UNECE

CONFERENCE OF EUROPEAN STATISTICIANS

Expert Meeting on Statistical Data Confidentiality

1-3 December 2021, Poland

Fingerprinting relational data.

Tanja Šarčević (SBA Research)

tsarcevic@sba-research.org

Abstract

Fingerprinting is a method of embedding a traceable mark into digital data to verify the owner and identify the recipient of a released copy of a data set. This is crucial when releasing data to third parties, especially if it involves a fee, or if the data is of sensitive nature, due to which further sharing and leaks should be discouraged and deterred from.

Fingerprints are achieved by introducing modifications to the data that encode the owner's and recipient's identifiers. Therefore, a robust fingerprint is required to achieve successful ownership protection while affecting the data as little as possible. We focus our research on a few challenges in the domain of fingerprinting relational data. We (i) propose a framework for evaluation and analysing fingerprinting methods for relational data with regards to risks relating to the removal of the mark and data utility, (ii) analyse the trade-off between fingerprint robustness and data utility and (iii) address the problem of fingerprinting categorical data as a use-case with a smaller bandwidth for imperceptible modifications and propose a correlation-preserving technique for categorical relational data.

Fingerprinting Relational Data

Tanja Šarčević*, Rudolf Mayer*, Andreas Rauber**

* SBA Research, Floragasse 7/5, Vienna, Austria, {tsarcevic,rmayer}@sba-research.org

** University of Technology, Vienna, Austria, rauber@ifs.tuwien.ac.at

Abstract. Fingerprinting is a method of embedding a traceable mark into digital data to (i) verify the owner and (ii) identify the recipient of a released copy of a data set. This is crucial when releasing data to third parties, especially if it involves a fee, or if the data is of sensitive nature, due to which further sharing and leaks should be discouraged and deterred from. Fingerprinting is achieved by introducing modifications that encode the owner's and recipient's identifiers to the data. Therefore, a robust fingerprint is required to achieve successful ownership protection while affecting the data as little as possible. We address a few challenges in the domain of fingerprinting relational data. We (i) address the problem of fingerprinting categorical data as a use-case with a smaller bandwidth for imperceptible modifications and propose a correlation-preserving technique for categorical relational data, (ii) propose a framework for evaluation and analysing fingerprinting methods for relational data with regards to risks relating to the removal of the mark and data utility and (iii) analyse the trade-off between fingerprint robustness and data utility.

1 Introduction

Digital fingerprinting is a method that helps to protect intellectual property for various types of data and allows tracing back the recipient of the shared data instance. By combining and embedding secret, owner- and recipient-specific, mark into the data, fingerprinting allows identifying the source of digital objects and identifying the source of unauthorised data leakage. Fingerprinting facilitates sharing full data with third parties, where different recipients of the data obtain differently marked content. Since fingerprinting does not control access to the data, it is considered *passive* protection tool.

Fingerprinting techniques were first developed for the multimedia domain [Boney et al., 1996]. The generally large amount of data required to represent this content (e.g. images or video) offers sufficient space to embed the marks without significantly affecting the actual content. Fingerprinting was later extended to other types of digital data, where the effects caused by marking are of a bigger concern. These types of content include e.g. text, software, graphs, sequential data and relational databases. [Venkatesan et al., 2001, Yilmaz and Ayday, 2020, Zhao et al., 2015]

