A
Major Project-II Report
On

ROBUST DATABASE WATERMARKING TECHNIQUE FOR NUMERICAL DATA

Submitted in Partial Fulfilment of the Requirement for the Degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

ARPIT VERMA
2K14/CSE/04
Under the Esteemed guidance of
Mr. MANOJ KUMAR

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road,
Delhi-110042.
JUNE, 2016

CERTIFICATE

This is to certify that Major Project-II Report entitled **"Robust Database Watermarking Technique for Numerical data"** submitted by **Arpit Verma, Roll No. 2K14/CSE/04** for partial fulfilment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the candidate work carried out by him under my supervision.

Mr. Manoj Kumar
Associate Professor
Department Of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the major Project-I1 work entitled "**Robust Database Watermarking Technique for Numerical data**" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master Of Technology (Computer Science and Engineering) is a bonafide report of Major Project-1I carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Arpit Verma
2K14/CSE/04

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Mr. Manoj Kumar for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. O.P.Verma, HOD, Computer Science & Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out. Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Arpit Verma
University Roll no: 2K14/CSE/04
M.Tech (Computer Science & Engineering)
Department of Computer Science and Engineering
Delhi Technological University
Delhi – 110042

ABSTRACT

The recent progress in the digital multimedia technologies has offered many facilities in the transmission, reproduction and manipulation of data. However, this advance has also brought the problem such as copyright protection for content providers. Digital watermarking is one of the proposed solutions for copyright protection of multimedia. A watermark embeds an imperceptible signal into data such as audio, image and video, which indicates whether or not the content is copyrighted. The Watermark is an informational bearing signal and it is an image or pattern identification that appears various shades of lightness/darkness. Watermark used on government document, currency. Watermark also used as security features of passport and other documents. Watermarking consist of embedding watermark with another signal like image, video signal. This process should not degrade the quality of signal. The watermarking communication channel is power limited and band limited. Due to requirement of imperceptible watermark, the power limit arise. And due to low pass multimedia signal like image, audio and video, the bandwidth limit arise. The Watermark signal is either unintentional or malicious interface, which distort the watermark. Claiming ownership rights on database is crucial issue in today's internet based application and in content distribution application. In this thesis we implement various techniques of database watermarking. In this we firstly encrypt the primary key and with the help of encrypted primary key we select the two tuple and then watermark it with the help of another tuple this method is also reversible

Keyword: database, watermarked database, primary key.

List of Figures

List of Figures

List of Abbreviations

DCT	Discrete-Cosine-Transform
DGW	Dynamic Graph Watermark
DWT	Discrete Wavelet Transform
FDOS	Function Dependency Oriented Sequence
HVS	Human Visual System
JNI	Java Native Interface
JVM	Java Virtual Machine
LSB	Least Significant Bits
MAC	Message Authentication Code
MBR	Minimum Bounding Rectangle
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
RDBMS	Relational Database Management System
SDSW	Semblance Based Disseminated Software Watermarking Algo
SIHS	Secure Information Hiding System
SVD	Singular Value Decomposition
SVG	Scalable Vector Graphic
WFF	Well Formed Formula

TABLE OF CONTENTS

Chapter 1

Introduction

The recent surge in the growth of the Internet results in offering of a wide range of web-based services, such as database as a service, digital repositories and libraries, e-commerce, online decision support system etc. These applications make the digital assets, such as digital images, video, audio, database content etc., easily accessible by ordinary people around the world for sharing, purchasing, distributing, or many other purposes. As a result of this, such digital products are facing serious challenges like piracy, illegal redistribution, ownership claiming, forgery, theft etc. Digital watermarking technology is an effective solution to meet such challenges. A watermark is considered to be some kind of information that is embedded into underlying data for tamper detection, localization, ownership proof, traitor tracing etc.

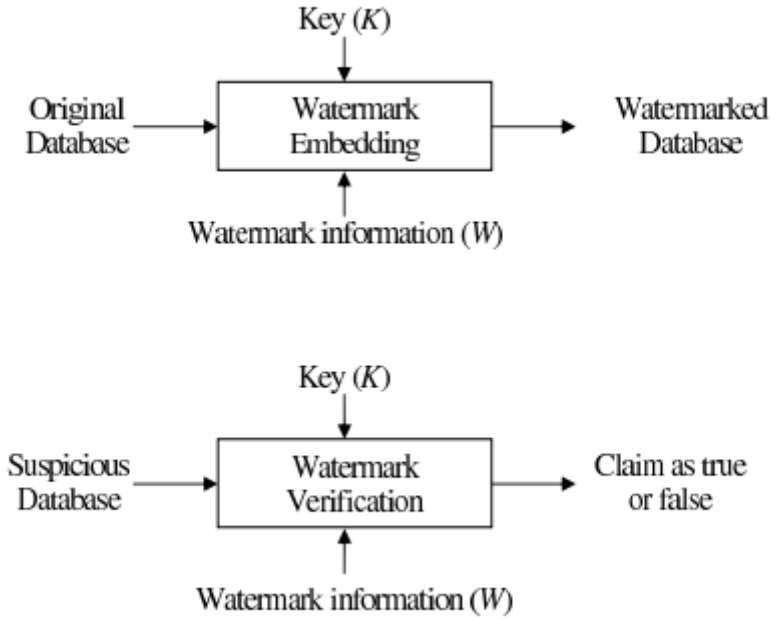

Figure1.1: Basic Watermarking Technique

Initially, most of the work on watermarking was concentrated on watermarking of still images, video, audio, VLSI design etc [2], [3], [4]. However, in the recent years watermarking

