Watermarking for Relational Database by using Threshold Generator

Dr. Yossra H. Ali' & Bashar Saadoon Mahdi *

Received on: 7/10/2009
Accepted on:2/12/2010

Abstract

Providing ownership on relational database is a crucial issue in today internet-based application environments and in many content distribution applications.This paper provides the effective watermarking technique to protect valuable numeric relational data from illegal duplications and redistributions as well as to claim ownership,the robustness of proposed system depending on using new hybrid techniques ,first technique MAC(Message Authentication Code) that used one way hash function SHA1 ,second technique is threshold generator base on simple combination of odd number of register and by using secret key in proposed system. Detecting the watermark neither requires access to neither the original data nor the watermark. The watermark can be detected even in a small subset of a watermarked relation as long as the sample contains some of the marks. The finally stage is the analysis of technique that used, our extensive analysis shows that the proposed technique is robust against various forms of malicious attacks and updates to the data.

Keywords: watermark, relational database, hash function, threshold generator.

العلامة المائية في قاعدة البيانات بواسطة استخدام مولد حد العتبة

الخلاصة

ان اثبات حقوق الملكية لقواعد البيانات العلائقية هي قضية مهمة في بيئات التطبيقات المستندة على الانترنيت وفي العديد من تطبيقات توزيع المحتويات. يوفر هذا البحث تقنية العلامة المائية بشكل فعال لحماية البيانات العددية العلائقية المهمة و الثمينة من النسخ واعادة التوزيع الغير شرعية لهذه البيانات بالاضافة الى ذلك لادعاء الملكية الخاصة ،حيث تعتمد قوة النظام المقترح على استخدام تقنيات هجينة , اولا تقنية MAC (رمز تخويل الرسالة) الذي يستخدم دوال ذات اتجاه واحد غير قابلة للكسر او الارجاع وهي خوارزمية SHA1 والتقنية الثانية هي استخدام مولد حد العتبة من تجميع عدد فردي من المسجلات الخطية وكذلك باستخدام المفاتيح السرية في النظام المقترح.

اكتشاف العلامة المائية لايتطلب الوصول الى البيانات الاصلية ولا الى العلامة المائية . العلامة المائية يمكن ان تكتشف من مجموعة صغيرة لقاعدة البيانات المعلمة لطالما العينة تحتوي على البعض من اجزاء العلامة المائية.

المرحلة الاخيرة هي تحليل التقنية المستخدمة ونتائج التحليل وضحت ان هذه التقنية تقاوم الاشكال المختلفة من الهجمات الخبيثة وكذلك ضد تحديث البيانات.

***Computer Sciences Department, University of Technology/ Baghdad**

https://doi.org/10.30684/etj.29.1.3
2412-0758/University of Technology-Iraq, Baghdad, Iraq

1. Introduction

The management of huge database becomes necessary for most information system such as banking, stock-control, payroll, personal records, and internet…etc. the reason for database is that those data represent of variety of life application. The relational data model represents data in the form of tables. The relational model based on mathematical theory and therefore has a solid theoretical foundation [1].

Watermarking works by deploying resilient information hiding techniques to insert an indelible mark in the data such that (i) the insertion of the mark does not destroy the value of the data (i.e., the data is still useful for the intended purpose); and (ii) it is difficult for an adversary to remove or alter the mark beyond detection without destroying the value of the data. Clearly, the notion of value or utility of the data is central to the watermarking process. This value is closely related to the type of data and its intended use. For example, in the case of software the value may be in ensuring equivalent computation, and for text it may be in conveying the same meaning (i.e., synonym substitution is acceptable). Similarly, for a collection of numbers, the utility of the data may lie in the actual values, in the relative values of the numbers, or in the distribution (e.g., normal with a certain mean) [2].

An important point about watermarking should be noted. By its very nature, a watermark modifies the item being watermarked. If the data to be watermarked cannot be modified then a watermark cannot be inserted. The critical issue is not to avoid changing the data, but to limit the change to acceptable levels with respect to the intended use of the data [2].