A fingerprint in the domain of relational (tabular) data is often realised by a pseudo-random pattern of modifications within the values of the dataset. Most stateof-the-art techniques address numerical data types [Li et al., 2005, Liu et al., 2005, Guo et al., 2006, Lafaye et al., 2008, Halder et al., 2010, Kamran and Farooq, 2018], and significantly fewer techniques exist for categorical data types, since their discrete nature causes larger disruptions in the data caused by marking [Sion, 2004, Bertino et al., 2005, Sarcevic and Mayer, 2020]. Fingerprinting categorical data is bounded by certain limitations, including the discrete nature of categorical values where the required modifications for embedding the marks cause a discrete (and not a minor) alteration and common correlations between attribute values that might be disrupted by modifications. Thus, a change to a categorical value is more perceptible than the (minor) change that is required for numerical values. In many real-world datasets, the attributes are of mixed type. Thus, being able to address only one type of attribute limits the usefulness of these fingerprinting techniques. The fingerprinting technique presented in [Kieseberg et al., 2014] utilises k-anonymisation to achieve privacy and ownership protection of a data set at the same time, making this scheme applicable to data sets with mixed attribute types. However, k-anonymity usually reduces the utility of the data [Šarčević et al., 2020], which entails potentially large utility losses for fingerprinted data sets.

In an attempt to bring fingerprinting to practical usage, one needs to address this shortcoming and extend the applicability of fingerprinting techniques.

The quality of a fingerprinting method can generally be assessed in two ways: (i) by the (remaining) **utility** of the fingerprinted data and (ii) by its **robustness** against malicious attacks and benign updates of the dataset. Data modifications introduced by a fingerprint inevitably decrease data utility, however, these effects can be diminished by carefully tuning the fingerprinting parameters. It is, therefore, important to assess and estimate the utility losses introduced by applying the desired fingerprinting scheme. An attack is a collective notion of different types of attempts to prevent the correct detection of a fingerprint. A malicious attacker might modify, delete or add values to the fingerprinted data with the aim to modify or erase the fingerprint. These modifications generally result in an additional decrease in data utility – therefore a fingerprint is considered robust if cannot be removed without significantly reducing data utility.

In this paper, we present our approach for fingerprinting data sets containing categorical data types. We further propose evaluation steps for assessing fingerprint robustness and data utility. We discuss the trade-off between robustness and utility, and the challenge of good parameter choice for fingerprinting.

The paper is organised as follows: In Section 2 we explain the background of fingerprinting relational data, in Section 3 we discuss the special case of fingerprinting categorical data, in Section 4 we introduce the evaluation process for robustness and utility of the fingerprinting schemes. Finally, in Section 5 we bring conclusions and future work.

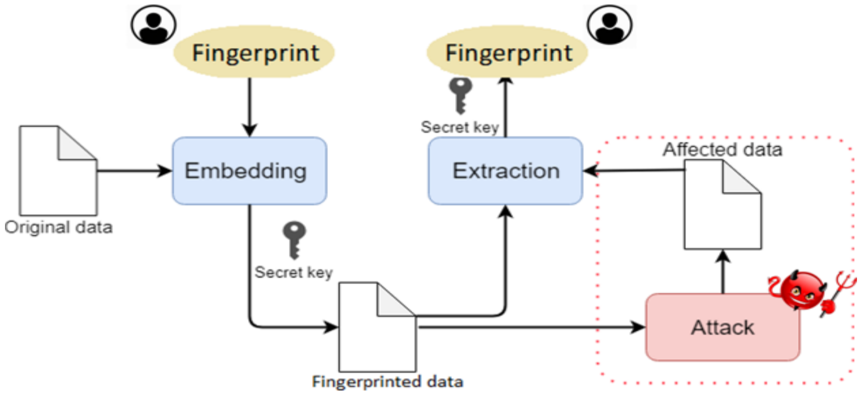

Figure 1: *Fingerprinting process*

2 Background

2.1 Fingerprinting schemes

A fingerprinting scheme is in principle encompassing two main processes - embedding (a.k.a. insertion) and extraction (a.k.a. detection). This workflow is shown in Figure 1. Within the embedding process, the fingerprint string is first created as a function of the owner's secret key and the recipient ID, usually using a hash function. Secondly, the fingerprint, i.e. the string which is unique to each recipient, is embedded into the data as a pattern of modifications made on data values. The pattern itself depends on the scheme. For example, [Liu et al., 2005] propose a scheme where the data is first divided into several blocks, of which each is associated with one fingerprint bit. The authors of [Guo et al., 2006] propose a pattern where the fingerprint bits are embedded in two separate phases. Nevertheless, the patterns depend on pseudo-random number generation, seeded by the owner's secret key. This ensures that only with access to the secret key, the pattern can be recreated, but not otherwise, even if the algorithm steps of the embedding scheme are known.

