An Evaluation on Robustness and Utility of Fingerprinting Schemes

Tanja Šarčević[1][0000−0003−0896−9193] and Rudolf Mayer[1][0000−0003−0424−5999]

SBA Research, Vienna, Austria
{tsarcevic, rmayer}@sba-research.org

Abstract. Fingerprinting of data is a method to embed a traceable marker into the data to identify which specific recipient a certain copy of the data set has been released to. This is crucial for releasing data sets to third parties, especially if the release involves a fee, or if the data contains sensitive information due to which further sharing and potential subsequent leaks should be discouraged and deterred from. Fingerprints generally involve distorting the data set to a certain degree, in a trade off to preserve the utility of the data versus the robustness and traceability of the fingerprint. In this paper, we will thus compare several approaches for fingerprinting for their robustness against various types of attacks, such as subset or collusion attacks. We further evaluate the effects the fingerprinting has on the utility of the datasets, specifically for Machine Learning tasks.

Keywords: Fingerprinting · Relational Databases · Data Utility

1 Introduction

An increased interest in data collection, sharing and analysis has lead to the emergence of data economies, where various stakeholders gather and store data, and others consume this data to create additional value. Data is thus on the one hand a valuable asset to its owner, and therefore any type of unauthorised distribution or usage of data by a third party, violating the owner's rights and rights of the authorised buyers, needs to be prevented. In some cases, it might be required to prove ownership of the data. On the other hand, the collected data often concerns individuals. It can either be data directly containing information about individuals, such as contact or residence information, or data about the behaviour of individuals, e.g. interaction with online resources, shopping preferences. For these situations, data leakages should be detectable, respectively attributable, i.e. it should be possible to trace the initial (authorised) receiver of a certain data set. Such a mechanism can on the one hand help in litigation cases, but on the other hand can also be a preventive measure that deters malicious behaviour, at least for some potential adversaries.

Fingerprinting techniques, which can be seen as a personalised version of generic watermarks applied to a digital object, can be utilised as a mechanism enabling ownership attribution. They generally embed a pattern in the data, i.e.

they distort the original data set to a certain extent. A good fingerprint should (i) be recognisable by the original owner of the data, (ii) not be detectable (and consequently, removable) by recipients of the data, (iii) be robust to intentional or unintentional modifications of the data, such as creating a subset, and (iv) should not lower the utility of the data too much.

The assumption in a fingerprinting scenario is that every recipient (e.g. a buyer) of the data has her own fingerprint attributed, therefore every copy that is fingerprinted and distributed by the owner is different from each other. By detecting the fingerprint within the dataset, the owner is able to detect the exact buyer of that instance of dataset.

Fingerprinting therefore usually relies on two steps: fingerprint insertion and fingerprint detection. In the first step, the fingerprint of a recipient is embedded into the dataset. Fingerprint detection then strives for detecting the fingerprint in a suspicious dataset in order to connect it with the recipient who distributed the dataset without authorisation (or is at least the first step in the chain from which the leakage originated). Fingerprint detection could be disrupted by (i) malicious attempts of the recipient to remove the fingerprint from the data, or (ii) by benign changes in the dataset, such as an well-intended sub-setting of the data, if only the subset is of relevance for a certain operation.

In this paper, we compare a number of popular fingerprinting algorithms for the above mentioned properties. We evaluate the robustness of the fingerprinting techniques towards various types of attacks by an adversary intending to disable the fingerprint. We then evaluate the effects of the fingerprint on the utility of the data by comparing the effectiveness of various machine learning models trained on both the original and the fingerprinted data sets.

The remainder of this paper is organised as follows. Section 2 discusses related work and introduces the fingerprinting schemes that we analyse. In Section 3, we describe our experiment setup and the data sets employed and, while we discuss the robustness towards attacks and the data utility aspects in our evaluation in Section 4. Finally, we provide conclusions and an outlook on future work in Section 5.

2 Related Work

Fingerprinting is, in the literature, often discussed as an extension of *watermarking*. Watermarking is an information hiding technique that allows identifying the source of digital objects by embedding secret owner-specific information into the dataset. Fingerprinting extends the functionality of watermark by providing the identification of the source of unauthorised data leakage. Fingerprint combines thus secret owner-specific and recipient-specific information embedded in a specific release of a digital object.

The concepts of fingerprinting and watermarking digital data firstly appear in domains of multimedia data and have been extensively studied over last two decades [6,16,7]. Most of these techniques were initially developed for images [15], and later extended to other modalities such as video [9] and audio [3].

Approaches for applying a watermarking scheme in other domains such as text and software have been studied as well. Techniques for watermarking text data typically exploit properties of text formatting and semantics. Watermarks are often introduced by altering the spacing between words and lines of text [14]. Other techniques rely on natural language processing and rephrasing some sentences in the text [2], thereby noticeably modifying the content, especially if more than one copy of the (differently fingerprinted) object is available.

Regarding **relational databases**, which is the focus of this work, most of the current state-of-the-art fingerprinting methods extend the watermarking technique proposed by Agrawal [1]. As mentioned above, the technique in principle contains two algorithms: watermark insertion and watermark detection.

The insertion step marks certain numerical attributes such that the least significant bits (LSBs) are altered. Thus this technique assumes that the dataset contains one or more numerical attributes. The number of LSBs available for marking is a trade-off between the robustness and imperceptibility of the mark. The insertion uses a cryptographic pseudo-random sequence generator $\mathcal{G}$, seeded by a secret key known only to the owner of the database and concatenated with the primary key attribute value of each tuple from a database. The numbers generated determine the bits to be marked, as well as the mark itself. It is computationally unfeasible to predict the next number generated by $\mathcal{G}$, thus unfeasible to guess the marking pattern without the knowledge of the owner's private key.

The detection calculates the same sequence as in the insertion algorithm, thus identifying which bits within the database should have been marked, and counts how many of them match the bits from a specific database. If the number of matches is "large", defined by a parameter called *significance level*, the database owner can suspect a leakage. The authors analysed the robustness of this technique against the number of malicious attacks: subset attacks, bit-flipping attacks, mix-and-match attack and false claim of ownership.