There is a need for watermarking database relations to deter their piracy, identify the unique characteristics of relational data which pose new challenges for watermarking, and provide desirable properties of a watermarking system for relational data. A watermark can be applied to any database relation having attributes which are such that changes in a few of their values do not affect the applications [3].

2. Watermarking Principles

Although steganography and watermarking both describe techniques used for covert communication, steganography typically relates only to covert point to point communication between two parties [4]. Steganographic methods are not robust against attacks or modification of data that might occur during transmission, storage or format conversion [5].

Watermarking, as opposed to steganography, has an additional requirement of robustness against possible attacks. An ideal steganographic system would embed a large amount of information perfectly securely, with no visible degradation to the cover object.

An ideal watermarking system, however, would embed an amount of information that could not be removed or altered without making the cover object entirely unusable.

As a side effect of these different requirements, a watermarking system will often trade capacity and

*Computer Sciences Department, University of Technology/ Baghdad

perhaps even some security for additional robustness [6].

The working principle of the watermarking techniques is similar to the steganography methods. A watermarking system is made up of a watermark embedding system and a watermark recovery system. The system also has a key which could be either a public or a secret key. The key is used to enforce security, which is prevention of unauthorized parties from manipulating or recovering the watermark. The embedding and recovery processes of watermarking are shown in Figure 1 and 2 .

For the embedding process the inputs are the watermark, cover object and the secret or the public key. The watermark used can be text, numbers or an image. The resulting final data received is the watermarked data $\boldsymbol{W}$ Figure 2 Digital watermarking – Decoding. The inputs during the decoding process are the watermark or the original data, the watermarked data and the secret or the public key. The output is the recovered watermark **W** [7].

3. The Proposed Watermarking Relational Database

The watermarking of relational data has significant technical challenges and practical applications to deserve serious attention from the database research community. A desiderata for a system for watermarking needs to be specified, followed by development of specific techniques. These techniques will most certainly use existing watermarking principles. However, they will also require enhancements to the current techniques as well as new innovations.

This system attempted to provide such a desiderata, to demonstrate the feasibility of watermarking relational data, it is presented an effective technique that satisfies these desiderata. This technique marks only numeric attributes and assumes that the marked attributes can tolerate changes in some of the values.

The basic idea of first embedding scheme (watermarking) is to ensure that some bit positions for some of the attributes of some of the tuples contain specific values. The tuples, attributes within a tuple, bit positions in an attribute, and specific bit values are all algorithmically determined under the control of a private key known only to the owner of the relation. This bit pattern constitutes the watermark. Only if one has access to the private key, can the watermark symbols be detected with high probability. The analysis shows that the watermark can withstand a wide variety of malicious attacks.

This technique marks only numeric attributes and assumes that the marked attributes are such that small changes in some of their values are acceptable and non obvious .All of the numeric attributes of a relation need not be marked. The data owner is responsible for deciding which attributes are suitable for marking.

We are watermarking a database relation whose scheme is R(PK,A0,A1,…,Av-1) where PK is the primary key attribute. For simplicity, assume that all v attributes A0,A1,..,Av-1 are candidates for marking .They are numeric attributes and their value are such changes in x bit position for all of them are imperceptible . y

is a control parameter that determines the number of tuples marked .table(1) below show the symbols that used in this paper.

3.1 Message Authenticated Code (MAC)[8]

A one-way hash function H operates on an input message M of arbitrary length and returns a fixed length hash value h= H (M). It has the additional characteristics that (1) given M it is easy to compute h, (2) given h, it is hard to compute M such that H (M)=h , and (3) given M , it is hard to find another message M` such that H (M)= H (M`) . SHA are a good choices for H .

A message authenticated code (MAC) is a one-way hash function that depends on a key. There are two functions well which are used in proposed system depend on SHA1 function [9].

- Hash1(r.p, Ķ) = H(r.p& H(r.p& Ķ& H(Ķ)))
- Hash2(V, Ķ) = H(Ķ& H(H(Ķ) ¤ H(V))

Where r.p is a primary key of attributes in relational, Ķ is a secret key known only to the owner, & represents concatenation, ¤ represents XOR operation, V is the variable. All function in above returns binary number 160 digit.