Fingerprint extraction is the reverse process of embedding. Using the secret key and the fingerprinted dataset, the marks are recreated into a fingerprint uniquely associated with a data recipient. One of the desired properties for a fingerprinting scheme is *blindness*. In a blind scheme, the extraction algorithm does not need the original dataset rows to identify the correct fingerprint. The blindness property contributes to security of the fingerprinting process, since less information needs to be securely stored in order to extract the fingerprint.

If the extraction process is not disrupted, the matching between a data copy and its recipient should be successful. However, the process may underlay certain malicious attacks or benign modifications to the dataset. This may affect the detection algorithm, therefore one needs to ensure the scheme is robust by design. Secondly, the fingerprinted data set is shared with third parties, and therefore its credibility and utility need to be preserved, despite the modifications made by the process of embedding. These two concerns are discussed in the following.

Parameters We can classify parameters relevant in the workflow in two main groups – parameters that are inherent properties of the dataset and the parameters that are describing properties of a fingerprinting scheme, i.e. the way the fingerprint is created and embedded into the dataset. The effect of the first group of parameters can be analysed throughout a big-scale evaluation using a great variety of datasets. In this work we focus on the latter group of parameters, namely:

- absolute, *n_marks*, or relative number of marks in the dataset (rel. to the total number of data rows) *%_marks*
- number of attributes chosen for marking, absolute *n_attributes* or relative *%_attributes*
- *magnitude* of modifications, e.g. for numerical values the number of least significant bits available for marking
- length of a fingerprint L

In different works, these parameters are expressed by differently named parameters (e.g. in [Li et al., 2005] *%_marks* → γ, *magnitude* → ξ in [Guo et al., 2006] *%_marks* → β, etc.), thus we unify the nomenclature to enable application of our process on a range of different schemes.

2.2 Robustness

The robustness of a fingerprinting scheme is measured as the resilience of the scheme against modifications and malicious attacks. In literature, robustness is usually measured as to how insusceptible the scheme's detection algorithm is against different types of attacks. For relational datasets these usually are [Kamran and Farooq, 2018]:

- subset attack (deletion attack): The attacker releases only a subset of the fingerprinted data set, either as a subset of tuples (records, rows) in a *horizontal* or a subset of attributes (features, columns) in a *vertical subset attack*
- flipping attack (alteration attack): the attack comes in different flavours for different data types. For categorical values, the attacker flips selected values to random values from the domain; for numerical values, the attacker flips some of the least significant bits.
- superset attack (record-insertion attack): the attacker adds (synthetic) rows to confuse the detection process
- additive attack: the attacker produces and embeds their own fingerprint on top of the existing one, to try and claim the false ownership of the data

Robustness may be expressed via a number of measures. As proposed in [Li et al., 2005], those are:

- *Misdiagnosis false hit* (fh^D): The probability of detecting a valid fingerprint from data that has not been fingerprinted.
- *Misattribution false hit* (fh^A): The probability of detecting an incorrect but valid fingerprint from fingerprinted data, i.e. a wrong suspect.
- *False negative* (fn): The probability of detecting no valid fingerprint from fingerprinted data, i.e. no suspect.
- *False miss* (fm): Inability to detect the correct fingerprint. False miss rate is the sum of the misattribution false hit and false negative, i.e. $fm = fh^A + fn$.

fh^D is a property of the fingerprinting scheme, while the latter three metrics measure the success of the attacks applied on the data (i.e. if there is no attack, they should be equal to zero). Successful attacks are considered those that cause a high rate for these three measures, while not rendering the data useless, i.e. not distorting the data too much by the attack. Data utility is often not discussed in detail nor evaluated in the literature. In our framework, we, however, include an evaluation of the data utility after an attack, which is described in Section 4.1.

