A Generic Distortion Free Watermarking Technique for Relational Databases

Sukriti Bhattacharya and Agostino Cortesi

Dipartimento di Informatica
Universita Ca' Foscari Venezia
Via Torino 155, 30170 Venezia, Italy
sukriti@dsi.unive.it, cortesi@unive.it
http://www.unive.it

Abstract. *In this paper we introduce a distortion free watermarking technique for relational databases based on the Abstract Interpretation framework. The watermarking technique is partition based. The partitioning can be seen as a virtual grouping, which does not change neither the value of the table's elements nor their physical positions. Instead of inserting the watermark directly to the database partition, we treat it as an abstract representation of that concrete partition, such that any change in the concrete domain reflects in its abstract counterpart. The main idea is to generate a binary image of the partition as a watermark of that partition, that serves as ownership proof as well as tamper detection.*

Key words: Database Watermarking, HMAC, Galois Connection, Abstract Interpretation.

1 Introduction

Watermarking is a widely used technique to embed additional but not visible information into the underlying data with the aim of supporting tamper detection, localization, ownership proof, and/or traitor tracing purposes [1]. Watermarking techniques apply to various types of host content. Here, we concentrate on relational databases. Rights protection for such data is crucial in scenarios where data are sensitive, valuable and nevertheless they need to be outsourced. Unlike encryption and hash description, typical watermarking techniques modify the ordinal data and inevitably cause permanent distortion to the original ones and this is an issue when integrity requirement of data are required. Database watermarking consists of two basic processes: watermark insertion and watermark detection [1], as illustrated in Figure 1. For watermark insertion, a key is used to embed watermark information into an original database so as to produce the watermarked database for publication or distribution. Given appropriate key and watermark information, a watermark detection process can be applied to any suspicious database so as to determine whether or not a legitimate watermark can be detected. A suspicious database can be any watermarked database or innocent database, or a mixture of them under various database attacks.

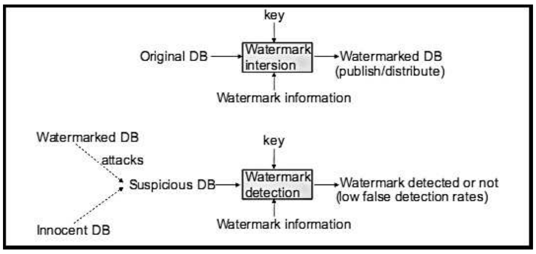

Fig. 1. Basic watermarking process

Watermarking has been extensively studied in the context of multimedia data for the purpose of ownership protection and authentication [11] [8]. The increasing use of relational database systems in many real life applications created an ever increasing need for watermarking database systems. As a result, watermarking relational database systems is now merging as a research area that deals with the legal issue of copyright protection of database systems.

The first well-known database watermarking scheme for relational databases was proposed by Agrawal and Kiernan [1] for watermarking numerical values. The fundamental assumption is that the watermarked database can tolerate a small amount of errors. Since any bit change to a categorical value may render the value meaningless, Agrawal and Kiernan's scheme cannot be directly applied to watermarking categorical data. To solve this problem, Sion [14] proposed to watermark a categorical attribute by changing some of its values to other values of the attribute (e.g., 'red' is changed to 'green') if such change is tolerable in certain applications. There have been other schemes proposed for watermarking relational data. In Sion et al.'s [15] scheme, an arbitrary bit is embedded into a selected subset of numeric values by changing the distribution of the values. The selection of the values is based on a secret sorting. In another work, Gross-Amblard [9]designs a query preserving scheme which guarantees that special queries (called local queries) can be answered up to an acceptable distortion.