Li [12] extends this watermarking technique into a fingerprinting technique, by embedding different bit-strings – *fingerprints* in different releases of the data. The owner generates a fingerprint from her secret key and the recipient's identifier, using a cryptographic hash function. This way, storing a recipient-to-fingerprint pair, and entailing security management for this database, is not required. The insertion step is similar to [1], additionally embedding the generated fingerprint by an XOR function applied on the mark (called *mask*) and a selected fingerprint bit. Also the detection step is similar to [1] – it locates the bits that should have been altered and compares the matching of the extracted fingerprint with recipients fingerprints, with a τ as a parameter related to the assurance of the detection process.

In [13] a block-oriented fingerprinting scheme, inspired by a fingerprinting scheme for images from [8], is presented. In the insertion step, the LSBs of numerical values are combined into a two-dimensional matrix and separated into blocks of size $\beta \times \beta$. All blocks receive a fingerprint, the position within the block being randomly selected. The fingerprint is produced in the same manner as in [12], using the owner's secret key and the recipients's identifier as seed. If the fingerprint is shorter than the number of blocks, it might be embedded multiple times.

The detection step first tries to restore the database to be examined by filling in the original values in case of data deletion. The expected location of the fingerprint bit is computed as in the insertion step, and the bit is recorded. As the fingerprint is embedded multiple times in the dataset, if most of the detected values for a single fingerprint bit are found, the detected fingerprint is said to be found, otherwise it is regarded as not found.

The *Watermill* scheme [5,11] further considers constraints of data alteration and treating fingerprinting as an optimisation problem. By using a declarative language the usability constraints that the fingerprinted dataset must meet are specified. One of two proposed fingerprinting strategies consists of translating the weight-independent constraints into an integer linear program (ILP) and using ILP solver to solve it. The second fingerprinting strategy is *pairing heuristics* for larger datasets where using ILP solver might not be efficient.

2.1 Fingerprinting Categorical Data

All of the previously mentioned fingerprinting techniques have one restriction in common – they are applicable only on numerical attributes since they are all bit-resetting techniques. Few solutions have been proposed for categorical data. One approach is the watermarking technique presented in [17,18], which, similar to the AK scheme, uses a pseudo-random sequence generator to choose tuples for marking, and marks categorical data by changing the values to another, also pseudorandomly chosen, value from the attribute domain. One of the requirements for the technique is the presence of the primary key in the dataset, which is together with owner's secret key used as a seed for pseudo-random sequence generator. In case of multiple categorical attributes in the dataset, the technique consists of several marking iterations, one categorical attribute at a time, where in each iteration the marking pattern of some attribute is additionally controlled by adding combination of other attributes' values to the seed of pseudo-random number generator. This method prevents the attribute removal attack, but (i) increases the complexity of the marking technique, (ii) is not suitable for database relations that need frequent updates and (iii) marks are possibly overlapping because a single attribute is marked several times. The authors do not mention possibility of extending this technique to fingerprinting technique, but claim robustness against serious attacks.

Another approach is a fingerprinting technique that incorporates the k-anonymity property into the fingerprinted data [10]. k-anonymity [19] strives to modify a dataset so that at least k data samples (individuals) become indiscernible, when considering quasi-identifying attributes. This is commonly achieved by generalising values in the dataset to a broader meaning. There are generally multiple solutions of achieving the same level of k by choosing different attributes to modify. The idea in the proposed scheme is therefore to utilise these multiple, equivalent versions of the dataset as one fingerprinted version for each recipient.

K-anonymity is applied on both categorical data and numerical, therefore this fingerprinting approach can, unlike the previous schemes, operate on categorical data in the process. However, there are also several limitations: (i) the number of available fingerprints is inherently limited to the number of different equivalent versions of achieving k-anonymity, (ii) the fingerprinted copies are generally rather different from each other, and thus certain attacks might be more feasible, (iii) the utility of the differently fingerprinted (anonymised) datasets can vary significantly, and (iv), the fingerprint can not be computed alone by the recipients identifier, but rather, a mapping of fingerprint and recipients needs to be stored, with all associated security risks.

We therefore do not consider this approach in this paper. Instead, we employ a rather simple modification of the above schemes for numerical data. We first convert the categorical data to an integer representation, by simply assigning increasing integer values to each unique categorical value (a process sometimes referred to as *label encoding* in data mining settings). We can then proceed to simply applying the fingerprinting scheme by modifying the LSBs of this numerical representation. After the modification is done, we convert the label-encoded variable back to the corresponding categorical value. This process works fine as long as the number of distinct values is a multiple of 2, and thus all modified numerical values have a corresponding categorical value. For other cases, we consider passing the modified value through a *modulo* function before the transformation to a categorical value. This ensures syntactical correct values in the dataset, but introduces potential issues with detecting the fingerprint, where a different numeric value might be expected than the one resulting from the modulo function. We will study the effects of these on the data utility as well as on the robustness of the fingerprint in our evaluation.

3 Experiment Setup

In this section, we describe the datasets used in our experiment, as well as the approach for the robustness and utility evaluation.

3.1 Datasets

For the empirical evaluation, we selected two publicly available datasets. The first dataset is the so-called *Forest Cover Type* dataset, obtained from the UCI Machine Learning repository[1]. The dataset contains measurements related to

[1] https://archive.ics.uci.edu/ml/datasets/covertype

the forest cover originally obtained from US Geological Survey (USGS) and US Forest Service (USFS) data. This dataset consists of 581,012 instances, each describing a Forest Cover Type by 54 attributes, which are Integer or Binary values. The output variable to be predicted is one of seven different cover types. As binary variables can be easily treated as numerical / integer types, this dataset can thus be considered to contain numerical values only. The dataset is chosen due to its desired properties of containing multiple integer-valued attributes; further, this dataset is often used for experiments in watermarking and fingerprinting literature [1,12]. For the purpose of fingerprint insertion, one extra attribute *id* is added to serve as the primary key, since the chosen fingerprinting techniques require the presence of a primary key for fingerprint embedding. 44 out of the 54 attributes of the dataset contain binary values – to minimise the impact of the distortion introduced by the fingerprint, we use the remaining 10 integer-valued attributes for embedding.