2.3 Data utility

Modifications introduced by data fingerprinting unavoidably change the data, and thus likely its utility. Preserving data utility is hence a conflicting goal that needs to be taken into consideration when fingerprinting relational data alongside the robustness of the scheme. The literature mentions two ways to measure the preserved utility on the fingerprinted data: (i) changes in data statistics such as mean, variance, distributions, etc. [Li et al., 2005], i.e. *data-oriented utility metrics* and (ii) effect of fingerprint modifications on learning tasks such as learning a predictive classification model [Šarčević and Mayer, 2019], so-called *task-oriented utility metrics*. The latter is a more specific notion of effectiveness in a scenario where a data holder has certain usage in mind for their data, such as a predictive task on one of the attributes. A utility notion along this lines is thus the performance loss when using fingerprinted data for the (same or a similar) task the data holder has aimed for, compared to performing it on the original data.

3 Fingerprinting categorical data

Fingerprinting categorical data in relational data set got considerably less attention in literature. The reason for this is certain limitations that categorical data poses in comparison to numerical data. Firstly, the embedding channels for modifications are rather low due to the discrete nature of categorical data. Numerical data has the advantage that modifying a value by a small margin will normally not affect the data as a whole. This, for example, is shown in our evaluations in Section 4. For categorical (nominal) data, it is hard, or even impossible, to quantify the amount of modification, hence hard to apply a minor modification. We assume that modifications such as changing one character in the value representation are not desirable modifications since it's easily perceivable and hence removable. Secondly, it is likely that the discrete modifications within the data will disrupt the mutual semantic correlations between the attributes. Modifying attributes independently may lead to obtaining non-consistent records, by introducing an uncommon or impossible combinations of values in the data. For example, consider a medical database containing attributes such as *sex* and *numberOfPregnancies* – a record containing (*sex* :male, *numberOfPregnancies*:1) will be highly suspicious.

Our approach for fingerprinting categorical data addresses the problem of disrupting the correlations in the data to ensure a less perceivable fingerprint [Sarcevic and Mayer, 2020]. The scheme follows the process outline described in Section 2 and Figure 1. The fingerprint creation and embedding pattern follow the steps of [Li et al., 2005]. This means that the fingerprint is created using a hash function and a combination of the data holder's secret key and the recipient's ID. The location of the values to be modified by the fingerprint bits is made based on a pseudo-random number generation seeded by the owner's secret key. Further, whether or not the chosen value will be modified depends on the value of the corresponding fingerprint bit, 0 or 1. This step is useful for recreating the fingerprint in the detection phase. For value modification, we propose a k-NN-based method. Once the value to be modified (defined by $(attribute, row)$) is determined by the pseudo-random number generator, the algorithm finds the n neighbours of the *row* of the chosen value. The new value is then chosen from the set of *attribute* values in the neighbourhood, weighted by their frequency. The insertion algorithm enables preserving the correlations between categorical attributes by eliminating the occurrences of value combinations that were not initially in the original data set, and preserving the low frequency of value combinations that were already rare in the original data set.

The detection phase inverts the embedding process: from marked values, the pseudo-random number sequence is recovered and hence the fingerprint bits. The full algorithmic steps are shown in Algorithm 1.1 and 1.2 from [Sarcevic and Mayer, 2020]. The scheme is however not blind, since the values from the original dataset are necessary to extract the fingerprint. Our future work will consider adapting the extraction algorithm in a way that less information about the original dataset is needed, such as attribute histograms instead of full data entries.