All of the work cited so far ([1][9][2][14][15]) assume that minor distortions caused to some attribute data can be tolerated to some specified precision grade. However some applications in which relational data are involved cannot tolerate any permanent distortions and data's integrity needs to be authenticated. In order to meet this requirement, we further strengthen this approach: we propose a distortion free watermarking algorithm for relational databases, and we discuss it in the abstract interpretation framework proposed by Patrick and Radhia Cousot [5] [6] [7]. In [3], we presented a first proposal in this direction, focusing on partitions based on categorical values present in the table and generating a watermark as a permutations of the ordering of the tuples. Here we go one step further, by removing the constraints on the presence of categorical values in the table: we consider now any partitioning, and we generate out of it a binary image which contains the same number of rows but one less number of columns of the actual partition. The contribution of this paper is thus much more sophisticated and completely orthogonal to [8]. The binary image serves the purpose of the ownership proof and tamper detection of the associated partition. We prove that this is an abstract representation of the actual partition by showing the existence of a Galois connection between the concrete and the abstract partition (i.e. the binary image). Therefore, any modification in the concrete partition will reflect in the abstract counterpart. We state the soundness condition regarding this alteration. The robustness of the proposed watermarking obviously depends on the size of the individual groups, so it is specifically designed for large databases. The resulting watermark is robust against various forms of malicious attacks and updates to the data in the table.

The paper is organized as follows. In Section 2, we formalize the definition of tables in relational database, and a formal definition of watermarking process of a table in relational database is given. Section 3 illustrates how distortions and watermarking are related. In Section 4, we show the data partitioning algorithm. In Section 5, we present the watermark generation algorithm for a data partition, and we explain why this watermark is considered as an abstract representation of the concrete partition. In Section 6, we propose the watermark detection algorithm. The robustness of the technique is discussed in Section 7. Finally, we draw our conclusions in Section 8.

2 Preliminaries

This section contains an overview of Galois connection and some formal definitions of tables in relational database and database watermarking [4] [10].

Definition 2.1 (Partial Orders)

A partial order on a set D is a relation $\sqsubseteq \in \wp(D \times D)$ with the following properties:

- $\forall d \in D : d \sqsubseteq d$ (reflexivity)
- $\forall d, d' \in D : (d \sqsubseteq d') \wedge (d' \sqsubseteq d) \Longrightarrow (d = d')$ (antisymmetry)
- $\forall d, d', d'' \in D : (d \sqsubseteq d') \wedge (d' \sqsubseteq d'') \Longrightarrow (d \sqsubseteq d'')$ (transitivity)

A set with a partial order defined on it is called partially ordered set, poset. Following are definitions of some commonly used terms with respect to Partial order (L, $\sqsubseteq$).

Definition 2.1.1 (Lower Bound)

X $\subseteq$ L has l $\in$ L as lower bound if $\forall$ l' $\in$ X : l $\sqsubseteq$ l'.

Definition 2.1.2 (Greatest Lower Bound)

X $\subseteq$ L has l $\in$ L as greatest lower bound l if $l_0 \sqsubseteq$ l whenever l_0 is another lower bound of X.It is represented by the operator $\sqcap$. glb(X)= $\sqcap$ X.

Definition 2.1.3 (Upper Bound)

X $\subseteq$ L has l $\in$ L as upper bound if $\forall$ l' $\in$ X : l' $\sqsubseteq$ l.

Definition 2.1.4 (Least Upper Bound)

X $\subseteq$ L has l $\in$ L as least upper bound if l $\sqsubseteq l_0$ whenever l_0 is another upper bound of X. It is represented by the operator $\sqcup$. lub(X)= $\sqcup$ X.

Definition 2.2 (Complete Lattice)

A complete lattice (L, $\sqsubseteq$, $\sqcup$, $\sqcap$, $\top$, $\bot$) is a partial ordered set (L, $\sqsubseteq$) such that every subset of L has a least upper bound as well as a greatest lower bound.

- The greatest element $\top = \sqcup L$

- The least element $\bot = \sqcap L$

For instance (L, $\sqsubseteq$, $\sqcup$, $\sqcap$, $\top$, $\bot$) where L = {1, 2, 3, 4, 6, 9, 36}, the partial order $\sqsubseteq$ is defined by $n \sqsubseteq m \Leftrightarrow (m \bmod n = 0)$, $\bot$= 1 and $\top$ = 36 is a complete lattice. It can be represented using Hasse diagram as shown below

