Distortion-Free Authentication Watermarking

Sukriti Bhattacharya and Agostino Cortesi

Ca' Foscari University of Venice
Via Torino 155, 30170 Venezia, Italy
{sukriti,cortesi}@unive.it

Abstract. In this paper[1] we introduce a distortion free watermarking technique that strengthen the verification of integrity of the relational databases by using a public zero distortion authentication mechanism based on the Abstract Interpretation framework.

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. Watermarking techniques apply to various types of host content. Digital multimedia watermarking technology was suggested in the last decade to embed copyright information in digital objects such images, audio and video. Most watermarking research concentrated on watermarking multimedia data objects such as still images [7] and video [1], [2], [3] and audio [4], [5], [6]. However, 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. Techniques developed for multimedia data cannot be directly used for watermarking relational databases, because relational and multimedia data differ in a number of important respects [8].

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. However some applications in which relational data are involved cannot tolerate any permanent distortions and data's integrity needs to be authenticated. In this paper we further strengthen this approach by refining the distortion free watermarking algorithm [9] for relational databases and discuss it in abstract interpretation framework proposed by Patrick Cousot and Radhia Cousot [10], [11], [12].

In Section 2, the basic database watermarking process is described along with the most relevant existing database watermarking techniques. In Section 3 we introduce the formal definitions and mathematical notations that we are going

[1] This paper is a revised and extended version of [17]

to use in the remaining sections. The actual watermarking process is introduced in Section 4. We illustrate database authentication issues in Section 5. Section 6 discusses the robustness of the algorithm to handle different watermarking attacks. In Section 7, we conclude by discussing the main advantages of our scheme.

2 Basic Watermarking Process

The motivation for database watermarking is to protect databases, especially those published online as parametric specifications, surveys or life sciences biometric data, from tampering and pirated copies. Database watermarking consists of two basic processes: watermark insertion and watermark detection, 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.

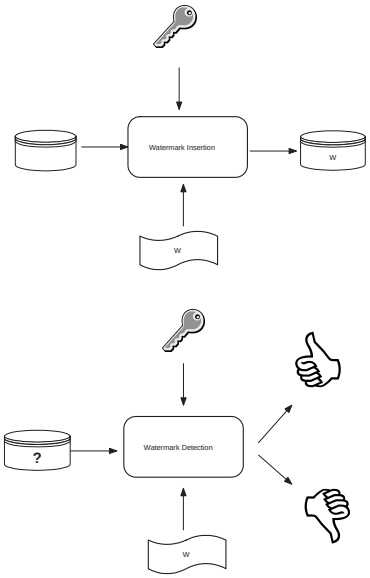

Fig. 1. Basic database watermarking process

The first well-known database watermarking scheme was proposed by Agrawal and Kiernan [8] for watermarking numerical values in relational databases. Agrawal

and Kiernans scheme has been extended by Ng and Lau to watermark XML data [13]. This scheme is further extended and deployed on a XML compression system. In [14] Gross-Amblard introduce interesting theoretical results investigating alterations to relational data in a consumer-driven framework in which a set of parametric queries are to be preserved up to an acceptable level of distortion. Agrawal and Kiernan's scheme cannot be directly applied to watermarking categorical data. To solve this problem, Sion [25] proposed to watermark a categorical attribute by changing some of its values to other values of the attribute if such change is tolerable in certain applications. In [15], Li, Guo, and Jajodia introduced a distortion-free scheme for watermarking categorical data. Guo et al. [16] proposed another fragile watermarking scheme that can further improve the precision in tamper localization, assuming that the database relation to be watermarked has numerical attributes and that the errors introduced in two least significant bits of each value can be tolerated. Recent works and surveys can be found in [23], [24].