3.2 Threshold Generator (Thresholdstream)[10]

This paper uses this technique to determine the value 0 or 1, which is inserted to mark bit in marked attributes of marked tuple in relational database. Threshold employs a variable (odd) number of LFSRs .Threshold is a nonlinear generator, lets assume that we use three LFSRs,then the output generator can be written as :

b= (a1^a2) XOR (a1^a3) XOR (a2^a3)

Where a is input (one bit) and the symbol ^ is AND operation

The linear complexity:

n1n2+n1n3+n2n3 where n is the length of LFSR

Threshold generator in this work consists of three LFSR's connected as shown in Figure (3). The concept is if more than half the output bits are 1, then the output of the generator is 1, if more then half the output bit are 0 then the output of the generator is 0.

3.3 The watermarking Insertion Algorithm

Algorithm 1 Watermark Concealing Algorithm

Input:

// only the owner knows the secret key Ķ.

// R is the relation to be marked.

// the parameter **У, ν**, and χ are also private to owner.

Output: New value of r.A (relation of attribute)

Begin

1- for each tuple r € R do

2 - Tuple marked = hash1 (r.p, Ķ) mod **У** // marked tuple

3 - if (hash1 (r.p, Ķ) mod **У** equals 0) then

4 - Attribute marked i = hash1 (r.p , Ķ) mod **ν** // marked attribute

5 - bit marked j = hash1 (r.p , Ķ) mod χ // bit position

6- $r.A_i$ =concealing ($r.A_i$, tuple marked, attribute marked, bit marked j, Ķ)

// watermark generator and embedding in database

6.1- concealing (number **τ** , tuple marked, attribute marked bit marked j, Ķ) **return** number

6.2 - T1= **hash2**(tuple marked , Ķ) // SHA1 hash function

6.3- T2= **hash2**(attribute marked, Ķ)

6.4 - T3= **hash2**(bit marked j, Ķ)

6.5 - T=**Thresholdstream** (T1,T2,T3) // watermark generator by threshold generator of steam cipher

6.6- Set the LSB position of **τ** to 1^{th} bit position of T

7 - until ending of tuples in database

End

This section presents briefly the watermarking relational database steps in the algorithm 1 .Line 2 compute the index of tuple marked by using mod operation to the hash1 value on **Ў** .Line 3 determines if the tuple under consideration will be marked. Because of the use of MAC (hash1) only the owner who has the knowledge of the private key Ķ can easily determine which tuples have been marked. line 4 determines the attribute that will be marked amongst the **ν** candidate attributes. For a selected attribute, line 5 determines the bit position amongst χ least significant bits that will be marked; the results of the tests in lines 4 and 5 depend on the private key of the owner. For erasing a watermark, therefore, the attacker will have to guess not only the tuples, but also the marked attribute within a tuple as well as the bit position.

Line 6 concealing subroutine sets the selected bit to 0 or 1 depending on the result of sets operation that explain in line from 6.1 to 6.6.

The inputs of subroutine are the value of marked attributes and index of tuple marked and index of attribute marked and index of bit marked finally the secret key.

In line 6.2 the T1 is the hash value 160 digit computed from hash2 function for inputs(tuple marked and secret key) , In line 6.3 the T2 is the hash value 160 digit computed from hash2 function for inputs(attribute marked and secret key) , In line 6.4 the T3 is the hash value 160 digit computed from hash2 function for inputs(bit marked and secret key) ,line 6.5 T is the hash value 160 digit generating by **thresholdstream** function of steam cipher , finally line 6.6 Set the LSB position of **τ** (attribute value) to 1^{th} bit position of T .

3.4 watermark Detection Algorithm

Algorithm 2 Watermark Detecting Algorithm

Input:

// the parameter **Ķ, Ў, ν, χ** ,and **lencount** have the same used for watermark insertion.

