DATABASE AUTHENTICATION BY DISTORTION FREE WATERMARKING

Sukriti Bhattacharya and Agostino Cortesi
Dipartimento di Informatica, Universita Ca' Foscari di Venezia, Via Torino 155, 30170 Venezia, Italy
sukriti@dsi.unive.it, cortesi@unive.it

Keywords: Database watermarking, ZAW, Public key watermark, Abstract interpretation.

Abstract: In this paper 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. 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 gray scale image of the partition as a watermark of that partition, that serves as tamper detection procedure, followed by employing a public zero distortion authentication mechanism to verify the ownership.

1 INTRODUCTION

Because people pay much attention to data mining, more and more research institutions begin to buy the databases to analyze.The enterprises would also like to sell their data warehouses for the institutions to do their research if the data does not concern customers personal data. The market of databases is flourishing because this kind of demand and supply market is developed. But the database is easier to be copy and abuse, and the internet is popular so that the information propagates more rapidly. The information passes through the internet without monitor and could be destroyed or altered. The consumer of the information would have no idea about the validity of the information received.

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 (Agrawal et al., 2003). Watermarking techniques apply to various types of host content. Here, we concentrate on relational databases. Database watermarking consists of two basic processes: watermark insertion and watermark detection (Agrawal et al., 2003), 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.

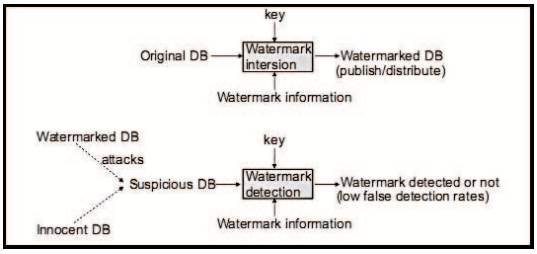

Figure 1: Basic watermarking process.

Watermarking has been extensively studied in the context of multimedia data for the purpose of ownership protection and authentication (Cox et al., 2001) (Johnson et al., 2000). 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 (Agrawal et al., 2003) 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 (Sion et al., 2004) 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 (Sion, 2004) 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 (Gross-Amblard, 2003)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 (Agrawal et al., 2003)(Gross-Amblard, 2003)(Sion et al., 2004)(Sion, 2004), 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. To meet this requirement, we further strengthen this approach and propose a distortion free watermarking algorithm for relational databases and discuss it in abstract interpretation framework proposed by Patrick Cousot and Radhia Cousot (Cousot and Cousot, 1977) (Cousot and Cousot, 1992) (Cousot, 2001) (Cousot and Cousot, 2004) (Cousot and Cousot, 2007).

In (Bhattacharya and Cortesi, 2009a) we presented a 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.Then in (Bhattacharya and Cortesi, 2009b) we faced the same issue by a more sophisticated and completely orthogonal approach that allows us by removing the constraints on the presence of categorical values in the table and by considering any partitioning generate a binary image that serves the purpose of temper detection of that associated partition. Here, we go one step further. Namely 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. Instead of binary image, we generate a gray scale image to strengthen the verification of integrity and we employ a zero distortion public authentication mechanism (Wu, 2003) for ownership proof. We prove it as 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 gray scale 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 the overall architecture 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.

Observe that our proposal improves both with respect to our previous works on distortion free watermarking and with respect to the application of hash functions to the whole database: in fact the authentication certificate we produce as a watermark does not depend on the order of the tuples belonging to the same partition set. This makes our approach scalable to large databases while a simple hash function approach obviously does not scale well.

The paper is organized as follows. In section 2, we formalize the definition of tables in relational database and the watermarking process. Section 3 illustrates how distortions and watermarking are related. In section 4, we present the data partitioning algorithm and explain the partitioning in the abstract interpretation framework. The watermark generation algorithm for a data partition is illustrated in section 5. In section 6, we propose the watermark detection algorithm. In section 7, we introduce a zero distortion public authentication mechanism. The robustness of the proposed technique is discussed in section 8. Finally we draw our conclusions in section 9.