3 Preliminaries

We shall use the capital letters at the beginning of the alphabet to denote single attributes $(A, B...)$, and $dom(A)$ will be the domain of the attribute A. For sets of attributes we shall use the letters at the end of the alphabet $(X, Y...)$, and $dom(X)$ will be the cartesian product of all the attribute domains $A \in X$. $R, S...$ will denote relational schemes (sets of attributes). $\mathcal{A}(R)$ is a set of attributes and $\mathcal{T}(R)$ is the set of tuples, over which relation schema R is defined. Relations, i.e, instances of relational schemes, will be denoted by small letters such as $r, s...$ and tuples by $t, u....$ To emphasize that the attribute A belongs to the relation r, we shall use the notation $r.A$. The value of an attribute A in a tuple t, will be represented by $t[A]$. A relational table R is a function,

$$\forall t \in \mathcal{T}(R) \wedge \forall A \in \mathcal{A}(R) : \mathcal{T}(R) \times \mathcal{A}(R) \rightarrow t[A] \in dom(A)$$

The symbol $\times$ stands for the usual cartesian product.

Example 1. Consider the EMPLOYEE table (Table 1) as example,

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

Table 1. EMPLOYEE relation

– $\mathcal{T}(EMPLOYEE)$: $\{t_1, t_2, t_3\}$.

– $\mathcal{A}(EMPLOYEE)$: $\{emp\_no, emp\_name, emp\_rank\}$.
– EMPLOYEE:
$t_1(emp\_no) = 100;\ t_1(emp\_name) = Jhon;\ t_1(emp\_rank) = Manager$
$t_2(emp\_no) = 101;\ t_2(emp\_name) = David;\ t_2(emp\_rank) = Programmer$
$t_3(emp\_no) = 103;\ t_3(emp\_name) = Albert;\ t_3(emp\_rank) = HR$

Now, let us consider the definition of watermarking in case of relational databases,

Definition 1. (Watermarking) *A watermark W for a relation R is a predicate such that $W(R)$ is true and the probability of $W(R')$ being true with $R' \in \wp(\mathcal{T}(R') \times \mathcal{A}(R'))\backslash R$ is negligible.*

Definition 2. (Partial order) *A binary relation $\sqsubseteq$ on a set C is a partial order on C if the following properties hold:*

– *reflexivity:* $\forall x \in C : x \sqsubseteq x$;
– *antisymmetry:* $\forall x, y \in C : x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y$;
– *transitivity:* $\forall x, y, z \in C : x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$;

Definition 3. (Complete lattice) *A complete lattice (L, $\sqsubseteq$, $\sqcup$, $\sqcap$, $\top$, $\bot$) is a partial ordered set (L, $\sqsubseteq$) such that every subsets of L, has a least upper bound as well as a greatest lower bound.*

– *The greatest element* $\top = \sqcap\emptyset = \sqcup L$
– *The least element* $\bot = \sqcap L = \sqcup\emptyset$

Definition 4. (Galois connection) *Two posets $\langle C, \sqsubseteq_1\rangle$ and $\langle L, \sqsubseteq_2\rangle$ and two monotone functions $\alpha : C \to L$ and $\gamma : L \to C$ such that:*

– $\forall c \in C : c \sqsubseteq_1 \gamma(\alpha(c))$
– $\forall l \in L : \alpha(\gamma(l)) \sqsubseteq_2 l$

form a Galois connection, equivalently denoted by (C, α, γ, L).

A *Galois connection* (C, α, γ, L) where $\forall l \in L : \alpha(\gamma(l)) = l$ is called a *Galois insertion*. Figure 2 provides a graphical representation of the Galois connection features.

4 Distortion Free Database Watermarking

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 data are 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 underage 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

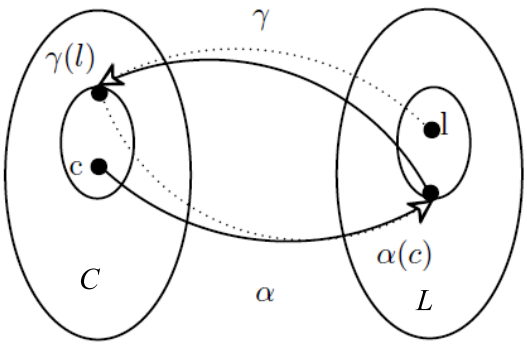

Fig. 2. Galois connection

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, do not change. It is clear from the above examples, that simple bounds on the change of numerical values are often not sufficient to prevent side effects of a watermarking insertion.