// totalcount = matchcount = 0

Output: detect WM **Begin**

1- For each tuple s € S do

2- tuple marked = **hash1**(s.p, Ķ) mod **Ў**

3- if (**hash1**(s.p, Ķ) mod **Ў** equals 0) then

4- attribute marked i = **hash1**(s.p , Ķ) mod **ν**

5- bit marked j = **hash1** (s.p , Ķ) mod χ

6- totalcount = totalcount + 1

7- matchcount = matchcount + wm detection ($s.A_i$, tuple marked, attribute marked , bit marked j, Ķ)

7.1- WM detection (number **τ** , tuple marked, attribute marked,bit marked j, Ķ) **return** number

7.2- T1= **hash2**(tuple marked , Ķ)

7.3- T2= **hash2**(attribute marked, Ķ)

7.4- T3= **hash2**(bit marked j, Ķ)

7.5- T=**Thresholdstream** (T1,T2,T3)

7.6- if (LSB bit of τ equal 1^{th} bit position of T) then **return 1** else **return 0**

8- goto 1

9- Ґ=threshold (totalcount , α)

10 - if (matchcount $\geq$ Ґ) then ***suspect piracy*** (detect watermark)

This section presents briefly the steps to extract watermark from suspected relational database (S), all steps from 1 to 3 are explained in insertion algorithm.

Lines 4 and 5 determine the attribute and the bit position that must have been marked. The subroutine **wm detection** compares the current bit value with the value that must have been set for that bit by the watermark concealing algorithm. Thus know how many tuples are tested (totalcount) and how many of them contain the expected bit value (matchcount).

Line 6 increases the totalcount that determined how many tuples marked in insertion algorithm, line 7 increased matchcount when **wm detection** subroutine returns 1. matchcount determined how many tuples match with marked tuples in insertion algorithm.

In **wm detection** subroutine the inputs of subroutine are the value of marked attributes and index of tuple marked , index of attribute marked , index of bit marked and finally the secret key .

In line 7.2 T1 is the hash value 160 digit computed from hash2 function for inputs(tuple marked and secret key) , In line 7.3 T2 is the hash value 160 digit computed from hash2 function for inputs(attribute marked and secret key) , In line 7.4 T3 is the hash value 160 digit computed from hash2 function for inputs(bit marked and secret key) ,line 7.5 T is the hash value 160 digit generating by **thresholdstream** function of steam cipher ,that same steps in concealing algorithm.

The line 7.6 if LSB bit of τ equal 1^{th} bit position of T then **return 1** else **return 0**

Line 8 goto to line 1 to take each tuple in relation database and processing it from line 2 to line 7, until end of relational database, line 10 the matchcount is compared with the minimum count returned by the threshold function in line 9 for the test to succeed at the chosen level of significance α.

3.5 Implementation

We ran experiments in Windows2003 with 2.0 GHz CPU and 512 MB RAM. Algorithms are applied to Forest Cover Type dataset,available from University of California at Irvine KDD Archive (http://kdd.ics.uci.edu/databases/covertype/covertype.html). We choose the first 5,000 tuples to form a smaller relation in our experiment. We add a sequence number attribute served as the primary key. The first integer attribute ranging from 1859 to 3858 is predefined to be marked. The candidate bit positions to be marked are the first 3 bits right before the radix point. We used the string of "covertype" (dataset's name) as the secret key and an 8-bit string of "01001101" ("M" in ASCII code) as the fingerprint to be embedded. We chose $\gamma1 = \gamma2 = 20$; $\alpha1 = \alpha2 = \alpha3 = 0.01$.

The tuples selected to be marked by the first embedding process is 265 and 250 marked tuples by the second embedding process. So the total mark ratio is approximately 1/10. We can see

in table 2 that the mean and variance of the marked attribute hardly changed after inserting, so the

alteration for watermarking is small enough to maintain the usability of the data.

The most popular and special attacks to relational databases are the subset selection attack, subset addition attack and subset alteration attack. These attacks are corresponding to the most

frequent operations on relational databases: select, insert, delete and modify **.**

3.6 Analysis of Proposed System

This section analyzes the properties of the proposed watermarking technique.

3.6.1 Probabilistic Framework

In the line 9 of algorithm2 we used threshold subroutine the input of threshold function is totalcount that assume W that refer to marked tuples in database .in detection algorithm the owner looks at W bit and observes the number of bit value match those assigned by the concealing subroutine.