2 PRELIMINARIES

This section contains an overview of Galois connection (Cousot and Cousot, 1977) (Cousot and Cousot, 1992) (Cousot and Cousot, 2007) and some formal definitions (Haan and Koppelaars, 2007) and (Collberg and Thomborson, 2002) of tables in relational database and database watermarking.

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 l 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 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 $\bot = \sqcap L = \sqcup\emptyset$

For instance (L, $\sqsubseteq$, $\sqcup$, $\sqcap$, $\top$, $\bot$) where L = {1, 2, 3, 4, 6, 9, 36}. $\sqsubseteq$ = |, $\bot$= 1 and $\top$ = 36 is a complete lattice. It can be represented using Hasse diagram as shown below

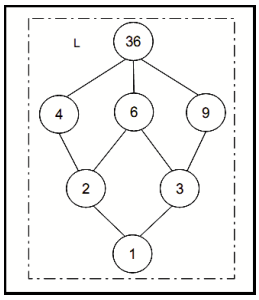

Figure 2: Complete lattice.

Definition 2.3 (Galois Connection). Let C (concrete) and A (abstract) be two domains (or lattices). Let α : C$\rightarrow$ A and γ : A$\rightarrow$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).
- $\forall$ a $\in$ A : $\alpha(\gamma(a)) \sqsubseteq a$
- $\forall$ c $\in$ 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 $\Leftrightarrow$ 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. $\forall$ t $\in$ 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 (Sion et al., 2004)(Sion, 2004)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 (HMAC, 2002).

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$(\mathbf{D}, \Re, \mathbf{m})$.

1: **for** each tuple $r \in D$ **do**
2: partition $\leftarrow$ HMAC($\Re$ | r.P) mod m
3: insert r into $S_{partition}$
4: **end for**
5: **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.

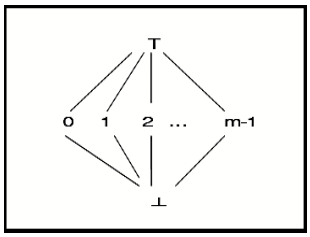

Figure 3: Lattice of the abstract domain.

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 $\subseteq$ D, and given a set of natural number i $\in \mathbb{N}$, we can define a concretization map γ as follows:
$\gamma(\top) = D$
$\gamma(\perp) = \emptyset$

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

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

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

The two functions α and γ described above yield a Galois connection (Cousot and Cousot, 1977) (Cousot and Cousot, 1992) 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 (Schneier., 1996) 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, through which we select a field (attribute) in D. 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 D 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 MSBs and n LSBs such that $m+n=8$. Our aim is to make a gray scale image as the watermark of that associated partition, so the value of each cell must belongs to [0 to 255] range.Formally, the watermark W_k corresponding to the k^{th} partition $[S_k]$ is generated as follows,

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

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: $t.W_k^i = (m\,MSBs \mid n\,LSBs)_{10} \, mod \, 256$ 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 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 $\{\langle Bob, 10010101\rangle, \langle 10000, 10111111\rangle, \langle London, 11110111\rangle, \langle Manager, 10110101\rangle\}$. The entire concept is illustrated by the following diagram (Figure 4)

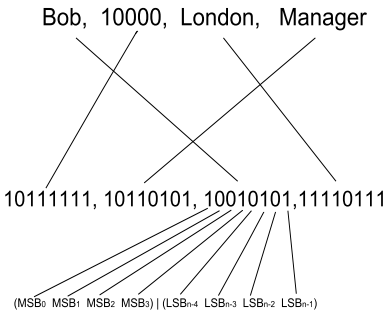

Figure 4: Watermark generation for a single tuple.

So if there are n number of tuples in the partition $[S_k]$, *genW* generates a gray scale 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 MSB LSB combination is for thwarting potential attacks that modify the data as it simply produces an integrity certificate.