4 Fingerprint scheme evaluation process

An evaluation process of fingerprinting schemes contains two main parts: (i) analysis of the utility of the fingerprinted data and (ii) robustness analysis. The aim of the data holder is to embed a robust fingerprint into the data, i.e. a fingerprint that can not be easily removed by malicious attacks or benign updates on the dataset, while at the same time keeping the utility of the dataset on an acceptable level. In the following, we discuss the particular elements of the two parts of the process in detail.

4.1 Data utility

The proposed evaluation encompasses the utility evaluation according to two groups of metrics described in Section 4.1.

Data-oriented metrics help to get an insight into the amount of modification that is introduced due to the fingerprint. For instance, we measure the change in mean and variance of numerical attributes

Task-oriented utility metrics are measured under the assumption that the purpose of the data is to serve as a training set for predictive modelling with a defined target attribute and, to that end, estimate the utility loss. For that purpose, we train a number of well-known machine learning models, using the fingerprinted data as a training set, and evaluate the models on the original, unmodified holdout set via performance metrics such as *accuracy* or *F1 score* in case of the classification task, or e.g. *mean squared error (MSE)* for a regression task. We then train and evaluate the same types of models on the original data and compare the performances. The *performance loss* (i.e. *accuracy loss*, *MSE loss*, ...) serves as a main metric for assessing data utility in the task-oriented group of metrics either as (i) *absolute performance loss* (the absolute difference in performance metric) or (ii) *relative performance loss* (absolute performance loss divided by the performance on original data set).

Table 1: *Effect on F1 score and classification accuracy with Logistic Regression, on the Adult dataset fingerprinted using [Li et al., 2005] via absolute performance loss, with ϵ denoting the number of LSBs available for modification (i.e. magnitude)*

	$\xi = 1$		$\xi = 2$		$\xi = 6$	
$\%\_marks$	F1	accuracy	F1	accuracy	F1	accuracy
2%	-0.15%	-0.07%	-0.02%	-0.03%	-0.03%	-0.02%
4%	-0.25%	-0.14%	-0.13%	-0.06%	-0.14%	-0.06%
8%	-0.46%	-0.22%	-0.27%	-0.12%	-0.39%	-0.15%
17%	-0.68%	-0.38%	-0.41%	-0.22%	-0.80%	-0.33%
33%	-2.12%	-1.01%	-1.08%	-0.52%	-1.33%	-0.62%

We observed trends in effects on classification accuracy and F1 score that the fingerprinting has under different parameter settings in [Šarčević and Mayer, 2019], obtaining usually a minor degradation in performance. This can be shown by the example of our results in Table 1, where we compare the F1 score loss and accuracy loss of the logistic regression models using Adult dataset[1] under different settings

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

for parameters *%_marks* and *magnitude*.

4.2 Robustness

The robustness of the scheme is assessed by its resilience against modifications on the datasets, or other attempts of confusing the detection process of the scheme.

Attacker model We define the attacker model, a *white-box naive attacker*, which will be applied to all considered attacks in the continuation of the paper.

White-box access: The attacker is assumed to know the algorithmic steps of the embedding and extraction processes, and all fingerprinting parameters, such as length of a fingerprint, strength and magnitude. Only the owner's secret key remains unknown to the attacker.

Naive attacker: The attacker does not use any background knowledge about the data set and all the modifications (flipping, deletion, etc.) are applied randomly to the data values, i.e. each value has an equal probability to be attacked. We use the naive attacker model to create a baseline for the robustness estimation and comparison with attacks by the attacker with certain background knowledge in our future work.

Robustness estimation The robustness estimation in our process is focused on *misdiagnosis false hit* metric ($fh^D \in [0,1]$) and *false miss* metric ($fm \in [0,1]$), which encompasses the other two mentioned metrics in Section 2.2, *misattribution false hit* and *false negative*. Robustness is empirically evaluated by recording the detection rate of the scheme under random attacks. Lower fh^D and higher fm indicate stronger attacks. In the following, we highlight some robustness evaluation results from [Šarčević and Mayer, 2019].

