

# wmdb.\*: Rights Protection for Numeric Relational Data

Radu Sion, Mikhail Atallah, Sunil Prabhakar \*

In this demonstration we introduce **wmdb.\***, our solution for numeric relational data rights protection through watermarking. Rights protection for relational data is important in areas where sensitive, valuable content is to be outsourced. A good example is a data mining application, where data is sold in pieces to parties specialized in mining it. Different avenues are available, each with its own advantages and drawbacks. Enforcement by legal means is usually ineffective in preventing theft of copyrighted works, *unless* augmented by a digital counter-part, for example watermarking. While being able to handle higher level semantic constraints such as classification preservation, our solution also addresses important attacks, such as subset selection, random and linear data changes.

Digital Information Hiding as a method of Rights Protection (also known as Digital Watermarking), hides an indelible “rights witness” (watermark) within the digital Work to be protected, by slightly altering it. Its main purpose is to protect the Work from unauthorized duplication and distribution by enabling provable ownership. The soundness of such a method relies on the assumption that (i) the insertion of the mark does not destroy the value of the Work (i.e. it is still useful for its *intended purpose*); and that (ii) it is difficult for a malicious adversary (Mallory) to remove or alter the mark beyond detection without destroying the value of the Work. Mallory, and the ability to resist his attacks (mostly aiming at removing the embedded watermark) turn out to be one of the major concerns in the design of a watermarking solution. Clearly, the notion of value or utility of the object is central to the watermarking process. This value is closely related to the type of data and its intended use. For example, in the case of software the value may be in ensuring equivalent computation, and for text it may be in conveying the same meaning (i.e. synonym substitution is acceptable). Similarly, for a collection of numbers, the utility of the data may lie in the actual values, in the relative values of the numbers, or in the distribution (e.g. normal with a certain mean). The main challenges in this new domain derive from the fact that, since the associated data types do not have fixed, well defined semantics (as compared to multimedia) and may be designed for machine ingestion, identifying the available “bandwidth” for watermarking becomes as important as the actual encoding algorithms.

In [1] we introduced a solution for watermarking numeric relational content composed of: (i) a resilient

watermarking method for relational data (ii) a technique for enabling user-level run-time control over properties that are to be preserved as well as the degree of change introduced (iii) a complete, user-friendly implementation for numeric relational data. The main contribution of this work is the realization of the fact that in the relational database setting it is important to preserve structural and semantic properties of the data, as part of any sound watermarking procedure. Sometimes it is undesirable or even impossible to map higher level semantic constraints into low level (combined) change tolerances for individual tuples or attributes. It should be noted that not all constraints of the database need to be specified. A practical approach would be to begin by specifying a mean square error bound on individual items. Further semantic or structural constraints that the user would like to preserve can be added to these basic constraints. The practically infinite set of potential semantic constraints that can be desired/imposed on a given data set makes it such that a different, more versatile, “data goodness” (i.e. semantically) assessment method is required.

We solve this problem by treating each of these properties as a constraint on the usability of the data. We allow the expression of such constraints in terms of arbitrary code or SQL queries over the relations, with associated requirements (usability metric functions). For example, the requirement that the result of the join (natural or otherwise) of two relations does not change by more than 3% can be specified. Constraints that arise from the schema (chiefly key constraints), can easily be specified in a form similar to (or derived from) SQL *create table* statements. In addition, integrity constraints (e.g. such as *end\_time* being greater than *begin\_time*) can be expressed. The watermarking algorithm is then applied with these constraints as input. At each step, data quality is re-evaluated and performed data alterations are un-done if required. Using this approach we can ensure that any changes made by the watermarking algorithm do not violate the required properties.

Here we feature our implementation and perform some of the experiments discussed in [1]. For example we show how various higher level semantic constraints such as classification preservation and maximum absolute change bounds are naturally handled and how random alteration attacks are well survived.

## References

- [1] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data. In *Proceedings of ACM SIGMOD*, 2003.

\*Computer Sciences, Purdue University, West Lafayette, IN, 47907, USA, [sion, mja, sunil]@cs.purdue.edu