them, makes them unusable, for any situation. Which is not the intended situation for the academic community (e.g. researchers). Certain types of data should be made available for study considering the aspect of data privacy protection, which is what this kind of anonymization and concealing techniques do. Additionally, the supervision of a regulator about the enforcement and application of privacy policies could be beneficial in terms of data privacy protection.

5 Conclusion

Considering the importance of data privacy in our society, in this article, we focused on anonymization and concealing methods as well as actual experiments. We presented algorithms, metrics, and tools that can be used to perform data anonymization. One of the methods presented, the DNA-inspired information concealing algorithm, was used to conduct concealing experiments over text and audio files. The ability (not present in many algorithms) to conceal unstructured data demonstrated its usefulness not only by concealing a variety of text files, but also audio files. One of the main characteristics of the algorithm being analyzed

was the ability to deal with unstructured data, instead of tables, XML or CSV files. However, other than the analysis of the algorithm, it is evident there is still room for improvements and new ideas. Suggestions that can improve privacy protection were given. Including the example of a solution that could integrate an authentication system with anonymization and concealing methods. This suggestion is something that will be studied and analyzed in a future work. Along with a more comprehensive analysis of the state of the authentication systems and methods.

Acknowledgments. This work has been partially supported by the projects EUBRA-BIGSEA and ATMOSPHERE, funded by the European Commission under the EU/BR Cooperation Programme, Horizon 2020 grant agreement no 690116 and 777154.

References

1. Kencl, L. and Loebl, M.: DNA-inspired information concealing: A Survey, Computer Science Review (2010)
2. Slagell, A. and Lakkaraju, K. and Luo, K.: FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs. Large Installation System Administration Conference (2006)
3. Mivule, K. and Anderson, B.: A Study of Usability-Aware Network Trace Anonymization. Science and Information Conference (2015)
4. Sweeney, L.: K-anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzzines and Knowledge-based Systems (2002)
5. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzzines and Knowledge-based Systems (2002)
6. Shannon, C.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review (2001)
7. Kraskov, A. and Stgbauer, H. and Andrzejak, R. and Grassberger, P.: Hierarchical Clustering Based on Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)
8. Davies, D. and Bouldin, D.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence (1979)
9. Singhal, A.: Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 24.4 (2001)
10. Prasser, F. and Kohlmayer, F.: Putting Statistical Disclosure Control into Practice: The ARX Data Anonymization Tool. Medical Data Privacy Handbook (2015)
11. Templ, M. and Kowarik, A. and Meindl, B.: Statistical Disclosure Control for Micro-Data Using the R Package sdcMicro. Journal of Statistical Software (2015)
12. Poulis, G. and Gkoulalas-Divanis, A. and Loukides, G. and Skiadopoulos, S. and Tryfonopoulos, C.: SECRETA: A System for Evaluating and Comparing RElational and Transaction Anonymization algorithms. 17th International Conference on Extending Database Technology (2014)
13. Bochkarev, V. and Shevlyakova, A. V and Solovyev, V.: Average word length dynamics as an indicator of cultural changes in society. arXiv preprint arXiv:1208.6109 (2012)