Table 2: *Misdiagnosis false hit rate fh^D for exemplary fingerprint sizes L in bits*

L	8	16	32	64	128
$fh^D(n\_marks = 800))$	0.7208	0.0052	2.70×10^{-7}	7.30×10^{-16}	5.31×10^{-33}
$fh^D(n\_marks = 400)$	0.9151	0.0084	7.01×10^{-7}	4.92×10^{-15}	2.42×10^{-31}

Misdiagnosis false hit depends on the length of the fingerprint L and the absolute number of marks *n_marks*. Table 2 shows the exponential dependence of the fh^D on L, hence choosing a fingerprint of length ≥ 32 is a rule of thumb.

Furthermore, we analysed fm rates for a number of malicious attacks on the dataset and we compared the rates between different datasets (Adult and Forest Cover Type[2]), as shown in Table 3 for *horizontal subset attack* and between different schemes in Table 4 for *flipping attack*. A general observation is that setting the

[2] https://www.kaggle.com/c/forest-cover-type-prediction

Table 3: *Dataset comparison: Experimental results of subset attack success (fm) against the scheme [Li et al., 2005], on the Forest Cover Type dataset (left) and Adult (right), where p' denotes the strength of the attack*

$\%\_marks$	$p' = 80\%$	$p' = 95\%$	$p' = 99\%$
17%	0	0	0.01
4%	0	0	1.0
2%	0	0.19	1.0
1%	0	0.99	1.0

$p' = 80\%$	$p' = 95\%$	$p' = 99\%$
0.20	0.95	1.0
0.99	1.0	1.0
1.0	1.0	1.0
1.0	1.0	1.0

Table 4: *Scheme comparison: Experimental results of flipping attack success (fm) on the Adult dataset against scheme [Li et al., 2005] (left) and Block scheme [Liu et al., 2005] (right), where p' denotes the strength of the attack*

$\%\_marks$	$p' = 30\%$	$p' = 40\%$	$p' = 45\%$
20%	0	0.50	0.56
8%	0	0.50	1.0
4%	0	0.54	1.0

$p' = 30\%$	$p' = 40\%$	$p' = 45\%$
0	0	0.50
0	0.50	0.92
0.08	0.50	1.0

$\%\_marks$ to high values (i.e. marking a lot of data values) results in lower fm rates, i.e. better robustness for the scheme, making this one of the main control parameters for achieving good robustness of the scheme. From the comparison on fm between datasets, there is a clear advantage in marking a bigger dataset, such as Forest Cover Type, over a smaller one, in this case, Adult. This implies that it is crucial to consider data properties for defining a robust scheme and choosing its parameters.

4.3 Utility-robustness trade-off

From the evaluations on utility and robustness, it is evident that some parameters have conflicting effects on data utility and robustness – by tweaking parameters in favour of one, the other one would decline. Hence, a good trade-off needs to be achieved for applying fingerprints successfully. For some parameters such as L, a general rule for the choice exists, that would lead to a robust scheme. However, other parameters such as n_marks or $magnitude$ should likely be differently set depending on data properties.

5 Conclusions and Future Research

This paper summarises the recent developments in the field of fingerprinting relational data sets such as utility and robustness evaluation on different techniques, task-oriented utility metrics and fingerprinting categorical data with a focus on preserving the semantics of the data set.

The focus of our future work is to bridge the gap to the practical application of fingerprinting techniques. To this end, we aim at two main goals: (i) designing tools for fingerprinting relational data in practice, and (ii) aiding the fingerprint parameter choice for the data-holder. The first goal requires an approach for a unifying fingerprinting scheme, that would apply to all data types in a data set and an open-source implementation. The second goal is motivated by the observation that choosing a good parameter setting is a black-box for a data holder unless a sufficient utility and evaluation analysis has been done. Hence, we recognise a need for a parameter choice guideline, that would guide a data holder through the choice and indicate the robustness risks and expected utility losses. To this end, we will expand our analysis to capture the general patterns and trends of robustness and utility for certain parameter choices and dataset properties and based on this, (semi-)automatise the process of parameter choice.