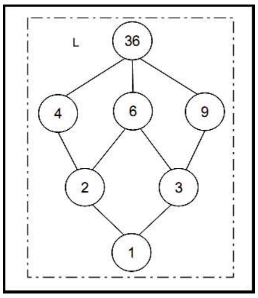

Fig. 2. Complete Lattice

Definition 2.3 (Galois Connection)
Let C (concrete) and A (abstract) be two domains (or lattices). Let α : C→ A and γ : A→C be an abstraction function and a concretization function, respectively. The pair of functions (α,γ) form a Galois Connection if:

- both α and γ are monotone (order preserving).
- ∀ a ∈ A : $\alpha(\gamma(a)) \sqsubseteq a$
- ∀ c ∈ C : $c \sqsubseteq \gamma(\alpha(c))$

α and γ uniquely determine each other.
Definition 2.4 (Function)
Let Π_i be the projection function which selects the i-th coordinate of a pair. F is a function over the set A into set B $\Leftrightarrow F \in \wp(A \times B) \bigwedge (\forall p_1, p_2 \in F : p_1 \neq p_2 \Rightarrow \Pi_1(p_1) \neq \Pi_1(p_2)) \bigwedge \{\Pi_1(p) | p \in F\} = A$.

Definition 2.5 (Set Function)
A set function is a function in which every range element is a set. Formally, let F is a set function ⇔ F is a function and ($\forall c \in dom(F) : F(c)$ is a set).
For instance, we can express information about companies and their locations by means of a set function over the domain {Company, Location}. Namely,
(Company;{'Natural Join', 'Central Boekhuis', 'Oracle', 'Remmen & De Brock'})
(Location, {'New York', 'Venice', 'Paris'}).

Definition 2.6 (Table)
Given two sets H and K, a table over H and K is a set of functions T over the same set H and into the same set K. i.e. ∀ t ∈ T: t is a function from H to K.
For instance consider a table containing data on employees:

Table 1. EMPLOYEE

emp_no	**emp_name**	**emp_rank**
100	John	Manager
101	David	programmer
103	Albert	HR

The table is represented by the set of functions t_1, t_2, t_3 where dom(t_i) = emp_no, emp_name, emp_rank and for instance t_1(emp_name) = John.

There is a correspondence between tuples and functions. For instance, t_1 corresponds to the following tuple: (emp_no, 100), (emp_name, John), (emp_rank, manager). The first coordinates of the ordered pairs in a tuple are referred to as the attributes of that tuple.

Definition 2.7 (Watermarking)
A watermark W for a table T over H into K, is a predicate such that W(T) is true and the probability of $W(T')$ being true with $T' \in \wp(H \times K) \backslash T$ is negligible.

3 Distortions by Watermarking

It is often hard to define the available *bandwidth* for inserting the watermark directly. Instead, allowable distortion bounds [14][15]for the input data can be defined in terms of consumer metrics. If the watermarked data satisfies the metrics, then the alterations induced by the insertion of the watermark are considered to be acceptable. One such simple yet relevant example for numeric data, is the case of maximum allowable mean squared error (MSE), in which the usability metrics are defined in terms of mean squared error tolerances as

$$(S_i - V_i)^2 < t_i, \forall i = 1...n \quad and \tag{1}$$

$$\sum_i^n (S_i - V_i)^2 < t_{max} \tag{2}$$

where
$\mathbb{S} = s_1, ..., s_n \subset \mathbb{R}$, is the data to be watermarked,
$\mathbb{V} = v_1, ..., v_n$ is the result,
$\mathbb{T} = t_1, ..., t_n \subset \mathbb{R}$ and
$t_{max} \in R$ define the guaranteed error bounds at data distribution time.

In other words $\mathbb{T}$ defines the allowable distortions for individual elements in terms of MSE and t_{max} its overall permissible value.

However, specifying only allowable change limits on individual values, and possibly an overall limit, fails to capture important *semantic features* associated with the data, especially if the data is structured. Consider for example, the *age* data in an Indian context. While a small change to the age values may be acceptable, it may be critical that individuals that are younger than 21 remain so even after watermarking if the data will be used to determine behavior patterns for under-age drinking. Similarly, if the same data were to be used for identifying legal voters, the cut-off would be 18 years. In another scenario, if a relation contains the start and end times of a web interaction, it is important that each tuple satisfies the condition that the end time be later than the start time. For some other application it may be important that the relative ages, in terms of which one is younger, not change. Other examples of constraints include: *uniqueness*, each value must be unique; *scale*, the ratio between any two number before and after the change must remain the same; and *classification*, the objects must remain in the same class (defined by a range of values) before and after the watermarking. As is clear from the above examples, simple bounds on the change of numerical values are often not sufficient to prevent side effects of a watermarking operation.