The second dataset is the *Adult* dataset, obtained as well from the UCI Machine Learning repository[2]. This dataset contains 15 attributes in 30,162 samples (after removing samples containing missing values), where the attributes are both numerical and categorical (five continuous numerical and ten categorical). This dataset will thus be used for evaluating the effect of the simple fingerprinting technique for categorical data, as mentioned in Section 2.1. This dataset contains five categorical attributes that have a number of distinct values that is not a power of two, which is potentially problematic for our fingerprinting scheme because the marking algorithm may produce values out of the domain of categorical attribute. The algorithm in that case applies modulo function as an error correction step and may erase the mark.

3.2 Robustness Analysis

Fingerprinting schemes should be robust against different attacks that aim at preventing the correct detection of the fingerprint. Modifying, deleting and adding values to the fingerprinted data, which can be both benign updates and malicious attacks, can modify or erase the fingerprint. A robust fingerprinted scheme should make it difficult for an attacker to erase the fingerprint, to modify it in the way that an innocent recipient is indicted as a culprit, or to modify unmarked data such that a valid fingerprint is detected.

We will analyse robustness against different attacks using robustness measures proposed in [12].

- **Misattribution false hit** (fh^A): The probability of detecting an incorrect (but valid) fingerprint from fingerprinted data, i.e. a fingerprint of a different recipient.
- **False negative** (fn): The probability of not detecting the valid fingerprint from fingerprinted data.

[2] `https://archive.ics.uci.edu/ml/datasets/adult`

- **False miss** (fm): The probability of failing to detect an embedded fingerprint correctly. The *false miss rate* is the sum of the false negative and misattribution false hit rates, i.e. $fm = fh^A + fn$.
- **Misdiagnosis false hit** (fh^D): The probability of detecting a valid fingerprint from data that has not been fingerprinted. This measure differs from the others as it does not measure the success of a malicious attack or benign updates on the dataset. In contrast to the ability of the detection algorithm to detect the correct fingerprint from the pirated (and fingerprinted) data, the fingerprinting scheme may also, purely by chance, extract a valid fingerprint from unmarked data.

We will experimentally perform the following attacks to the fingerprinted data sets:

- **Subset attack** In the attempt to erase the fingerprint from the dataset, the attacker may release only a subset of tuples of a fingerprinted dataset. In our attack model, we assume the attacker selects each tuple independently with probability p to include it in the pirated dataset. We also assume no other updates on dataset are applied and no other attacks performed. As each fingerprint might be embedded multiple times in a dataset, a subset attack therefore succeeds when all embedded bits for at least one fingerprint bit are deleted.
- **Superset attack** In this attack, additional tuples to the fingerprinted data are added. This attack considers only addition of new tuples, while the original set of tuples remains unchanged. The sources of the additional tuples can be various, such as related datasets with similar attributes, artificial tuples with some semantic meaning, tuples generated from the dataset itself – or the values can be completely random. This attack can only be applied on fingerprinting schemes whose algorithms do function without the access to the original dataset (e.g. AK scheme). Otherwise it is trivial to compare the distributed dataset to the original and remove the tuples that are added by an attacker. In other cases, defending against such an attack can be helped by syntactical examination of the dataset – completely randomly generated tuples might be easy to spot. Also semantic information on the database can serve as a preliminary step in deletion of the superfluous tuples.
- **Bit-flipping attack** The attacks mentioned above to not alter the values of the original tuples – however, an attacker may change these values in attempt to destroy the fingerprint. In a bit-flipping attack, some bits are selected and flipped. The choice of the bits is assumed random, as the attacker in our threat model is defined as having no knowledge about the fingerprint insertion scheme.
- **Additive Attack** In the additive attack [1], the attacker tries to claim the ownership of a dataset by inserting an additional fingerprint in the dataset he received. The competing ownership claims can be resolved if there exists at least one bit that both the owner and the attacker have marked, each with a different value. The way to resolve the ownership claim competition is to determine which owner's marks win, i.e. which mark has overwritten

the other. The winning owner's mark was inserted later, therefore his claim of ownership is false. In case there is no overwritten mark, one approach for dealing with the false claims of ownership could be to ask both the owner and the attacker to produce the original dataset, i.e. the dataset before it was fingerprinted, and to demonstrate the presence of the fingerprint in each other's original datasets. The real owner will be able to demonstrate the presence of her fingerprint in attacker's original unlike the attacker in the owner's original.

3.3 Utility Analysis

Besides the robustness, the effect of embedding fingerprints on the data utility is of interest. Fingerprinting datasets entails introducing distortions to the values, which might have a negative impact on the utility of the data, similarly as it is the case when data sensitisation methods are applied [4] The utility of a fingerprinted dataset, for researchers, economists or other data analysts, can thus be measured by the extent to which it preserves aggregate and statistical information. A *utility metric* quantifies the utility of a modified dataset. In general, utility can be measured by two approaches. One approach is to utilise one or more quantitative measures of information loss (see [4] for an overview). As these measures do not necessarily reflect the final utility of a *machine learning model*, a second approach is to measure the effects of the fingerprinting on the quality of the analysis based on the data. In this paper, we employ both approaches.

For the measures on the data itself, we analyse the mean and variance of attributes, resp. the changes of those statistical moments introduced by the fingerprinting. We first discuss the expected behaviour on the example of the AK scheme, while the estimation is generally similar for the other schemes.

The procedure of embedding the fingerprint generally is controlled by the parameter γ, the number of attributes v, and the number of least significant bits ξ. In a dataset with η tuples, on average η/γ tuples are selected for marking, and within each of those tuples, a single bit of a single attribute is selected for marking. As the mark value is calculated as XOR of the fingerprint bit and pseudorandomly selected mask bit, the bit value will match the original value on average half of the times and therefore not lead to a change. Thus, a value of a tuple i will be selected and changed with probability $P\{L_i = 1\} = \frac{1}{2\gamma v}$. The changes in the attributes after fingerprinting, i.e. the errors introduced, are $\{\Delta_1, \Delta_2, ..., \Delta_\eta\}$, i.i.d. random variables. Each $\Delta_i, 1 \leq i \leq \eta$, is defined as $\Delta_i = L_i S_i 2^{U_i}$, where $S_i \in \{-1, 1\}$, depending whether the perturbed value is smaller or greater than the original value, both with probability 0.5, and $U_i \in \{0, 1, ..., \xi - 1\}$ is the uniformly distributed variable representing position of the marked bit.