Our proposed watermarking technique is partition based. The partitioning can be seen as a virtual grouping, which neither change the value of the table's elements nor their physical positions. This partitioning phase is interpreted in abstract interpretation framework as *relational table abstraction*. 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. This is called *partition abstracting*. The main idea is to generate a image (binary [9] or grey scale [17]) of the partition as a watermark of that partition, that serves as ownership proof (certificate) as well as tamper detection, namely *authentication* phase. The overall scheme is depicted in Figure 3.

4.1 Partitioning

In this section we will describe three partitioning algorithms. We recall the first two forms [22], [9]and [17], respectively. Here, we propose a new one based on pattern tabaleau, and describe the *abstraction* associated with each partitioning scheme.

Partition Based on Categorical Attribute Let R be a given relational data table and $C \in \mathcal{A}(R)$ be a categorical attribute. The (finite) value set $\mathcal{V} \subseteq dom(C)$ is the set of values of C that are actually present in R. We can partition the tuples in R by grouping the values of attribute C [22] as

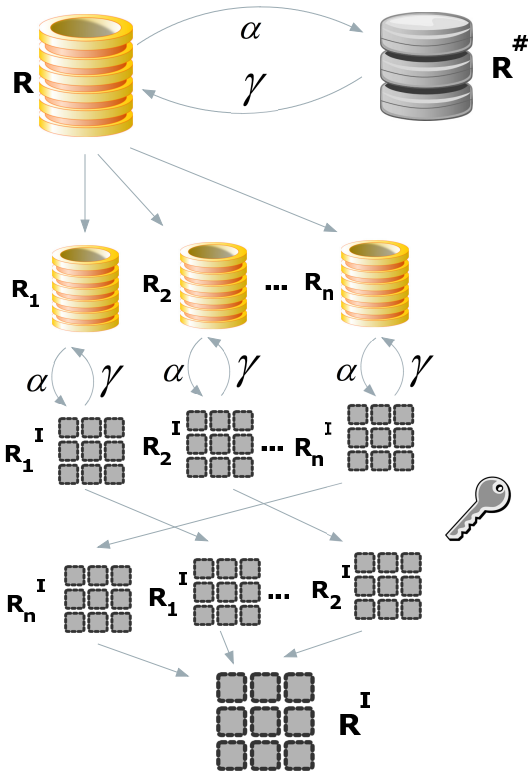

Fig. 3. Block diagram of the over all process

$$P = \{[v_i] : 1 \leq i \leq N\}, \text{ where } \forall t \in \mathcal{T}(R) : t[C] = v_i \Leftrightarrow t \in [v_i].$$

The frequency q_i of v_i is the number of tuples in $[v_i]$. The data distribution of C in R is the set of pairs $\tau = \{(v_i, f_i)|1 \leq i \leq N\}$. So the entire database can be partitioned into N fixed mutual exclusive areas based on each categorical value v_i.

Abstraction Given a relation R a categorical attribute $C \in \mathcal{A}(R)$ and $P = \{[v_i] : 1 \leq i \leq N\}$, for each set $S \subseteq R$, We can define a concretization map γ_x as follows:

$$\begin{cases} \gamma_x(v_i, h) = S \subseteq R \mid \forall t \in S : t \in [v_i] \wedge \text{size of S is h} \\ \gamma_x(\top) = T \\ \gamma_x(\bot) = \emptyset \end{cases}$$

The best representation of a set of tuples with attribute C is captured by the corresponding abstraction function α_x :

$$\alpha_x(S) = \begin{cases} (v_i, h) \text{ if } \forall t \in \mathcal{T}(S) : t[C] = v_i \wedge \text{size of S is h} \\ \top \text{ if } \exists t_1, t_2 \in S : t_1[C] \neq t_2.x \\ \bot \text{ if } S = \emptyset \end{cases}$$

We may prove that (α_x, γ_x) form a Galois insertion with α_x monotone and γ_x weakly monotone, i.e. $(v, u) \leq (v, m) \Rightarrow (\cup\gamma(v, u)) \subseteq (\cup\gamma(v, m))$.

Secret Partitioning The second partitioning algorithm [9], [17], partitions the relational data table based on a secret key $\Re$ with P as the primary key attribute and N is the number of tuples in R. R is partitioned into m non overlapping partitions, $[S_0], ..., [S_{m-1}]$, such that each partition $[S_i]$ contains on average $(\frac{N}{m})$ tuples from R. 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 $r \in \mathcal{T}(R)$, the data partitioning algorithm computes a message authenticated code (MAC) using HMAC [18].

