Master Thesis Proposal

Fingerprinting Relational Databases; Quality Evaluation and Impact on Learning Tasks

Advisor: Ao.univ.Prof. Dr. Andreas Rauber

February 20, 2019

1 Motivation and Problem Statement

Nowadays the trends of sharing and processing digital data have highly increased. Machine learning is experincing the great growth that is expected to continue increasing. Big requirements for ML-based systems are the amount, as well as the quality of the data. Data is a valuable asset to its owner, therefore any type of unauthorized distribution or usage of data by the third parties violates the owner's rights and rights of the authorized buyers. It is data owner's interest to detect data leakages and to prove the ownership of the data. One step further in ownership protection is to detect the source of such leakage, i.e. the person that distributed the data without the owner's authorization.
Watermarking is a technique of hiding copyright information within the digital asset. Fingerprinting is another class of information hiding techniques which strengthens the ownership protection adding the property of leakage source detection. By fingerprinting, each buyer's identification is embedded within the data, producing that way a distinct copy of the data for each of the buyers. This area has been widely studied in the domain of multimedia data, while in the area of relational databases there are a few proposed approaches.
Usually, the fingerprint is embedded in the database such that the values are slightly altered, which can affect the quality of the data. Once embedded in the data, the fingerprint should be detectable only by the data owner and it should not be easily removed from the data by any operations over the database, such as removal or adding the tuples, neither by minor alterations of the data. Altering the data might sound like a limitation for the buyer, but according to the majority of the proposed techniques, both for watermarking and fingerprinting, this is mandatory to achieve the watermark/fingerprint robustness.
The challenge is to find an acceptable tradeoff between robustness and data quality and utility.

2 Expected Results

The expected outcome of this thesis is a thorough experimental evaluation of different fingerprinting techniques and the analysis of effects that fingerprinted data have on learning tasks, such as classification. The experimental evaluation will analyze a few properties that a fingerprinting scheme should satisfy:

Detectability The owner should be able to detect and distinguish a fingerprint from the dataset. The fingerprint should be detectable also from the dataset subset, as well as the modified version of the dataset.

Imperceptibility The utility of the data should not be significantly affected by modifications caused by fingerprinting. In the literature, this is usually measured by changes in mean and variance of the numerical attributes. Besides mean and variance, we will measure data imperceptibility by measuring the performance of the fingerprinted data on a classification task using different classifiers.

Robustness Fingerprinting schemes should be robust against benign dataset operations and malicious attacks that may remove or modify embedded fingerprint. Benign database operations are those with no aim of unauthorised usage or release of the database, such as deleting, adding and updating tuples. Malicious attacks include selective modifications of the fingerprinted database, releasing a subset of a database or modifying and erasing the embedded fingerprint. The malicious attacks that will be considered, as commonly mentioned in the literature are:

1. bit-flipping attack: values of some bits in the fingerprinted data are inverted so that fingerprint cannot be detected correctly
2. subset attack: the attacker releases only a subset of the fingerprinted dataset
3. superset attack: the attacker adds additional tuples to the fingerprinted dataset
4. collusion attack: attackers with access to multiple fingerprinted copies of data (with different fingerprints) create a new copy of the data where neither of the embedded fingerprints might be detectable, thus no member of the malicious coalition might be implied as a traitor
5. additive attack: a special case of bit-flipping attack where the attacker adds an additional fingerprint on the fingerprinted data that might distort the initial fingerprint

The evaluation is also expected to provide insight into relations between different robustness, imperceptibility and detectability levels.

3 Methodological Approach

Literature Review The literature related to watermarking and fingerprinting relational databases will be reviewed in order to gather information about the existing techniques in the area. We are also interested in the existence of implementations of any of the techniques because it would speed up the second part of the approach. The research will be done on measures used for describing robustness, imperceptibility and detectability of the fingerprinted data.

Implementation of fingerprinting techniques In order to perform the empirical evaluation, the approaches where the implementation is not available must be implemented by ourselves. For this, the well-defined algorithm steps of fingerprinting methods will be followed. The system should allow easy changing of the parameter setting for the fingerprinting method.

Empirical evaluation of fingerprint robustness for different fingerprinting techniques The appropriate datasets will be chosen for the experiment runs. We will test the robustness against the malicious attacks and benign operations on the dataset already stated in section 2. The experiments should be run with different parameter settings so that it is possible to explain the impact of each of the parameters to fingerprint robustness. The techniques will be compared based on their resilience to attacks.

Empirical evaluation of the quality effects on a fingerprinted dataset for different fingerprinting techniques The fingerprint brings a certain distortion to the data. Therefore, we want to measure the utility of the fingerprinted data and compare it to the original dataset. The first step will be to examine some measures regarding the data itself, for example, the mean and variance of numerical attributes. Those results will be analyzed and compared to the theoretical results or experimental results from some of the previous works.

Empirical evaluation of the impact of a fingerprinted dataset on a specific learning target This is a second step of measuring the data utility. In this part, we will measure the utility regarding some specific learning task and performance of the fingerprinted data compared to the original dataset. We will choose the appropriate data, preferably the same as in previous steps. It is also necessary to choose a learning task and the target attribute. We will choose a few different supervised machine learning tasks to get some broader insights. We will use the same pipeline to perform the task using original dataset and the fingerprinted one in order to compare the results. The results will be evaluated based on the data precision metrics, as data accuracy, precision, etc. Ideally, we want to find the relationship between measures from this phase and the previous phase, and examine the tradeoff between resilience to malicious attacks and quality effects.