The expected **mean** value of the changed attribute values is

$$\overline{x}' = (1/\eta)\sum_{i=1}^{\eta} x_i + \overline{\Delta} = (1/\eta)\sum_{i=1}^{\eta} x_i + (1/\eta)\sum_{i=1}^{\eta} \Delta_i$$

It can be shown that the expected mean error $\overline{\Delta}$ of a single attribute value is

$$E[\Delta_i] = \frac{1}{2}L_i 2^{U_i} - \frac{1}{2}L_i 2^{U_i} = 0, \forall i : 1 \leq i \leq \eta,$$

thus the expected error in attribute mean value after embedding the fingerprint is 0.

The expected variance of the perturbed attribute values is

$$V'_x = \frac{1}{\eta}\sum_{i=1}^{\eta}[(x_i + \Delta_i) - (\overline{x} + \overline{\Delta})]^2.$$

where the error in variance can be shown to be

$$\frac{1}{\eta}\sum_{i=1}^{\eta}(\Delta_i - \overline{\Delta})^2 + 2 * \frac{1}{\eta}\sum_{i=1}^{\eta}(x_i - \overline{x})(\Delta_i - \overline{\Delta}).$$

The expected error in computing the variance is thus given by

$$E[V_\Delta] \approx \frac{2^{2\xi}}{6\gamma v\xi}.$$

Also, we will employ the second approach, by directly using the fingerprinted dataset as an input to the machine learning model building, and evaluate the quality of the result. We approached the building of a classification model by applying several machine learning algorithms, namely k-nearest Neighbours (k-NN), Logistic Regression, and Random Forests. All classifiers are implemented in the Python sklearn package[3]. We present the resulting accuracy and F1-measure scores in the tables in Section 4.

4 Evaluation

4.1 Robustness Evaluation

Misdiagnosis false hit We briefly derive an expected value for this error for the AK scheme. Assume that the detection algorithm from the unmarked data extracts a potential fingerprint $f = (f_0, ..., f_{L-1})$, i.e. some bit string of length L. Furthermore, assuming that a single fingerprint bit f_i is extracted from the dataset multiple times, it is decided to be a single value (0 or 1) if that value is extracted more than $\tau\omega_i$, where ω_i is the number of times f_i is extracted. Due to the use of pseudo-random mask bits in this scheme, each time f_i is extracted, it will be extracted as 0 or 1 with a probability of 0.5, which is modelled as an independent Bernoulli trial. Once when the detection algorithm is done processing the dataset, the probability of the value of one fingerprint bit f_i of the extracted potential fingerprint f being 0 is $B(\lfloor\tau\omega_i\rfloor; \omega_i, 0.5)$, and the same

[3] https://scikit-learn.org/stable/ (specifically, we used version 0.20.3)

probability stands for f_i being 1. Therefore, the algorithm detects the potential fingerprint with the probability $\prod_{i=0}^{L-1} 2B(\lfloor \tau\omega_i \rfloor; \omega_i, 0.5)$. The probability that the extracted fingerprint is matching one of the N valid ones equals to choosing N bit strings out of 2^L possible ones: $N/2^L$. Now the overall misdiagnosis false hit rate is

$$fh^D = \frac{N}{2^L} \prod_{i=0}^{L-1} 2B(\lfloor \tau\omega_i \rfloor; \omega_i, 0.5)$$

The misdiagnosis false hit rate is exponentially dependant on the length of the fingerprint L. The rate can be reduced by increasing L. Table 1 shows the misdiagnosis false hit rate under different values of L and $\omega_i \approx \{100, 50\} : \forall i \in \{0, ..., L-1\}$, where $N = 100$ and $\tau = 0.5$ are fixed values. We can see that for $L \gg log(N)$ we can almost completely avoid the misdiagnosis false hit ($fh^D \simeq 0$), becoming thus an important influence on the fingerprint size to be chose.

Table 1. Misdiagnosis false hit rate for exemplary fingerprint sizes

L	8	16	32	64	128
$fh^D(\omega_i = 100)$	0.7208	0.0052	2.70×10^{-7}	7.30×10^{-16}	5.31×10^{-33}
$fh^D(\omega_i = 50)$	0.9151	0.0084	7.01×10^{-7}	4.92×10^{-15}	2.42×10^{-31}

Subset Attack For the AK Scheme, assuming that each fingerprint bit f_i is embedded ω_i times, the probability that all embedded bits for f_i are deleted is $(1-p)^{\omega_i}$. The probability that no valid fingerprint will be detected from the dataset is then

$$fm = 1 - \prod_{i=0}^{L-1} (1 - (1-p)^{\omega_i}).$$

We show empirically the success of a subset attack, with an attack performed on the Forest Cover Type dataset (where $\eta = 581,012$ and $v = 10$), using different parameter settings. The experimental results, for $L = 96$ and $\xi = 4$, are shown in Table 2, where every experiment is run 500 times. We can see from Table 2 that the results roughly match the theoretical expectation. The best rate of success have those attacks where the most of the tuples are deleted (>95%), and the percentage of fingerprinted tuples is low (γ is high). Therefore, we can argue that the AK scheme is robust against subset attacks.

It has to be considered that as few as 1% of the tuples in this example is approximately 5,810 tuples, which for the attacker might still be an acceptable amount of tuples to release without authorisation, and to perform the successful subset attack if γ is set high enough ($\gamma \geq 25$). In those cases where p' is large, γ should be set to the smaller value, since the probability for a successful subset attack decreases when γ decreases for the same p'.

Table 2. Experimental results of subset attack success against the AK scheme, on the Forest Cover Type dataset

	$p' = 70\%$	$p' = 80\%$	$p' = 90\%$	$p' = 95\%$	$p' = 99\%$
$\gamma = 6$	0	0	0	0	0.004
$\gamma = 12$	0	0	0	0	0.5
$\gamma = 25$	0	0	0	0	1.0
$\gamma = 50$	0	0	0.002	0.194	1.0
$\gamma = 100$	0	0	0.20	0.9975	1.0

For e.g. the *block scheme* algorithm, it is crucial to have the same number of tuples and attributes, and their right sequence, in the suspicious database to be able to detect a valid fingerprint. When the attacker removes tuples, the detection scheme first has to replace these with the corresponding ones from the original dataset. In general, for this scheme the number of tuples to be removed is much smaller – with half of the dataset still available, the success rate for large values of γ reaches values comparable to the best chance presented for the AK scheme. Theoretical success of the subset attack against the block scheme is shown in Table 3.