Using the property that secure hash functions generate uniformly distributed message digests, this partitioning technique places $(\frac{N}{m})$ 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 reported in Table 2.

$get\_partitions(R, \Re, m)$
for each tuple $r \in \mathcal{T}(R)$ do $partition \leftarrow HMAC(\Re \mid r[P])\ mod\ m$ insert r into $S_{partition}$ return$(S_0, ..., S_{m-1})$

Table 2. Secret partitioning

Abstraction Given R and m partitions $\{[S_i], 0 \leq i \leq (m-1)\}$, for each set $T \subseteq R$, and given a set of natural number $i \in \mathbb{N}$, we can define a concretization map γ as follows:
$\gamma(\top) = R$; $\gamma(\bot) = \emptyset$

$$\gamma(i) = \begin{cases} T \subseteq R & \text{if } \forall t \in T : i = HMAC(\Re|t[P])\ \ mod\ \ m \\ \emptyset & \text{Otherwise} \end{cases} \quad (1)$$

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

$$\alpha(T) = \begin{cases} \perp \text{ if } S = \emptyset \\ i \text{ if } \forall t \in T : HMAC(\Re \mid t[P]) \ \ mod \ \ m = i \\ \top \ Otherwise \end{cases} \tag{2}$$

The main advantage of this partitioning is that, it is not limited to any particular type of attribute, like categorical or numerical attribute.

Partitioning Based on Pattern Tableau Using the intersection operator over tuples we could build the tuples lattice of a relation. A closed tuple will thus subsume all tuples agreeing on the same values, i.e. the values of non empty variables in the closed tuple. This notion of set of tuples agreeing on the same values for a given set of attributes X has already been defined in database theory for horizontal decomposition purposes [20]. Let us consider the following definitions to define the partitioning.

Definition 5. (Pattern tableau) *A pattern tableau $R^{\#}$ with all attributes from $\mathcal{A}(R)$, where each row $t_p \in \mathcal{T}(R^{\#})$ and each attribute $A \in \mathcal{A}(R)$, $t_p[A]$ is either:*

- *a constant $a \in dom(A)$.*
- *an empty variable $\top$ which indicates that the attribute does not contribute to the pattern.*

Definition 6. (X-complete property) *The relation r is said to be X-complete if and only if $\forall t_1, t_2 \in r$ we have $t_1[X] = t_2[X]$.*

Informally, a relation is X-complete if all tuples agree on the attributes X.

Definition 7. (X-complete-pattern) *We call X-complete-pattern of an X-complete relation r, denoted by $\mathscr{P}(X, r)$, the pattern tuple on which tuples of r agree.*

Since r is X-complete, its X-complete-pattern defines at least the attributes in X, i.e. those attributes do not have the $\top$ value.

Definition 8. (X-complete horizontal decomposition) *The set of all X-complete fragment relations of r, $\mathscr{R}_X(r)$ is defined formally as $\mathscr{R}_X(r) = \{r' \subseteq r | r \ \ is \ \ X-complete\}$.*

Definition 9. (Set of X-patterns) *The set of all X-complete-patterns of an X-complete decomposition, $\Gamma(X, r)$ is formally defined as $\Gamma(X, r) = \{\mathscr{P}(X, r') | r' \in \mathscr{R}_X(r)\}$.*

Example 2. Consider Table 3,

- The $AB - complete$ horizontal decomposition of r is $\mathscr{R}_{AB}(r) = \{\{t_1, t_2\}, \{t_3, t_4\}, \{t_5, t_6, t_7, t_8\}, \{t_9, t_{10}\}\}$.
- The set of $AB - complete$ patterns is $\Gamma(AB, r) = \{(a_1, b_1, c_1, d_1, \top, f_1); (a_2, b_1, c_2, d_2, \top, f_1);\ (a_2, b_2, \top, \top, e_1, f_2);\ (a_1, b_2, c_2, d_1, \top, f_2)\}$.