4 State of the Art

Most of the current state-of-art fingerprinting methods extend the watermarking technique proposed by Agrawal et al. [1]. The technique in principle contains two algorithms: watermark insertion and watermark detection.
The watermark insertion algorithm marks certain numerical attributes such that the least significant bits (LSBs) are altered, therefore this technique assumes that the database contains one or more numerical attributes. The database owner is left to decide the number of LSBs available for marking ξ, such that the changes of marked attributes stay imperceptible. The insertion algorithm then uses a cryptographic pseudorandom sequence generator $\mathcal{G}$ seeded by a secret key known only to the owner of the database concatenated with the primary key attribute value of each tuple from a database. The numbers generated by $\mathcal{G}$ determine the tuples, attributes within the tuples and LSBs within the attribute values to be marked, as well as the mark itself. It is computationally infeasible to predict the next number in the cryptographic pseudorandom sequence. Thus, it is computationally infeasible to guess the marking pattern without the knowledge of the owner's private key.
The detection algorithm contains calculating the sequence using a cryptographic pseudorandom sequence generator with the same seeds as in the insertion algorithm. Those sequences will be the same because repeated executions of the generator seeded by the same value always produce the same sequence. Thus, the detection algorithm finds which bits within the database should have been marked and counts how many of them match the bits from the database suspected for piracy. If the number of matches is "large", the database owner can suspect the piracy. On the other hand, if the matching is "too small", the database owner can suspect that the attacker somehow identified the marking pattern and recreated the original database values. The number of matches necessary for suspecting piracy is defined by a parameter called *significance level*.
The authors analyzed the robustness of this technique against the number of malicious attacks, namely, subsetting attacks, bit-flipping attacks, mix-and-match attack and false claim of ownership.

Li et al. in [2] extended this watermarking technique into a fingerprinting technique. Marking scheme embeds different bitstrings - *fingerprints* in different releases of data. The owner generates buyer's fingerprint from the owner's secret key and the buyer's serial number using a cryptographic hash function. This method of generating the fingerprints avoids storing buyer-fingerprint pairs and additional security management for this database. Similarly to the watermarking technique [1], this fingerprinting technique consists of the insertion algorithm and detection algorithm.
The insertion algorithm follows the steps for selecting the tuple, attribute, LSBs and mark the same as in insertion scheme from [1], and additionally embeds the generated fingerprint by XOR function applied on the mark (in this algorithm called *mask*) and a selected fingerprint bit.
The aim of the detection algorithm of the fingerprinting methods, in general, is

to determine whether the suspicious database is pirated, as well as to identify the source of the unauthorized release of the database. The detection algorithm from [2] reverts the fingerprint insertion process similarily to [1]. It locates the bits that should have been altered and compares the matching of the extracted fingerprint with buyers' fingerprints. The exact bit matching of size τ of the extracted fingerprint to some buyer's fingerprint implies this buyer to be a traitor. τ is a parameter related to the assurance of the detection process.

In [3], authors propose a block-oriented fingerprinting scheme for relational databases inspired by a fingerprinting scheme for images from [4]. This method also relies on altering the LSBs at certain locations in the database.
In the insertion algorithm, the LSBs of numerical values from the database (as much LSBs as it is allowed to change) are combined into a two-dimension image and separated into blocks of size $\beta \times \beta$. Then a pseudorandom number generator is used to decide the block and position within the block where the fingerprint should be embedded until all blocks are marked. Fingerprint bits will be embedded again if there is still unmarked blocks left when all fingerprint bits have been embedded. A fingerprint is produced in the same manner as in previous techniques, using the cryptographic hash function seeded by the owner's secret key and the user's serial number.
Detection stage consists of sorting the suspicious database according to the primary keys and filling out the database with the original values in case of deletion. The location where the fingerprint bit is supposed to be is calculated as in insertion algorithm 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 1, the detected fingerprint is said to be 1, otherwise 0.

Authors of watermarking and fingerprinting system *Watermill* [5, 6] further extend the methods from [1] and [2] by considering the constraints of data alteration and treating fingerprinting as an optimization 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.

In paper [7], the authors gave a classification and brief analysis of relevant watermarking and fingerprinting techniques.

All of the above fingerprinting techniques have one restriction in common - they are applicable only on numerical attributes since they are all bit-resetting techniques. It would not make sense to make such alterations on categorical data.

5 Relevance to the Curricula of Logic and Computation

- 186.112 Heuristic Optimization Techniques
- 181.140 Database Theory
- 181.190 Problem Solving and Search in Artificial Intelligence
- 184.702 Machine Learning
- 107.386 Classification and Discriminant Analysis
- 199.082 Machine Learning Security
- 188.982 Privacy Enhancing Technologies
- Mathematical Programming
- Modeling and Solving Constrained Optimization Problems

References

[1] R. Agrawal, P. J. Haas, and J. Kiernan, "Watermarking relational data: framework, algorithms and analysis," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 12, no. 2, pp. 157–169, 2003.

[2] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: Schemes and specialties," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 34–45, 2005.

[3] S. Liu, S. Wang, R. H. Deng, and W. Shao, "A block oriented fingerprinting scheme in relational database," in *International Conference on Information Security and Cryptology*, pp. 455–466, Springer, 2004.

[4] T. K. Das and S. Maitra, "A robust block oriented watermarking scheme in spatial domain," in *International Conference on Information and Communications Security*, pp. 184–196, Springer, 2002.

[5] C. Constantin, D. Gross-Amblard, and M. Guerrouani, "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] J. Lafaye, D. Gross-Amblard, C. Constantin, and M. Guerrouani, "Watermill: An optimized fingerprinting system for databases under constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 532–546, 2008.

[7] M. Kamran and M. Farooq, "A comprehensive survey of watermarking relational databases research," *arXiv preprint arXiv:1801.08271*, 2018.