Table 3. The probability of a successful subset attack in block scheme

	$p' = 30\%$	$p' = 40\%$	$p' = 45\%$	$p' = 50\%$
$\beta = 5$	0	0	0	1.0
$\beta = 10$	0	0	0.001	1.0
$\beta = 15$	0	6.8233×10^{-7}	0.2320	1.0
$\beta = 20$	0	9.7949×10^{-4}	0.8301	1.0
$\beta = 30$	2.0832×10^{-7}	0.2151	0.9998	1.0

The *extended AK scheme* for categorical data described in Section 2.1 differs from original AK scheme in an additional step in the fingerprinting embedding for categorical values. As mentioned before, we trade the strength of detection algorithm for fingerprinting categorical data successfully, as the additional operations in the fingerprint insertion phase cause errors in the detection phase that cannot be avoided. Having errors in unaffected fingerprinting scheme increases also the vulnerability of the scheme to attacks. To show this, we conducted experiments are on Adult dataset, which contains categorical data. We measure the success of a subset attack on the extended AK scheme over 500 runs and parameters set as follows: $L = 80$, $\xi = 1$, $\tau = 0.5$, $\gamma = \{3, 6, 12, 25, 50, 100\}$ and $p' = \{0.30, 0.60, 0.80, 0.90, 0.95, 0.99\}$, where p' represents the percentage of tuples that are deleted. The results are shown in Table 4.

Even though the detection algorithm is able to detect the correct fingerprint from the full set of tuples, the errors introduced by the modulo operation are enhancing the success of the attack. For a comparison, the results attack success results when no error correction step has been applied, are given in Table 5.

Table 4. Experimental results of subset attack success, on the Adult dataset

	$p' = 30\%$	$p' = 60\%$	$p' = 80\%$	$p' = 90\%$	$p' = 95\%$	$p' = 99\%$
$\gamma = 3$	0.0	0.0	0.0	0.004	0.22	1.0
$\gamma = 6$	0.08	0.18	0.20	0.354	0.954	1.0
$\gamma = 12$	0.078	0.0	0.212	0.97	1.0	1.0
$\gamma = 25$	0.012	0.284	0.99	1.0	1.0	1.0
$\gamma = 50$	0.346	1.0	1.0	1.0	1.0	1.0
$\gamma = 100$	0.976	1.0	1.0	1.0	1.0	1.0

In this experiment, the fingerprint is embedded only in numerical values of the Adult dataset, otherwise using the same scheme. If an error correction step is being applied, the attack success rate is generally higher. Only for small values of γ, and if not a large portion of tuples are deleted, the scheme is robust to subset attacks.

Table 5. Experimental results of subset attack success for the case where fingerprint is marking only numerical values, on the Adult dataset

	$p' = 30\%$	$p' = 60\%$	$p' = 80\%$	$p' = 90\%$	$p' = 95\%$	$p' = 99\%$
$\gamma = 3$	0.0	0.0	0.0	0.0	0.07	1.0
$\gamma = 6$	0.0	0.0	0.0	0.0	0.11	0.98
$\gamma = 12$	0.0	0.0	0.16	0.97	1.0	1.0
$\gamma = 25$	0.0	0.11	0.98	1.0	1.0	1.0
$\gamma = 50$	0.15	0.98	1.0	1.0	1.0	1.0
$\gamma = 100$	0.97	1.0	1.0	1.0	1.0	1.0

Bit-flipping attack As an example, for the Block scheme, we assume that the attacker examines every bit available for fingerprinting independently and selects it for flipping with probability p. Let us approximate the number of times that each fingerprint bit is embedded in the data to ω. For the detection algorithm to fail to recover the correct fingerprint bit, at least $(1-\tau)\omega$ embedded bits corresponding to the single fingerprint bit f_i must be changed, i.e. more than $\omega - \lceil \tau\omega \rceil + 1$ bits must be changed. The probability that one fingerprint bit is destroyed is $B(\omega - \lceil \tau\omega \rceil + 1; \omega, p)$. The probability that the entire fingerprint will be detected incorrectly is therefore

$$fm = 1 - (1 - B(\omega - \lceil \tau\omega \rceil + 1; \omega, p)^L).$$

We run experiments on the Forest dataset both for Block scheme and AK scheme. Table 6 shows the obtained empirical results for the success of the bit-flipping attack on the block scheme where each experiment is run 100 times, while Table 7 shows the results for the AK scheme.

We can observe that the number of bits to be flipped needs to be rather high - more than 30% of the bits available for fingerprinting, to achieve an attack

Table 6. Experimental results of the bit-flipping attack on the Block scheme, for the Forest Cover Type data

	p=30%	p=40%	p=45%	p=50%
$\beta = 5$	0	0	0.50	1.0
$\beta = 10$	0	0.50	0.50	1.0
$\beta = 15$	0	0.50	0.92	1.0
$\beta = 20$	0.08	0.50	1.0	1.0

Table 7. Experimental results of the bit-flipping attack on the AK scheme, for the Forest Cover Type Data

	$p = 20\%$	$p = 30\%$	$p = 40\%$	$p = 45\%$
$\gamma = 6$	0	0	0.50	0.56
$\gamma = 12$	0	0	0.50	1.0
$\gamma = 25$	0	0	0.54	1.0
$\gamma = 50$	0	0.50	0.72	1.0
$\gamma = 100$	0	0.36	1.0	1.0

with a certain guarantee of success. Such a large modification is expected to render the utility of the dataset obtain rather low. Choosing smaller β for the Block scheme or γ for the AK scheme contributes to better robustness against bit-flipping attack.

Additive attack We consider a scenario where the attacker tries to claim the ownership of the dataset by inserting an additional fingerprint in the received dataset. The competing ownership claims can be resolved if there exists at least one bit that both the owner and the attacker have marked, each with a different value. In that case it is possible to decide which mark appeared later, "on top of the other". In all of the considered techniques it is justified to conclude that the odds of finding such conflicting bits are low, unfortunately for the owner.