The probability is that at least **Γ́(minimum marked tuple)** out of **W** random bits ,each bit is equal to 0 or 1 with equal probability. Therefore, the subroutine **subroutine** threshold(W, α) **return** minimum Γ́.

α is the probability that owner will discover her watermark in a database relation not marked by her. By choosing lower values of **α**, owner can increase her confidence that if the detection algorithm finds owners watermark in a suspected relation, it probably is a pirated copy.

3.6.2 Detectability

The watermark detectability depends on two important values:

1- the significance level **α**

2- the number of marked tuples **W**.

The latter in turn depends on the number of tuples in the relation **n** and the gap parameter **У**.

Watermark Detection Figure (4) plots the proportion of marked tuples that must have the correct watermark value for successful detection (i.e., Γ́/ **У**). We have plotted the results for relations of different sizes, assuming **α**= 0.01.

Figure (4) Proportion of correctly marked tuples needed for detectability .To compute the proportion of correctly marked to the proposed system we take the relational databases in three size :

1- the size 10000 tuples
2- the size 100000 tuples
3- the size 1000000 tuples

The X-axis refer to the percentage of tuples marked that compute by 1/y*100

The percentage of tuples marked 0.002%, 0.02%, 0.2%, and 2% correspond to the **У** values of 50000, 5000, 500, and 50 respectively,(ex: 1/5000*100=0.02).

the Y-axis refer to proportion of correctly marked tuples, the figure (4) shows that the required proportion of correctly marked tuples decreases as the marked tuples increases. Of course, you need more than 50% of the correctly marked tuples to differentiate a watermark from a chance occurrence, but with an appropriate choice of **У**, this percentage can be made less than 50%. This figure also shows that for larger relations.

The plotted in Figure 5 required proportion of correctly marked tuples for various values of **α**. The results are shown for a 500000 tuple relation. Clearly, we need to proportionately find a larger

number of correctly marked tuples as the value of **α** decreases.

More importantly though, even for very low values of **α**, it is possible to detect the watermark. Even for **У**=50000 where there are only 50 marked tuples out of 500000 tuples. This proportion is good because we use the techniques of stream cipher depending on the hash function.

3.6.3 Robustness

Now analyze the robustness of our watermarking technique against various forms of malicious attacks. Owner has marked **W** tuples. For detecting her watermark, attacker uses the significance level of **α** that determines the minimum number of tuples **Ґ** out of **W** that must have her mark intact.

Bit-Flipping Attack The attacker tries to destroy owner's watermark by flipping the value at the bit positions which guesses have been marked. The analysis and results are similar to the zero-out and randomization attacks.

Assume that attacker magically knows the values of the **ν** and χ parameters used by owner. The value of χ is assumed to be the same for all of **ν** attributes. Since attacker does not know which bit positions have been marked, he randomly chooses ς tuples out of **n** tuples. For every selected tuple, he flips all of the bits in all of χ bit positions in all of **ν** attributes. To be successful, he should be able to flip at least **T`=W-Ґ+1** marks.

3.6.4 Inevitability Attack

In this type of attack the attacker trying to find a key that yields satisfactory the watermark for some value of level of significance α.

For high values of **α**, the attacker can stumble upon such a key by repeatedly trying different key values. This attack failed for using low values of **α**.

3.6.5 Design Trade-Offs

Watermarking technique in proposed system has four important tunable parameters: (i) **α**, the test significance level, (ii) **У**, the gap parameter that determines the fraction of tuples marked, (iii) **ν**, the number of attributes in the relation available for marking, and (iv) χ, the number of least significant bits available for marking. Based on the analysis presented in this section, Figure (6) summarizes in the important trade-offs when selecting the values for these parameters.

4. Conclusions

The following conclusions are drown from the present work:

1- System Analysis shows that the watermark can withstand a wide variety of malicious attacks, because of using the one-way hash function SHA-1, and using Threshold Generator in the watermarking algorithm.

2- The basic idea of the watermarking techniques is to ensure that LSB bit position for some of the attributes of some of the tuples contain specific value. The tuples, attributes within tuple, bit position in an attribute, and specific bit value are all algorithmically determined under the control of private keys and by using SHA-1 algorithm's and Threshold Generator.