4 Data Partitioning

In this section we present a data partitioning algorithm that partitions the data set based on a secret key $\Re$. The data set D is a database relation with scheme D(P,$C_0, ..., C_{v-1}$), where P is the primary key attribute, $C_0, ..., C_{v-1}$ are it's v attributes, and η is the number of tuples in D. The data set D is to be partitioned into m non overlapping partitions, $[S_0], ..., [S_{m-1}]$, such that each partition $[S_i]$ contains on average $(\frac{\eta}{m})$ tuples from the data set D. Partitions do not overlap, i.e., for any two partitions $[S_i]$ and $[S_j]$ such that $i \neq j$ we have $[S_i] \cap [S_j] = \emptyset$. In order to generate the partitions, for each tuple $r \in D$, the data partitioning algorithm computes a message authenticated code (MAC) using HMAC [12].

Using the property that secure hash functions generate uniformly distributed message digests this partitioning technique places $(\frac{\eta}{m})$ tuples, on average, in each partition. Furthermore, an attacker cannot predict the tuples-to-partition assignment without the knowledge of the secret key $\Re$ and the number of partitions m which are kept secret. Keeping it secret makes it harder for the attacker to regenerate the partitions. The partitioning algorithm is described below:

Algorithm 1 : **get_partitions**$(\mathbf{D}, \Re, \mathbf{m})$

```
for each tuple $r \in D$ do
    partition ← HMAC($\Re$ | r.P) mod m
    insert r into $S_{partition}$
end for
return $(S_0, ..., S_{m-1})$
```

Consider the lattice A = $\langle \mathbb{N}, \bigcup\{\perp, \top\}, \sqsubseteq \rangle$, where $\perp \sqsubseteq i \sqsubseteq \top$ and $\forall$ i, j $\in \mathbb{N}$, $i \neq j$, i and j are uncomparable with $\sqsubseteq$. The lattice is shown in Figure 3.

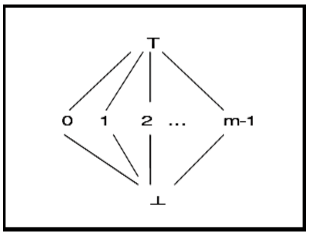

Fig. 3. Lattice of the abstract domain

Given a data set $D \in (P \times C_0 \times C_1 \times ... \times C_{v-1})$ and m partitions $\{[S_i], 0 \leq i \leq (m-1)\}$, for each set T ⊆ D, and given a set of natural number i ∈ ℕ, we can define a concretization map γ as follows:
$\gamma(\top) = D$
$\gamma(\bot) = \emptyset$

$$\gamma(i) = \begin{cases} T \subseteq D & \text{if } \forall t \in T : i = HMAC(\Re|t.P) \ \ mod \ \ m \\ \quad\emptyset & \text{Otherwise} \end{cases} \tag{3}$$

The best representation of a set of tuples is captured by the corresponding abstraction function α :

$$\alpha(T) = \begin{cases} \bot \text{ if } S = \emptyset \\ \ i \text{ if } \forall t \in T : HMAC(\Re|t.P) \ \ mod \ \ m = i \\ \top \ Otherwise \end{cases} \tag{4}$$

The two functions α and γ described above yield a Galois connection [6] between D, i.e. the actual data set and the lattice depicted in figure 2.