Let us take AK Scheme as an example. Suppose that the data fingerprinted by the owner is marked ω times with parameters γ, v and ξ and that the attacker performs the fingerprinting insertion algorithm with parameters γ', v' and ξ'. Under the usual probabilistic model of AK scheme's bit-marking process, the probability that a specified bit marked by original fingerprint is also marked by the attacker is the product of probabilities that the tuple containing the bit is chosen for marking ($1/\gamma'$), that the attribute containing the bit is also chosen for marking ($1/v'$) and that the specified bit is chosen ($1/\xi'$). The probability that the attacker's mark is different from the original mark is 1/2, so that the overall probability that the specified bit is a conflict bit is $1/(2\gamma' v' \xi')$. The tuples are marked independently of each other, therefore the probability that the attack is successful, i.e. no conflicting bits are found, is

$$P\{success|\omega\} = (1 - \frac{1}{2\gamma' v' \xi'})^{\omega}.$$

For example, let the dataset have 500,000 tuples and let $\omega = 1000$. Assume that attacker wants to increase his chances of success, i.e. minimise the likelihood to overwrite an existing fingerprinted bit, thus she sets $\gamma' = 10,000$ (a rather large value, considering this means that only $1/10,000$ tuples will be marked), $v' = 10$ and $\xi' = 5$, then $P\{success|\omega\} = (1 - 10^{-6})^{1000} \approx 0.999$.

4.2 Utility

Utility measured on the Data For the utility evaluation on the data directly, we discuss the results of applying the AK scheme on the Forest Cover Type dataset. We choose a set of values for the parameters, specifically $\gamma = \{12, 25, 50, 100\}$, and $\xi = \{4, 8\}$. Table 8 contains recorded changes in the variance introduced by fingerprinting for each of the attributes and parameter setting. These measured values support the analysis previously made on errors in mean and variance of the attribute values in Section 3.3.

The error in the mean in all of the cases of this experiment was zero or very close to zero, thus only the error in the variance is presented in the table. The largest changes are, as expected, occurring when γ is small and ξ is big, i.e. in the cases where more tuples are selected and more bits of a value are available for marking. The errors in variance between cases with the same γ value and different xi vary noticeable, implying that the imperceptibility of the fingerprint is sensitive to the number of LSBs available for marking. The magnitude of the unperturbed values of the variances in general does not affect the relative error of the perturbed counterparts. The only exception is the attribute "HD-Roadways" with large original values for both mean and variance.

Table 8. Change in variance introduced by the AK fingerprinting scheme, on the Forest Cover Type dataset

		γ	100		50		25		12	
		ξ	4	8	4	8	4	8	4	8
Attribute	**Mean**	**Variance**								
Elevation	2,959	78,391	0	+1	0	+1	+1	+5	+1	+9
Aspect	156	12,525	0	+1	0	+1	+1	+5	0	+8
Slope	14	56	0	+1	0	+3	0	+5	0	+11
HD-Hydrology	269	45,177	0	+1	0	+1	0	+2	+1	+2
VD-Hydrology	46	3,398	0	+1	0	+2	0	+4	0	+9
HD-Roadways	2,350	2,431,276	0	+10	0	+10	-1	+5	+2	+37
Hillshade-9am	212	717	0	+1	0	+2	0	+4	0	+9
Hillshade-noon	223	391	0	+1	0	+2	0	+4	0	+10
Hillshade-3pm	143	1,465	0	+1	0	+2	0	+4	0	+8
HD-Fire-Points	1,980	1,753,493	0	-2	0	+5	0	+8	+1	+30

Table 9 shows that for the Block scheme, there is also an impact on the mean values, even though still a rather marginal one. However, for the variance, the changes in values are now much more pronounced than for the AK scheme, especially when setting higher values for ξ. While some changes in variance occur in attributes that have a rather high variance, and therefore constitute only a small relative change, for attributes like *Hillshade-3pm* or especially *Hillshade-noon*, the differences are also relatively large, with an increase of 11% and 51% percent, respectively.

Table 9. Change in mean and variance introduced by fingerprinting with the Block scheme, on the Forest Cover Type dataset

	β	30		25		15		10		β	30		25		15		10	
	ξ	4	8	4	8	4	8	4	8	ξ	4	8	4	8	4	8	4	8
Attribute	**Mean**									**Variance**								
Elevation	2,959									78,391	0	+13	+1	+15	+1	+48	+1	+178
Aspect	156									12,525	0	+7	0	+12	0	+35	0	+127
Slope	14						+1			56	0	+12	0	+18	0	+48	0	0
HD-Hydrology	269									45,177	0	+6	+1	+4	+1	+13	+2	0
VD-Hydrology	46				+1		+1		+1	3,398	0	+10	0	+15	0	+38	0	+87
HD-Roadways	2,350									2,431,276	0	+3	0	+3	0	+44	-2	0
Hillshade-9am	212									717	0	+11	0	+15	0	+41	0	+8
Hillshade-noon	223								-2	391	0	+11	0	+16	0	+45	0	+200
Hillshade-3pm	143				-1		-1		-1	1,465	0	0	0	+13	0	+35	0	+160
HD-Fire-Points	1,980									1,753,493	0	0	0	-4	0	+54	0	+68

The fingerprinting scheme that deals with categorical data requires a different type of measure for data utility since mean and variance are not applicable in this case. One possible measure is the number of changes introduced by marking the data.

Table 10 shows the utility effects on the Adult dataset (which contains 30,162 tuples) introduced by the extended AK scheme for fingerprinting categorical data. The utility of numerical attributes is still measured by mean and variance, where the difference in the mean is negligible (it does not exceed 0.02 and is therefore excluded from the table). The change in variance introduced by errors for numerical attributes is also rather small, as it was the case with previously presented schemes. For each categorical attribute we count how many changes in values are introduced by the fingerprint. The Number of values that change in a single categorical attribute is approximately $30,162/(2\gamma v)$. For the presented set of parameters, the introduced total number of changes is $< 4\%$ of the total number of tuples in the dataset. Due to the random nature of fingerprint insertion process, the distributions of attributes are not significantly affected.