3- The detectability and robustness of a watermark depend on the significance level (**α)** and the number of marked tuple **(W**). The latter in turn depends on the number of tuples (**n)** and gap parameter (**У)** that shown in system analysis steps.

4- The proposed system provides an effective and reliable solution to protect valuable numeric relational

data from illegal duplications and redistributions.

5. References

1-Date C. J. , an Introduction to Database Systems, Person Addison-Wesley publishing Company, U.S.A. 2004

2- Radu Sion. *Rights Assessment for Discrete Digital Data, Ph.D. dissertation*. Computer Sciences, Purdue University, 2004

3 -R. Agrawal and J. Kiernan. Watermarking Relational Databases. In Proceedings of 28th *International Conference on Very Large Data Bases*, Hong Kong, China, 2002.

4 -Johnson N.F., Jajodia S., and Duric, Z., *Information hiding: Steganography and watermarking attacks and countermeasures*, Kluwer academic Publishers, 2000.

5 -S. Katzenbeisser and F. A. Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking*.Artech House, 2000.

6 -Shoemaker C., "*Hidden bits: A survey of techniques for digital watermarking*",Independent study, EER 290, spring 2002.

7 -Wang Y. Doherty, J.F., and Van Dyck, R.E., "*A watermarking algorithm for fingerprinting Intelligence images*",
Conference on Information Science and Systems, The John Hopkins University, March 21-23, 2001.

8 -Cox I., Miller M., Linnartz J.P. and Kalker T., "*A Review of Watermarking Principles and Practices*", in Digital Signal Processing for Multimedia Systems (Parhi, K. and Nishitani, T., eds.), Ch. 17, 1999.

9 -B.Schneier.Applied cryptography John Wiley. second edition 1996

10 -American Bankers Association Key Hash Message
Authentication Code ,ANSA x9.71 Washington dc 2000.

Table (1) Notations

n	Number of tuples in the relation
ν	Number of attributes in the relation available for marking
χ	Number of least significant bits LSB available for marking in an attribute
1/y	Fraction of tuples marked
α	Significance level of the test for detecting a watermark
Γ	Minimum number of correctly marked tuples needed for detection
T	Output of Threshold Generator is binary number
τ	Attribute Value
K	Secret Key
WM	watermark