5.1 Functional Abstraction

Theorem 1 (Galois Connection). Given a table D $\subseteq$ $C_0 \times C_1 \times C_2 \times ... \, C_{v-1}$, let $\mathbb{B}^v$ is the set of all binary sequences of length v. We can define abstraction and concretization function between $\wp(C_0 \times C_1 \times C_2 \times ... \, C_{v-1})$ and $\wp(\mathbb{B}^v)$ as follows
$\alpha(S)$ = $\{genW(S, \Re)(r) \mid r \in S\}$
$\gamma(W)$ = $\{t \in S \subseteq D \mid genW(S, \Re)(t) \in W\}$. Then α and γ form a Galois connection (Cousot and Cousot, 1977) (Cousot and Cousot, 1992).

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

The data set (table) $D \subseteq \wp(C_0 \times C_1 \times ...C_{v-1})$ and the watermark $W \subseteq \wp(\mathbb{B}^v)$. By Theorem 1 (D, W, α, γ) form a Galois Connection. The function $genW : D \to W$ is the watermark generation function described above. $\forall t \in D$, $f_{alt} : D \to D$ and $\forall t^{\#} \in W$, $f^{\#}_{alt} : W \to W$ 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:

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

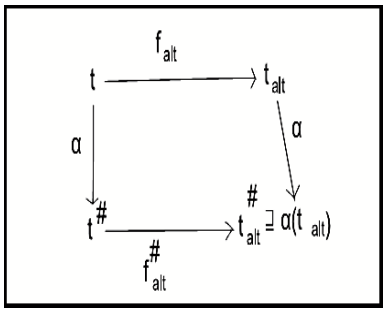

Figure 5: Soundness.

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

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** $t.W^i_k = (m\,MSBs \mid n\,LSBs)_{10}\, mod$ 256 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=ω **then**
13: // ω = 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 $W^{t,i}_k$, 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 ZERO DISTORTION AUTHENTICATION WATERMARKING (ZAW)

So far, we have a set of gray scale images corresponding to a data table D. Each gray scale image W_k (k=1 to m) is associated with m partitions S_k (k=1 to m) of D. And image W_k is said to be the abstraction of partition S_k. Now the authentication of database owner is necessary. We employ the zero-distortion authentication watermarking (ZAW) (Wu, 2003) to authenticate the table which introduces no artifact at all. Figure 6 describes the framework of the ZAW scheme which does not modify the host content but transforms the host into its equivalence.

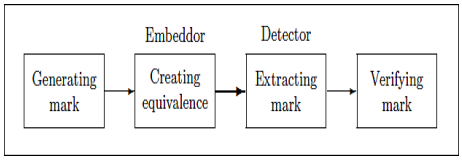

Figure 6: ZAW framework.

Without loss of generality we assume that the table D is fragmented into m independent gray scale

images W_0; W_1; ...;W_{m-1}.Each image does not depend on any other images. If we consider D as the concrete table then W (composition of all image fragments W_0; W_1; ...;W_{m-1}) can be considered as its abstract counterpart (Theorem 1). An equivalent image W' can be derived from π_k using Myrvold and Ruskey's linear permutation ranking algorithm (Myrvold and Ruskey, 2000). 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 m (section 4) as public key. *unrank* can be treated as a encryption algorithm based on the private key M.

Algorithm 4: Unrank$(\mathbf{m},\mathbf{M},\pi)$.

1: **for** (i=0; i<m ; i++) **do**
2: π_i = i
3: **end for**
4: **if** m $>$ 0 **then**
5: **swap**($\pi[m-1]$, π[M mod m])
6: **Unrank**($m-1$, $\lfloor M/m \rfloor$, π)
7: **end if**
End

The algorithm *rank* can be treated as decryption algorithm based on the public key m.

Algorithm 5: rank$(\mathbf{m},\pi,\pi^{-1})$.

1: **if** m = 1 **then**
2: **return** 0
3: **end if**
4: s = $\pi[m-1]$
5: **swap**($\pi[m-1],\pi[\pi^{-1}[m-1]]$)
6: **swap**($\pi^{-1}[s],\pi^{-1}[m-1]$)
7: **return**(s+m * rank($m-1$, π, π^{-1}))

8 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

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

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

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*.