T	A	B	C	D	E	F
t_1	a_1	b_1	c_1	d_1	e_1	f_1
t_2	a_1	b_1	c_1	d_1	e_2	f_1
t_3	a_2	b_1	c_2	d_2	e_2	f_1
t_4	a_2	b_1	c_2	d_2	e_3	f_1
t_5	a_2	b_2	c_2	d_2	e_1	f_2
t_6	a_2	b_2	c_2	d_1	e_1	f_2
t_7	a_2	b_2	c_1	d_1	e_1	f_2
t_8	a_2	b_2	c_1	d_2	e_1	f_2
t_9	a_1	b_2	c_2	d_1	e_2	f_2
t_{10}	a_1	b_2	c_2	d_1	e_1	f_2

Table 3. An instance relation r of the schema R

Partitions are the $X-$*complete* horizontal decomposition of R, $\mathscr{R}_X(R)$, where $X \subseteq A$, such that the following two conditions must be satisfied,

- $\bigcup_{\forall r \in \mathscr{R}_X(R)} r = R$
- $\bigcap_{\forall r \in \mathscr{R}_X(R)} r = \emptyset$

Abstraction Given a relation R and m partitions ($X -$ *complete* horizontal decompositions of R) $\{[R_i], 1 \leq i \leq m\}$, then $r \subseteq R$, and given a set of $X -$ *complete* patterns, we can define a concretization map γ as follows:

$\gamma(\top) = R$; $\gamma(\bot) = \emptyset$
$\gamma(\mathscr{P}(X, r)) =$

$$\begin{cases} r \subseteq R & \forall t \in T(r), \forall a \in A : \textit{ either } t(a) = \mathscr{P}(X,r)(a) \textit{ or } \mathscr{P}(X,r)(a) = \top \\ \emptyset & \text{Otherwise} \end{cases}$$

The best representation of a set of tuples is captured by the corresponding abstraction function α :
$\alpha(r) =$

$$\begin{cases} \bot & \text{if } r = \emptyset \\ \mathscr{P}(X,r) & \textit{if } \forall t \in T(r), \forall a \in A : \textit{ either } t(a) = \mathscr{P}(X,r)(a) \textit{ or } \mathscr{P}(X,r)(a) = \top \\ \top & \textit{Otherwise} \end{cases}$$

Example 3. Table 4 shows the concrete relation instance r and associated abstract relation instance $r^{\#}$. Notice that, each tuple $t_p \in \mathcal{T}(r^{\#})$ is associated with a non-overlapping partition in r.

T	A	B	C	D	E	F	T	A	B	C	D	E	F
t_1	a_1	b_1	c_1	d_1	e_1	f_1	$\top$	a_1	b_1	c_1	d_1	$\top$	f_1
t_2	a_1	b_1	c_1	d_1	e_2	f_1							
t_3	a_2	b_1	c_2	d_2	e_2	f_1	$\top$	a_2	b_1	c_2	d_2	$\top$	f_1
t_4	a_2	b_1	c_2	d_2	e_3	f_1							
t_5	a_2	b_2	c_2	d_2	e_1	f_2	$\top$	a_2	b_2	$\top$	$\top$	e_1	f_2
t_6	a_2	b_2	c_2	d_1	e_1	f_2							
t_7	a_2	b_2	c_1	d_1	e_1	f_2							
t_8	a_2	b_2	c_1	d_2	e_1	f_2							
t_9	a_1	b_2	c_2	d_1	e_2	f_2	$\top$	a_1	b_2	c_2	d_1	$\top$	f_2
t_{10}	a_1	b_2	c_2	d_1	e_1	f_2							