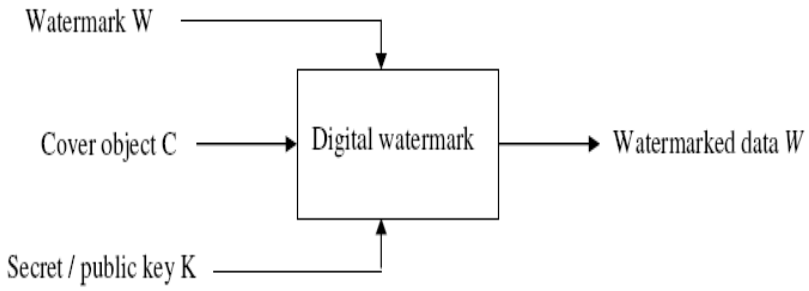

Figure (1) Digital Watermarking – Embedding

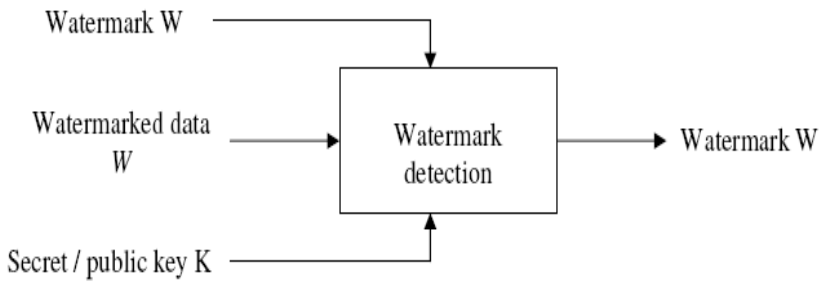

Figure (2) Digital Watermarking – Recovery

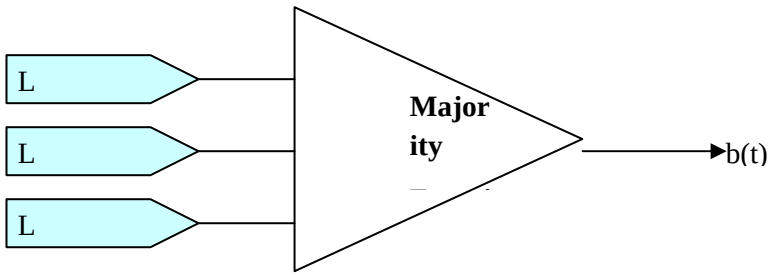

Figure (3) Threshold Generator

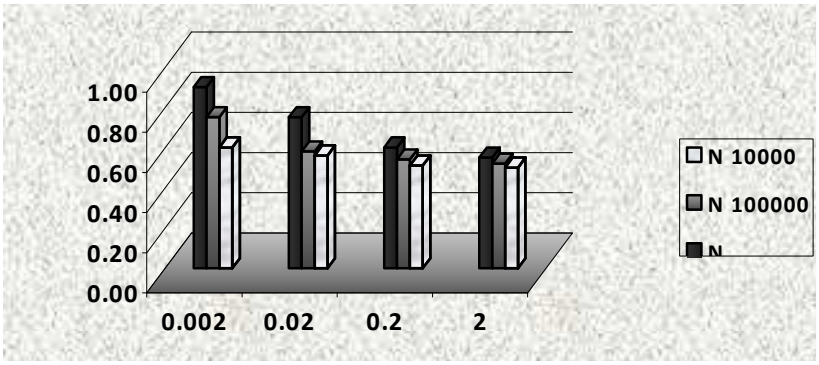

Figure (4)Tuples marked

n =500000 , ν =1 , χ =1

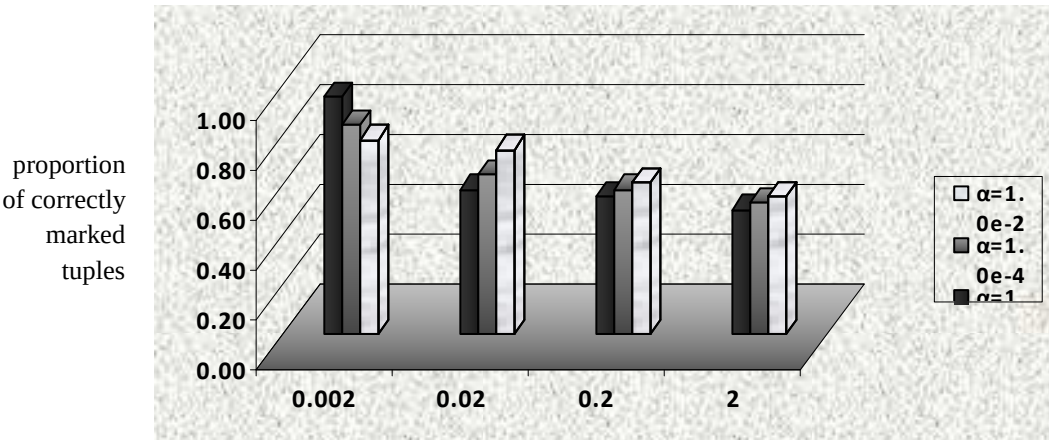

Figure (5) Proportion of Correctly Marked Ttuples need for Decreasing *α*

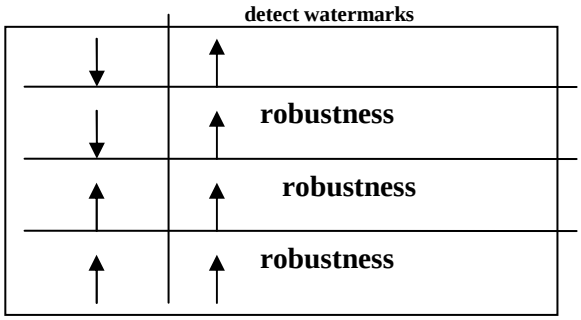

Figure (6) Design Trade-Offs