5 Watermark Generation

We are interested in a watermark generation process starting from a partition $[S_k]$ $0 \leq k \leq m]$, in a relational database table . The partitioning can be seen as a virtual grouping which does not change the physical position of the tuples. Let the owner of the relation D possess a watermark key $\Re$, which will be used in both watermark generation and detection. In addition, the key should be long enough to thwart brute force guessing attacks to the key. A cryptographic pseudo random sequence generator [13] G is seeded with the concatenation of watermark key $\Re$ and the primary key r.P for each tuple $r \in S_k$, generating a sequence of numbers.The MSBs (most significant bits) of the selected values are used for generating the watermark. Formally, the watermark W_k corresponding to the k^{th} partition $[S_k]$ is generated as follows,

Algorithm 2 : $\mathbf{genW(S_k, \Re)}$

```
1: for each tuple r $\in S_k$ do
2:   construct a row t in $W_k$
3:   for (i=0; i< $v$; i=i+1) do
4:     j= $G_i(\Re, r.P)$ mod $v$
5:     $t.W_k^i$ = MSB of the $j^{th}$ attribute in r
6:     delete the $j^{th}$ attribute from r
7:   end for
8: end for
9: return($W_k$)
```

Let us illustrate the above algorithm for a single tuple in any hypothetical partition of a table Employee = $(emp\_id, emp\_name, salary, location, position)$, where *emp_id* is the primary key which is concatenated along with the private key $\Re$ as in line 2 in the above algorithm to select random attributes. Here (1011) is the generated watermark for the tuple (Bob, 10000, London, Manager), where MSBs 1, 1, 1 and 0 are associated to Bob, 10000, London and Manager respectively.

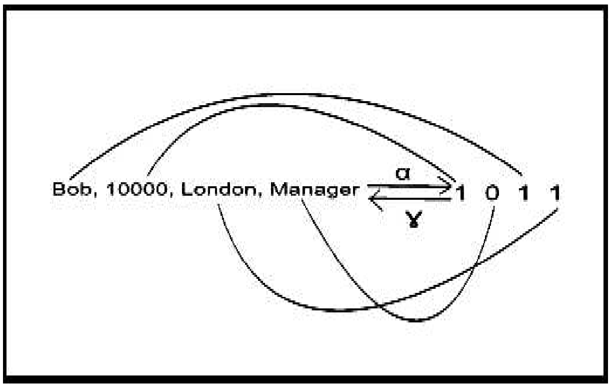

Fig. 4. Watermark generation for a single tuple

So if there are n number of tuples in the partition $[S_k]$, $genW$ generates a binary image $W_k^{n,v}$ as a watermark for $[S_k]$ partition. The whole process does not introduce any distortion to the original data. The use of MSBs is for thwarting potential attacks that modify the data. Namely, subset alteration attack where the attacker alters the tuples of the database through operations such as linear transformation. The attacker hopes by doing so to erase the watermark from the database.

5.1 Functional Abstraction

Theorem 1 (Galois Connection)

Given a table D $\subseteq C_0 \times C_1 \times C_2 \times \ldots C_{v-1}$, let $\mathbb{B}^v$ is the set of all binary sequences of length v. We can define abstraction and concretization function between $\wp\ (C_0 \times C_1 \times C_2 \times \ldots C_{v-1})$ and $\wp(\mathbb{B}^v)$ as follows
$\alpha(S) = \{genW(\text{S}, \Re)(\text{r}) \mid \text{r} \in \text{S}\}$
$\gamma(W) = \{\text{t} \in \text{S} \subseteq \text{D} \mid genW(\text{S}, \Re)(\text{t}) \in \text{W}\}$. Then α and γ form a Galois connection [6].

Proof:

$\alpha(S) \subseteq W$
$\Leftrightarrow \{genW(\text{S}, \Re)(\text{r}) \mid \text{r} \in \text{S}\} \subseteq \text{W}$
$\Leftrightarrow \forall\ \text{r} \in \text{S} : genW(\text{S}, \Re)(\text{r}) \in \text{W}$
$\Leftrightarrow \text{S} \subseteq \{\ \text{r} \mid genW(\text{S}, \Re)(\text{r}) \in \text{W}\}$
$\Leftrightarrow \text{S} \subseteq \gamma(\text{W})$.