Table 4. Concrete Relation r and the corresponding abstract relation $r^{\#}$

4.2 Watermark Generation

We are interested in a watermark generation process starting from a partition $[R_k]$ $1 \leq k \leq n]$, in a relational database table . The partitioning can be seen as a virtual grouping which does not change the physical position of the tuples as described in the last section. Let the owner of the relation R 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 $\mathcal{G}$ is seeded with the concatenation of watermark key $\Re$ and the primary key $r[P]$ for each tuple $r \in \mathcal{T}(R_k)$, generating a sequence of numbers, through which we select a field (attribute) in $\mathcal{A}(R)$. A fixed number of MSBs (most significant bits) and LSBs (least significant bits) of the selected field are used for generating the watermark of that corresponding field. The reason behind it is: a small alteration in that field in R will affects the LSBs first and a major alteration will affects the MSBs, so the LSB and MSB association is able to track the changes in the actual attribute values. So here we make the watermark value as the concatenation of m number of MSBs and n number of LSBs such that $m + n = 8$. Our aim is to make a *grey scale* image as the watermark of that associated partition, so the value of each cell must belongs to [0..255] range. Formally, the watermark (grey scale image) R_k^I corresponding to the k^{th} partition $[R_k]$ is generated in Table 5,

Example 4. 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 4 in the above algorithm to select random attributes. Here (10111111, 10110101, 10010101,11110111) is the generated watermark for the tuple (Bob, 10000, London, Manager), where we consider 4 MSBs concatenated with 4 LSBs. And the attribute watermark pair looks like $\{\langle Bob, 10010101\rangle, \langle 10000, 10111111\rangle, \langle London, 11110111\rangle, \langle Manager, 10110101\rangle\}$. As depicted in Figure 4.

$genW(R_k, \Re)$

for *each tuple* $t \in \mathcal{T}(R_k)$ do
construct a row $p \in \mathcal{T}(R_k^I)$
for $(i = 0; i < \|\mathcal{A}(R_k)\|; i = i + 1)$ do
$j = \mathcal{G}_i(\Re, r[P])\ mod\ \|\mathcal{A}(R_k)\|$
$p_i.R_k^I = (t[j]_{m\ MSBs} \mid t[j]_{n\ LSBS})_{10}\ mod\ 256$
delete the j^{th} attribute from t
endfor
endfor
return(R_k^I)

Table 5. Watermark generation

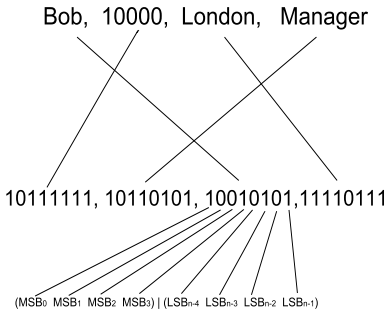

Fig. 4. Watermark generation for a single tuple

The whole process does not introduce any distortion to the original data. The use of MSB LSB combination is for thwarting potential attacks that modify the data as it simply produces an integrity certificate.

Abstraction Let us define the abstract framework behind this generation algorithm, let

- $\mathbb{R} = \{R : T \times A \to \mathbb{Z}\}$
- $\mathbb{R}^I = \{R^I : T \times A \to [0..255]\}$
- The abstraction function $\alpha : \mathbb{R} \to \mathbb{R}^I$ is defined as $\alpha(R)(t, a) = \overline{\alpha}(R(t, a))$ where $\overline{\alpha} : \mathbb{Z} \to [0..255]$.

4.3 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 R_k^I are needed to check a suspicious partition R'_k of the suspicious database relation R'. It is assumed that the primary key attribute has not been changed or else can be recovered. Table 6 states the watermark detection algorithm.

$genW(R_k, \Re)$

for *each tuple* $t \in \mathcal{T}(R_k)$ do
construct a row $p \in \mathcal{T}(R_k^I)$
for $(i = 0; i < \lvert\mathcal{A}(R_k)\rvert; i = i + 1)$ do
$j = \mathcal{G}_i(\Re, r[P])\ mod\ \lvert\mathcal{A}(R_k)\rvert$
if $p_i.R_k^I = (t[j]_{m\ MSBs} \mid t[j]_{n\ LSBS})_{10}\ mod\ 256$ then
matchC = matchC + 1
endif
delete the j^{th} attribute from t
endfor
endfor
if $matchC = \omega$ then
// ω = number of rows × number of columns in R_k^I
return true
else
return false
endif

Table 6. Watermark detection

The variable *matchC* counts the total number of correct matches. The authentication is checked by comparing the generated watermark bitwise. And after each match *matchC* is increased by 1. Finally, the total match is compared to the number of bits in the watermark image R_k^I associated with partition R_k to check the final authentication.