Utility on a Machine Learning Task In this section, we evaluate the utility of the fingerprinted data sets by comparing the effectiveness of a machine learning model on correctly predicting the target class of the datasets. As we are interested only in the changes in effectiveness as compared to the original dataset,

Table 10. Change in variance and value-flips introduced by fingerprinting with the extended AK scheme, on the Forest Cover Type dataset

	γ	50		25		12		6	
	ξ	2	4	2	4	2	4	2	4
Attribute	**Variance**								
Age	173	0	0	0	0	0	0	0	+0.05
Capital Gain	54,853,968	-1	-3	-5	-11	-23	-56	-31	-67
Capital Loss	163,457	0	-1	0	-1	-1	-2	-2	-5
Hours per Week	144	0	0	0	0	0	+0.2	0	+0.3
		Value Changes							
Workclass		26	19	45	45	81	90	165	165
Education		26	18	49	43	83	84	172	173
Marital Status		24	24	46	44	101	87	207	189
Occupation		23	20	44	47	75	73	148	135
Relationship		22	22	29	41	81	89	175	189
Race		19	20	47	51	87	91	160	174
Sex		12	5	19	13	39	25	77	46
Native country		19	21	45	30	94	78	173	164

the following results report the difference in the effectiveness scores F1 and classification accuracy (on a scale of $[0, 100]\%$).

On the Adult data set, we can conclude that the differences observed when using the Logistic Regression classifier (see Table 11 are rather minute, and would not constitute a noticeable degradation of effectiveness. The trend is the same also for other classifiers, as can be seen in Table 12 for k-NN, and Table 13 for Decision Trees, as well as with Random Forests and Gradient Boosting, which are not depicted here for brevity. In a few rare cases for the k-NN Classifier and Decision Tree Classifier the classification results obtained even improved, though by the same rather marginal order of magnitude as the observed decline.

Table 11. Effect on F1 score and classification accuracy with Logistic Regression, on the Adult dataset

	$\xi = 1$		$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 50$	-0.15%	-0.07%	-0.02%	-0.01%	-0.07%	-0.03%	-0.03%	-0.02%
$\gamma = 25$	-0.25%	-0.14%	-0.13%	-0.06%	-0.10%	-0.06%	-0.14%	-0.06%
$\gamma = 12$	-0.46%	-0.22%	-0.27%	-0.12%	-0.12%	-0.08%	-0.39%	-0.15%
$\gamma = 6$	-0.68%	-0.38%	-0.41%	-0.22%	-0.46%	-0.19%	-0.80%	-0.33%
$\gamma = 3$	-2.12%	-1.01%	-1.08%	-0.52%	-0.75%	-0.32%	-1.33%	-0.62%

For the Forest Cover Type dataset, the results are provided in Table 14 for Decision Trees, Table 15 for Random Forests, and Table 16 for Logistic

Table 12. Effect on F1 score and classification accuracy with KNN, on the Adult dataset

	$\xi = 1$		$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 50$	+0.05%	+0.03%	-0.10%	-0.05%	-0.06%	-0.02%	-0.02%	+0.01%
$\gamma = 25$	-0.10%	-0.05%	+0.05%	+0.02%	+0.07%	+0.03%	-0.02%	+0.03%
$\gamma = 12$	-0.32%	-0.19%	-0.10%	-0.06%	+0.02%	+0.03%	-0.20%	-0.04%
$\gamma = 6$	-0.70%	-0.42%	-0.50%	-0.22%	-0.36%	-0.15%	-0.60%	-0.21%
$\gamma = 3$	-1.79%	-1.02%	-0.70%	-0.36%	-0.61%	-0.22%	-0.81%	-0.32%

Table 13. Effect on F1 score and classification accuracy with Decision Tree, on the Adult dataset

	$\xi = 1$		$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 50$	+0.02%	-0.08%	+0.72%	-0.04%	+0.43%	-0.03%	-0.01%	-0.07%
$\gamma = 25$	-0.05%	-0.25%	+0.32%	-0.05%	+0.49%	-0.16%	+0.36%	-0.22%
$\gamma = 12$	-0.83%	-0.36%	-0.16%	-0.05%	+0.49%	-0.12%	-0.24%	-0.04%
$\gamma = 6$	-0.93%	-0.58%	-0.34%	-0.28%	+0.30%	-0.14%	-0.93%	-0.41%
$\gamma = 3$	-2.09%	-1.04%	-0.30%	-0.64%	-0.54%	-0.39%	+0.19%	-0.54%

Regression. Similar to the Adult dataset, we can note that there are very small effects on the classification accuracy and F1 score.

In experiments with both datasets the classification accuracy and F1 score generally slightly decrease for smaller γ, i.e. by introducing more error, which is expected. However, bigger errors introduced by fingerprinting did not significantly affect the performance of any of the classifiers. This property meets the requirement of a fingerprinting scheme to be imperceptible by the users and to keep the utility of the data on the reasonable level.

Table 14. Effect on F1 score and classification accuracy with Decision Trees, on the Forest Cover Type dataset

	$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 100$	0.0%	+0.01%	+0.17%	+0.01%	+0.16%	+0.01%
$\gamma = 50$	0.0%	+0.01%	0.0%	0.0%	0.0%	+0.01%
$\gamma = 25$	-0.0%	+0.01%	+1.15%	+0.31%	+1.17%	+0.32%
$\gamma = 12$	-0.01%	-0.01%	-0.01%	0.0%	-0.01%	-0.12%
$\gamma = 6$	-0.01%	0.0%	-0.04%	-0.01%	-0.49%	-0.18%

Table 15. Effect on F1 score and classification accuracy with Random Forests, on the Forest Cover Type dataset

	$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 100$	+0.02%	-0.03%	+0.04%	-0.05%	+0.04%	+0.02%
$\gamma = 50$	+0.08%	0.0%	+0.04%	+0.6%	+0.03%	+0.04%
$\gamma = 25$	+0.09%	+0.02%	-0.09%	-0.03%	-0.05%	-0.03%
$\gamma = 12$	-0.01%	-0.0%	+0.04%	+0.03%	-0.03%	-0.05%
$\gamma = 6$	-0.06%	-0.11%	-0.01%	-0.03%	-0.0%	-0.01%

Table 16. Effect on F1 score and classification accuracy with Logistic Regression, on the Forest Cover Type dataset