Acknowledgement

This work was partially funded by the “Industrienahe Dissertationen” program (No 878786) of the Austrian Research Promotion Agency (FFG), under project “IPP4ML” and the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078 (FeatureCloud).

References

[Bertino et al., 2005] Bertino, E., Ooi, B. C., Yang, Y., and Deng, R. H. (2005). Privacy and ownership preserving of outsourced medical data. In *21st International Conference on Data Engineering (ICDE’05)*, pages 521–532. IEEE.

[Boney et al., 1996] Boney, L., Tewfik, A. H., and Hamdy, K. N. (1996). Digital watermarks for audio signals. In *Proceedings of the third IEEE international conference on multimedia computing and systems*, pages 473–480. IEEE.

[Guo et al., 2006] Guo, F., Wang, J., and Li, D. (2006). Fingerprinting Relational Databases. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, SAC ’06, pages 487–492, New York, NY, USA. ACM. event-place: Dijon, France.

[Halder et al., 2010] Halder, R., Pal, S., and Cortesi, A. (2010). Watermarking techniques for relational databases: Survey, classification and comparison. *J. Univers. Comput. Sci.*, 16(21):3164–3190.

[Kamran and Farooq, 2018] Kamran, M. and Farooq, M. (2018). A Comprehensive Survey of Watermarking Relational Databases Research. *arXiv:1801.08271 [cs]*. arXiv: 1801.08271.

[Kieseberg et al., 2014] Kieseberg, P., Schrittwieser, S., Mulazzani, M., Echizen, I., and Weippl, E. (2014). An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata. *Electronic Markets*, 24(2):113–124.

[Lafaye et al., 2008] Lafaye, J., Gross-Amblard, D., Constantin, C., and Guerrouani, M. (2008). Watermill: An optimized fingerprinting system for databases under constraints. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):532–546.

[Li et al., 2005] Li, Y., Swarup, V., and Jajodia, S. (2005). Fingerprinting relational databases: schemes and specialties. *IEEE Transactions on Dependable and Secure Computing*, 2(1):34–45.

[Liu et al., 2005] Liu, S., Wang, S., Deng, R. H., and Shao, W. (2005). A Block Oriented Fingerprinting Scheme in Relational Database. In Park, C.-s. and Chee, S., editors, *Information Security and Cryptology – ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 455–466, Berlin, Heidelberg. Springer.

[Šarčević and Mayer, 2019] Šarčević, T. and Mayer, R. (2019). An evaluation on robustness and utility of fingerprinting schemes. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 209–228. Springer.

[Sarcevic and Mayer, 2020] Sarcevic, T. and Mayer, R. (2020). A correlation-preserving fingerprinting technique for categorical data in relational databases. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 401–415. Springer.

[Šarčević et al., 2020] Šarčević, T., Molnar, D., and Mayer, R. (2020). An analysis of different notions of effectiveness in k-anonymity. In *International Conference on Privacy in Statistical Databases*, pages 121–135. Springer.

[Sion, 2004] Sion, R. (2004). Proving ownership over categorical data. In *Proceedings. 20th International Conference on Data Engineering*, pages 584–595. IEEE.

[Venkatesan et al., 2001] Venkatesan, R., Vazirani, V., and Sinha, S. (2001). A graph theoretic approach to software watermarking. In *International Workshop on Information Hiding*, pages 157–168. Springer.

[Yilmaz and Ayday, 2020] Yilmaz, E. and Ayday, E. (2020). Collusion-resilient probabilistic fingerprinting scheme for correlated data. *arXiv preprint arXiv:2001.09555*.

[Zhao et al., 2015] Zhao, X., Liu, Q., Zheng, H., and Zhao, B. Y. (2015). Towards graph watermarks. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, pages 101–112.