5 Zero Distortion Authentication Watermarking (ZAW)

So far, we have a set of grey scale images corresponding to a data table R. Each gray scale image R_k^I (k=1 to n) is associated with n partitions R_k (k=1 to

n) of R. And image R_k^I is said to be the abstraction of partition R_k. Now the authentication of database owner is necessary. We employ the zero-distortion authentication watermarking (ZAW) [19] to authenticate the table which introduces no artifact at all.

Without loss of generality we assume that the table R is fragmented into n independent grey scale images R_1^I; R_2^I; ...;R_n^I. Each image does not depend on any other images. If we consider R as the concrete table then R^I (composition of all image fragments R_1^I; R_2^I; ...;R_n^I) can be considered as its abstract counterpart. An equivalent image can be derived from π_k using Myrvold and Ruskey's linear permutation ranking algorithm [21] by permuting the partitions in R^I. The algorithm *unrank* makes a permutation of the segments based on a secret number (M) only known to the database owner and this number can be considered as a private key of the owner. The owner can distribute the number of partitions n as public key.

$unrank(n, M, \pi)$	$rank(n, \pi, \pi^{-1})$
for $(i = 0; i < n; i++)$ do	if $(n = 1)$ then
$\pi_i = i$	return 0
endfor	endif
if$(m > 0)$ then	$s = \pi[n-1]$
$swap(\pi[n-1], \pi[M \ mod \ n])$	$swap(\pi[n-1], \pi[\pi^{-1}[n-1]])$
$unrank(n-1, \lfloor M/n \rfloor, \pi)$	$swap(\pi^{-1}[s], \pi^{-1}[n-1])$
endif	return (s+m * rank$(n-1, \pi, \pi^{-1})$)

Table 7. Encryption and Decryption algorithms

In Table 7 the *unrank* algorithm can be treated as the encryption algorithm based on the private key M, whereas the algorithm *rank* can be treated as the decryption algorithm based on the public key n.

6 Robustness

We analyze the robustness of our scheme by Bernoulli trials and binomial probability. 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*.

6.1 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 $\langle MSB_m | LSB_n \rangle$ association (grey scale entry) has the probability $\frac{1}{2^8}$ to match to the corresponding entry in R^I. Assume that for a non-watermarked data partition R'_q and watermarking data partition R_k, $|\mathcal{A}(R'_q)| = |\mathcal{A}(R_k)|$, $|\mathcal{T}(R'_q)| = |\mathcal{T}(R_k)|$ and $P.R'_q = P.R_k$, i.e. both have same number of tuples, attributes and same primary keys, respectively. Let $\omega = |\mathcal{A}(R_k)| * (m + n) * |\mathcal{T}(R_k)|$ is the size of the watermark. 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

$$F_h = B(\omega, \lfloor \frac{\omega}{\mathfrak{T}} \rfloor, \frac{1}{2^8})$$

6.2 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.

Subset Deletion Attack For tuple deletion and attribute deletion, the $\langle MSB_m | LSB_n \rangle$ association in the deleted tuples or attributes will not be detected in watermark detection; however, the other tuples or attributes will not be affected. Therefore, all detected bit strings will match their counterparts in the watermark, and the false miss is zero.

Subset Addition Attack 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 $\frac{1}{\mathfrak{T}}$ bit strings of those newly added tuples (which consists of $v\varsigma$ $\langle MSB_m | LSB_n \rangle$) must not match their counterparts

in the watermark (which consists of ω bits). 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

$$F_m = B(\upsilon\varsigma, \lfloor \frac{|\mathcal{A}(R_k)| * \varsigma}{\mathfrak{T}} \rfloor, \frac{1}{2^8})$$

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

7 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 ensures both authentication and integrity.
- As 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.
- This watermarking process has an advantage over hash function, as it does not depend on the ordering of the tuples.

Acknowledgements

Work partially supported by RAS L.R. 7/2007 Project TESLA.