	$\xi = 2$		$\xi = 4$		$\xi = 6$	
	F1	accuracy	F1	accuracy	F1	accuracy
$\gamma = 100$	0.0%	0.0%	+0.01%	0.0%	-0.01%	+0.01%
$\gamma = 50$	0.02%	0.0%	+0.01%	0.0%	-0.01%	+0.01%
$\gamma = 25$	0.0%	0.0%	0.01%	0.01%	-0.05%	+0.02%
$\gamma = 12$	0.0%	0.0%	-0.02%	0.0%	-0.11%	+0.02%
$\gamma = 6$	0.0%	0.0%	-0.03%	0.0%	-0.14%	+0.03%

5 Conclusions and Future Work

In this paper, we compared a number of previously published methods for fingerprinting relational databases with structured data. We then tested the robustness of the schemes against various types of attacks, such as sub-setting or bit-flipping. We further analysed empirically, on two benchmark datasets, how the perturbation from the fingerprint embedding affects the data utility. We followed two approaches, on the one hand computing effects directly measurable on the data, such as mean or variance, and on the other hand by measuring the effects of the fingerprint on a specific machine learning target, by comparing the achievable results on classification effectiveness. We could observe that for the selected schemes, parameters and datasets, the effects on utility of the data on the machine learning task were rather small, which is an encouraging result from a security perspective.

Table 17 illustrates the impact of common parameters on the robustness against attacks respectively on the data utility - the number of marks ω, the number of LSBs available for marking ξ, the detection threshold τ, the length of a fingerprint L, and number of recipients N. When increasing the values of these parameters, an upwards arrow denotes an increase in robustness/utility, and a downwards arrow a decrease.

Parameter ω increases the robustness against each of the presented attacks, but decreases the utility of the data, leaving the owner of the dataset the decision of how much error is it acceptable to introduce as a trade-off for the robustness. Some other parameters rather have a conflicting effect on different robustness aspects. For instance, increasing the detection threshold τ, the technique loses its robustness against subset attack, bit-flipping attack and additive attack, but on the other hand gains robustness against misdiagnosis false hit. L shows the similar effect, except that it does not have an impact on the additive attack.

Table 17. Impact of parameters on robustness against attacks resp. on data utility

↑	ω	ξ	τ	L	N
Misdiagnosis false hit	↑		↑	↑	↓
Subset Attack	↑		↓	↓	
Bit-flipping Attack	↑	↑	↓	↓	
Additive Attack	↑	↓	↓		
Utility	↓	↓			

Future work will specifically deal in more detail with approaches for fingerprinting categorical data, as this aspect has not been studied extensively in the literature so far, while categorical data (e.g. in the form of binary categories) is present in several datasets, benchmark and from real world applications. We also want to extend the analysis to other datasets, to verify that the conclusions drawn in this paper are generally valid and can be used to effectively influence the choice of parameters to obtain a secure fingerprint against the decrease in data utility.

Acknowledgments

This work was partially funded by the EU Horizon 2020 research and innovation programme under grant agreement No 732907.

References

1. Agrawal, R., Haas, P.J., Kiernan, J.: Watermarking relational data: framework, algorithms and analysis. The VLDB Journal—The International Journal on Very Large Data Bases **12**(2), 157–169 (2003)
2. Atallah, M.J., Raskin, V., Crogan, M., Hempelmann, C., Kerschbaum, F., Mohamed, D., Naik, S.: Natural language watermarking: Design, analysis, and a proof-of-concept implementation. International Workshop on Information Hiding pp. 185–200 (2001)
3. Boney, L., Tewfik, A.H., Hamdy, K.N.: Digital watermarks for audio signals. In: Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems. pp. 473–480. IEEE (1996)
4. Chen, B.C., Kifer, D., LeFevre, K., Machanavajjhala, A.: Privacy-preserving data publishing. oundations and Trends in Databases **2**(1&#8211;2), 1–167 (Jan 2009). https://doi.org/10.1561/1900000008
5. Constantin, C., Gross-Amblard, D., Guerrouani, M.: Watermill: an optimized fingerprinting system for highly constrained data. In: Proceedings of the 7th workshop on Multimedia and security. pp. 143–155. ACM (2005)

6. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE transactions on image processing **6**(12), 1673–1687 (1997)
7. Cox, I.J., Miller, M.L., Bloom, J.A., Honsinger, C.: Digital watermarking, vol. 53. Springer (2002)
8. Das, T.K., Maitra, S.: A robust block oriented watermarking scheme in spatial domain. In: International Conference on Information and Communications Security. pp. 184–196. Springer (2002)
9. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. Signal processing **66**(3), 283–301 (1998)
10. Kieseberg, P., Schrittwieser, S., Mulazzani, M., Echizen, I., Weippl, E.: An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata. Electronic Markets **24**(2), 113–124 (Jun 2014). https://doi.org/10.1007/s12525-014-0154-x
11. Lafaye, J., Gross-Amblard, D., Constantin, C., Guerrouani, M.: Watermill: An optimized fingerprinting system for databases under constraints. IEEE Transactions on Knowledge and Data Engineering **20**(4), 532–546 (2008)
12. Li, Y., Swarup, V., Jajodia, S.: Fingerprinting relational databases: Schemes and specialties. IEEE Transactions on Dependable and Secure Computing **2**(1), 34–45 (2005)
13. Liu, S., Wang, S., Deng, R.H., Shao, W.: A block oriented fingerprinting scheme in relational database. In: International Conference on Information Security and Cryptology. pp. 455–466. Springer (2004)
14. Maxemchuk, N.F.: Electronic document distribution. AT&T technical journal **73**(5), 73–80 (1994)
15. O'Ruanaidh, J., Dowling, W., Boland, F.: Watermarking digital images for copyright protection. IEE Proceedings-Vision, Image and Signal Processing **143**(4), 250–256 (1996)
16. Petitcolas, F.A., Katzenbeisser, S.: Information Hiding Techniques for Steganography and Digital Watermarking (Artech House Computer Security Series). Artech House (2000)
17. Sion, R.: Proving ownership over categorical data. In: Proceedings. 20th International Conference on Data Engineering. pp. 584–595. IEEE (2004)
18. Sion, R., Atallah, M., Prabhakar, S.: Rights protection for categorical data. IEEE Transactions on Knowledge and Data Engineering **17**(7), 912–926 (2005)
19. Sweeney, L.: K-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(5), 557–570 (Oct 2002). https://doi.org/10.1142/S0218488502001648