8.1 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 $\langle MSB_m|LSB_n\rangle$ (m+n=8) association in data has the probability $\frac{1}{2^8}$ to match to 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 $\omega = v(m+n)q$ is the size of the watermark. where v is the number of attributes being watermarked and q is the number of tuples in partition S_q. The false hit is the probability that at least $\frac{1}{\tau}$ portion of ω can be detected from the non-watermarked data by sheer chance. When τ is the watermark detection parameter. It is used as a tradeoff between false hit and false miss. Increasing τ will make the robustness better in terms of false hit. Therefore, the false hit F_h can be written as

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

8.2 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.

8.2.1 Subset Deletion Attack

For tuple deletion and attribute deletion, the $\langle MSB_m|LSB_n\rangle$(m+n=8) 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.

8.2.2 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 $\omega = v(m+n)q$ 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

$$F_m = B(v\varsigma, \lfloor \frac{v\varsigma}{\mathfrak{T}} \rfloor, \frac{1}{2^8})$$

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

9 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;
- 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" and by RAS project TESLA.

REFERENCES

Agrawal, R., Haas, P. J., and Kiernan, J. (2003). Watermarking relational data: framework, algorithms and analysis. *The VLDB Journal*, 12(2):157–169.

Bhattacharya, S. and Cortesi, A. (2009a). A distortion free watermark framework for relational databases. In *ICSOFT (2)*, pages 229–234.

Bhattacharya, S. and Cortesi, A. (2009b). A generic distortion free watermarking technique for relational databases. In *ICISS*, pages 252–264.

Collberg, C. S. and Thomborson, C. (2002). Watermarking, tamper-proofing, and obfuscation - tools for software protection. *Software Engineering, IEEE Transactions on*, 28:735–746.

Cousot, P. (2001). Abstract interpretation based formal methods and future challenges. In *Informatics - 10 Years Back. 10 Years Ahead.*, pages 138–156. Springer-Verlag.

Cousot, P. and Cousot, R. (1977). 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.

Cousot, P. and Cousot, R. (1992). Abstract interpretation frameworks. *Journal of Logic and Computation*, 2:511–547.

Cousot, P. and Cousot, R. (2004). An abstract interpretation-based framework for software watermarking. In *POPL '04: Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 173–185. ACM.

Cousot, P. and Cousot, R. (2007). Abstract interpretation and application to static analysis. In *1st IEEE and IFIP International Symposium on Theoretical Aspects of Software Engineering*.

Cox, I. J., Miller, M. L., and Bloom, J. A. (2001). *Digital Watermarking: Principles and Practice.* Morgan Kaufmann Publishers.

Gross-Amblard, D. (2003). Query-preserving watermarking of relational databases and xml documents. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–201. ACM.

Haan, L. D. and Koppelaars, T. (2007). *Applied Mathematics for database Professionals.* Apress.

HMAC (2002). The keyed-hash message authentication code. Federal Information Process Standards Publication.

Johnson, Duric, and Jajodia (2000). *Information Hiding: Steganography and WatermarkingAttacks and Countermeasures.* Kluwer Publishers.

Myrvold, W. and Ruskey, F. (2000). Ranking and unranking permutations in linear time. *Information Processing Letters*, 79:281–284.

Schneier., B. (1996). *Applied Cryptography.* John Wiley and Sons.

Sion, R. (2004). Proving ownership over categorical data. In *In Proceedings of the IEEE International Conference on Data Engineering ICDE*, pages 584–596.

Sion, R., Atallah, M., and Prabhakar, S. (2004). Rights protection for relational data. In *In Proceedings of ACM SIGMOD*, pages 98–109. ACM Press.

Wu, Y. (2003). Zero-distortion authentication watermarking. In *Proc. International Conference on Information Security, LNCS 2851*, pages 325–337.