The data set (table) $D \subseteq \wp(C_0 \times C_1 \times \ldots C_{v-1})$ and the watermark $W \subseteq \wp(\mathbb{B}^v)$. By Theorem 1 (D, W, α, γ) form a Galois Connection. The function $genW : D \rightarrow W$ is the watermark generation function described above. $\forall t \in D$, $f_{alt} : D \rightarrow D$ and $\forall t^{\#} \in W$, $f^{\#}_{alt} : W \rightarrow W$ are the alteration functions that alter the tuples in both concrete and abstract domain, respectively. Therefore the soundness condition with respect to the alteration function can be stated as follows:

$$\forall t \in D : \alpha(f_{alt}(t)) \sqsubseteq f^{\#}_{alt}(\alpha(t))$$

$$\begin{tikzcd}
t \arrow[r, "f_{alt}"] \arrow[d, "\alpha"'] & t_{alt} \arrow[d, "\alpha"] \\
t^{\#} \arrow[r, "f^{\#}_{alt}"'] & t^{\#}_{alt} \sqsupseteq \alpha(t_{alt})
\end{tikzcd}$$

Fig. 5. Soundness

This means that, the proposed watermark process is sound whenever the diagram above commutes.

6 Watermark Detection

A very important problem in a watermarking scheme is synchronization, that is, we must ensure, that the watermark extracted is in the same order as that generated. If synchronization is lost, even if no modifications have been made, the embedded watermark cannot be correctly verified. In watermark detection, the watermark key $\Re$ and watermark W_k are needed to check a suspicious partition S'_k of the suspicious database relation D'. It is assumed that the primary key attribute has not been changed or else can be recovered.

Algorithm 3 : $\mathbf{detW(S'_k, \Re, W_k)}$

```
 1: matchC=0
 2: for each tuple r $\in S_k$ do
 3:    get the $r^{th}$ row, t of $W_k$
 4:    for (i=0; i < $v$; i = i+1) do
 5:       j= $G_i(\Re, r.P)$ mod $v$
 6:       if $t.W_k^i$ = MSB of the $j^{th}$ attribute in r then
 7:          matchC = matchC + 1
 8:       end if
 9:       delete the $j^{th}$ attribute from r
10:    end for
11: end for
12: if matchC=$\omega$ then
13:    // $\omega$ = number of rows × number of columns in $W_k$
14:    return true
15: else
16:    return false
17: end if
```

The variable matchC counts the total number of correct matches. We consider the watermark $W_k^{t,i}$, t= 1 to q_k(number of tuples in S_k) and i= 1 to v (number of attributes in relation). At the statement number 6 the authentication is checked by comparing the generated watermark bitwise. And after each match matchC is increased by 1. Finally at statement number 12, the total match is compared to the number of bits in the watermark image W_k associated with partition S_k to check the final authentication.

7 Robustness

We analyze the robustness of our scheme by Bernoulli trials and binomial probability as in [6]. Repeated independent trials in which there can be only two outcomes are called Bernoulli trials in honor of James Bernoulli (1654-1705).The probability that the outcome of an experiment that consists of n Bernoulli trials has k successes and n - k failures is given by the binomial distribution

$$b(n,k,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \qquad 0 \leq k \leq n$$

where the probability of success on an individual trial is given by p.

The probability of having at least k successes in n trials, the cumulative binomial probability, can be written as

$$B(n,k,p) = \sum_{i}^{k} b(n,i,p)$$

We will discuss our robustness condition based on two parameters *false hit* and *false miss.*

7.1 False hit

False hit is the probability of a valid watermark being detected from non-watermarked data. The lower the false hit, the better the robustness. When the watermark detection is applied to non-watermarked data, each MSB in data has the same probability $\frac{1}{2}$ to match or not to match the corresponding bit in the watermark. Assume that the non-watermarked data partition S_q has the same number q of tuples and has the same primary keys as the original data. Let $\omega = vq$ is the size of the watermark. where v is the no of attributes being watermarked and r is the number of tuples in partition S_r. The false hit is the probability that at least $\frac{1}{\mathfrak{T}}$ portion of ω can be detected from the non-watermarked data by sheer chance. When $\mathfrak{T}$ is the watermark detection parameter. It is used as a tradeoff between false hit and false miss. Increasing $\mathfrak{T}$ will make the robustness better in terms of false hit. Therefore, the false hit F_h can be written as