References

1. Hartung, F. and Girod, B.: Watermarking of uncompressed and compressed video Signal Processing. **66** (1998) 238–301
2. Langelaar, G. and Setyawan, I. and Lagendijk, R.: Watermarking digital image and video data: A state-of-art overview IEEE Signal Processing. **17** (2000) 20–46
3. Potdar, V. and Han, S. and Chang, E.: A survey of digital image watermarking techniques. In the Proceeding of the 3rd International IEEE Conference on Industrial Informatics (2005). 709–716.
4. Arnold, M. and Schumucker, M. and Wolthusen, S.: Techniques and Applications of Digital Watermarking and Content Protection. Artech House. ISBN: 10: 1580531113 (2003)

5. Bassia, P. and Pitas, L. and Nikolaidis, N.: Robust audio watermarking in the time-domain IEEE Trans. Multimedia. DOI: 10.1109/6046.923822 (2001)232–242
6. Lemma, A. and Aprea, J. and Kherkhof, L.: A temporal-domain audio watermarking technique. IEEE Trans. Signal Process. DOI: 10.1109/TSP.2003.809372 (2003)
7. Ruanaidh and Dowling and Boland: Watermarking digital images for copyright protection IEEE ProcVision Signal, Image Procesing. **143** (1996) 250–256
8. Agrawal, R. and Haas, P.J. and Kiernan, J.: Watermarking relational data: framework, algorithms and analysis. The VLDB Journal. **12** (2003) 157–169
9. Bhattacharya, S. and Cortesi, A.: A Generic Distortion Free Watermarking Technique for Relational Databases. In the Proceeding of fifth International Conference on Information Systems Security. LNCS **5905**(2009) 252–264
10. Cousot, P. and Cousot, R.: Systematic design of program analysis frameworks. In Proceedings of the 6th ACM Symp. on Principles of Programming Languages (1979)269–282,
11. Cousot, P. and Cousot, R. : Abstract interpretation frameworks. Logic and Comp (1992) 511–547
12. Cousot, P. : Abstract interpretation based formal methods and future challenges. Logic and Comp. LNCS 2000 (2000) 138–156 Keywords = Wilhelm, R., ed.: Informatics 10 Years Back, 10 Years Ahead, LNCS, Springer-Verlag ,
13. Ng, W. and Lau, H. L.: Effective approaches for watermarking xml data. DASFAA (2005) 68–8
14. Gross-Amblard, D.: Query-preserving watermarking of relational databases and xml documents. In Proceedings of the Nineteenth ACM SIGMOD-SIGACTSIGART Symposium on Principles of Database Systems (2003) 191-201
15. Li, Y. and Guo, H. and Jajodia, S.: Tamper detection and localization for categorical data using fragile watermarks. Digital Rights Management Workshop (2004) 73-82
16. Guo, H. and Li, Y. and Liu, A. and Jajodia, S.: A fragile watermarking scheme for detecting malicious modifications of database relations. 176 (2006)1350–1378
17. Bhattacharya, S. and Cortesi, A.: Database authentication by distortion free watermarking (Best student paper award). In the Proceeding of fifth International Conference on Software and Data technology, ICSOFT 2010 Athens, Greece, (2010) 219–226
18. The Keyed-Hash Message Authentication Code HMAC. FEDERAL INFORMATION PROCESS STANDARDS PUBLICATION (2002)
19. Yongdong, W.: Zero-Distortion Authentication Watermarking ISC (2003) 325–337
20. De Bra, P. and Paredaens, J.: An algorithm for horizontal decompositions. Inf. Process. Lett. 17(1983) 91–95
21. Myrvold, W. and Ruskey, F.: Ranking and unranking permutations in linear time. Information Processing Letters (2000)
22. , Bhattacharya, S. and Cortesi, A.: A distortion free watermarking framework for relational databases. In the Proceeding of forth International Conference on Software and Data technology ICSOFT 2009, Sofia, Bulgaria (2009)229–234
23. Halder, R. and Cortesi, A.: A persistent public watermarking of relational databases. In Proceedings of the 6th international conference on Information systems security. LNCS 6503 (2010) 216–230
24. Halder, R. and Cortesi, A.: Watermarking Techniques for Relational Databases: Survey, Classification and Comparison. Journal of Universal Computer Science, 16(21) (2010) 3164–3190, December 2010.
25. Sion, R.: Proving ownership over categorical data. In Proceedings of IEEE International Conference on Data Engineering, (2004) 584-596