$$F_h = B(\omega, \lfloor \frac{\omega}{\mathfrak{T}} \rfloor, \frac{1}{2})$$

7.2 False miss

False miss is the probability of not detecting a valid watermark from watermarked data that has been modified in typical attacks. The less the false miss, the better the robustness. For tuple deletion and attribute deletion, the MSBs in the deleted tuples or attributes will not be detected in watermark detection; however, the MSBs in other tuples or attributes will not be affected. Therefore, all detected MSBs will match their counterparts in the public watermark, and the false miss is zero. Suppose an attacker inserts ς new tuples to replace ς watermarked tuples with their primary key values unchanged. For watermark detection to return a false answer, at least $\frac{1}{\mathfrak{T}}$ MSBS of those newly added tuples (which consists of $\upsilon\varsigma$ MSBs) must not match their counterparts in the watermark (which consists of $\omega = \upsilon q$ bits, if the partition contains q tuples). also in this case $\mathfrak{T}$ is the watermark detection parameter, used as a tradeoff between false hit and false miss. Increasing $\mathfrak{T}$ will make the robustness worse in terms of false miss. Therefore, the false miss F_m for inserting ς tuples can be written as

$$F_m = B(\upsilon\varsigma, \lfloor\frac{\upsilon\varsigma}{\mathfrak{T}}\rfloor, \frac{1}{2})$$

The formulae F_h and F_m together, give us a measure of the robustness of the watermark.

8 Conclusions

As a conclusion, let us stress the main features of the watermark technique presented in this paper,

- it does not depend on any particular type of attributes (categorical, numerical);
- it is partition based, we are able to detect and locate modifications as we can trace the group which is possibly affected when a tuple t_m is tampered;
- neither watermark generation nor detection depends on any correlation or costly sorting among data items. Each tuple in a table is independently processed; therefore, the scheme is particularly efficient for tuple oriented database operations;
- it does not modify any database item; therefore it is distortion free.

Acknowledgements

Work partially supported by Italian MIUR COFIN '07 project "SOFT".

References

1. Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *In 28th Int Conference on Very Large Databases, Hong Kong*, pages 155–166, 2002.
2. Ali Al-Haj and Ashraf Odeh. Robust and blind watermarking of relational database systems. *Journal of Computer Science*, pages vol.4(12): 1024–1029, 2008.
3. Sukriti Bhattacharya and Agostino Cortesi. A distortion free watermarking framework for relational databases. In *Proc. 4th International Conference on Software and Data technology, ICSOFT'09, Sofia, Bulgaria*, pages 229–234, 2009.

4. Christian Collberg and Clark Thomborson. Watermarking, tamper-proofing, and obfuscation - tools for software protection. *IEEE Trans. Software Eng*, 28:735–746, 2000.
5. Patrick Cousot. Abstract interpretation based formal methods and future challenges. In *Informatics 10 years back, 10 years ahead, vol. 2000 Of Lecture Notes in Computer Science*, pages 138–156. Springer-Verlag, 2001.
6. Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL '77: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM Press, 1977.
7. Patrick Cousot and Radhia Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2:511–547, 1992.
8. Ingemar Cox. *Digital Watermarking: Principles and Practice.* Morgan Kaufman Publ. Inc., 2001.
9. David Gross-Amblard. Query-preserving watermarking of relational databases and xml documents. In *Proc. of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 191–201, 2003.
10. Lex de Haan and Toon Koppelaars. *Applied Mathematics for Database Professionals.* Apress, Berkely, CA, USA, 2007.
11. Zoran Jajodia Sushil Johnson Neil F. Duric. *Information hiding: steganography and watermarking. Attacks and countermeasures.* Kluwer Academic Publishers, 2000.
12. Norman Y. Mineta, Cheryl L. Shavers, Raymond G. Kammer, and William Mehuron. The keyed-hash message authentication code (HMAC). FEDERAL INFORMATION PROCESS STANDARDS PUBLICATION, 2002.
13. Bruce Schneier. *Applied Cryptography.* John Wiley & Sons, 1996.
14. Radu Sion. Proving ownership over categorical data. volume 0, page 584, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
15. Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data. volume 16, pages 1509–1525, Los Alamitos, CA, USA, 2004. IEEE Computer Society.