of database systems started to receive attention because of the increasing use of it in many real-life applications. Examples where database watermarking might be of a crucial importance include protecting rights and ensuring the integrity of outsourced relational databases in service provider model [5], in data mining technology where data are sold in pieces to parties specialized in mining it [6], online B2B interactions [7] etc. The idea to secure a database of map information (represented as a graph) by digital watermarking technique was first coined by Khanna and Zane in 2000 [8]. In 2002, Agrawal et al. proposed the idea of digital watermarking for relational database [9].

In general, the database watermarking techniques consist of two phases: *Watermark Embedding* and *Watermark Verification*. During watermark embedding phase, a private key *K* (known only to the owner) is used to embed the watermark *W* into the original database. The watermarked database is then publicly available. To verify the ownership of a suspicious database, the verification process is performed where the suspicious database is taken as input and by using the private key *K* (the same which is used during the embedding phase) the embedded watermark (if present) is extracted and compared with the original watermark information. Figure 1 depicts the basic database watermarking technique.

The relational data defers from multimedia data in many respects: (*i*) Few Redundant Data: Multimedia objects consists of large number of bits providing large cover to hide watermark, whereas the database object is a collection of independent objects, called tuples. The watermark has to be embedded into these tuples, (*ii*) Out-of-Order Relational Data: The relative spatial/temporal positions of different parts or components in multimedia objects do not change, whereas there is no ordering among the tuples in database relations as the collection of tuples is considered as set, (*iii*) Frequent Updating: Any portion of multimedia objects is not dropped or replaced normally, whereas tuples may be inserted, deleted, or updated during normal database operations, (*iv*) There are many psycho-physical phenomena based on human visual system and human auditory system which can be exploited for mark embedding. However, one cannot exploit such phenomena in case of relational databases.

Due to these differences between relational and multimedia data, there exist no image or audio watermarking method which is suitable for watermarking of relational databases. These differences give rise to many technical challenges in database watermarking as well.

1.1. Motivation

The use of digital data has become popular due to the availability of inexpensive resources including computer, software and the internet. It is now possible to store and

transmit digital media with high reliability. It is also possible to effortlessly to modify and replicate digital media with modern signal processing tools. Digital multimedia data processing is preferred over its analog counterpart in most application

Digital watermark has been used in number of application [10, 11, 18, 14]

Some important application are:

1. **Copyright Protection**: The owner of multimedia data may be embed a watermark that conveys copyright and ownership information. The watermark can prove ownership in court.

   Watermark can be used in conjunction with encryption [12, 13] to provide efficient copyright protection. Encryption ensure that the digital data is available to only the authorised users, i.e. user who have access to the decryption key. Since data must be decrypted to be used, it is vulnerable at some point in the system. Watermarking can complement digital encryption, thereby ensuring that the owner always have some form of control over their multimedia data.

2. **Copy Control**: A watermarking can be used in copy control system to enable or disable copying. The recording device may decide to allow or inhibit recording depending on the information conveyed by the watermark. Such a system has been proposed for allowing a copy once feature in digital video disk recording [1]. A similar feature can be incorporated into playback device [17].
3. **Fingerprinting:** The owner of multimedia data embeds a watermark that is unique to a particular copy (user) of the work. The watermark act as a digital fingerprint since it is associated with a unique copy of work. This can be useful tool in tracing the source of illegal copies of work. The paper by Cox et. al [10] provides further application of fingerprinting.
4. **Broadcast Monitoring:** In this application, a watermark is embedded in advertisement that are aired by broadcasting television station. An automated monitoring system can then track the number of times the advertisement is aired on television. This information can be used by the advertisers of book keeping purposes.
5. **Authentication:** We may need to ensure the authenticity or integrity of multimedia signals in a certain application such as legal cases and medical emerging. Fragile Watermarking [24] are used to indicate whether the data has been altered. This watermarks can also provide information about the location

of the host that has undergone alteration. Algorithm that are robust to compression such as JPEG but not to intentional tempering have been proposed [15, 16].

6. **Meta-data Tagging:** Mata-data convey auxiliary information about the host multimedia signal. Embedding information related to a patient in medical images or a time stamp in photographs are typical application of meta-data embedding. Existing methods include embedding the information in the header of data file and in the perceptually unimportant regions of signals like images and video. However, these methods are vulnerable to simple operations such as changing of the file format and image cropping.

   Watermarking is natural and effective way to transmit meta-data. By designing a robust algorithm, we may guarantee that the embedded data available even after the host signal has undergone alterations.

7. **Error Concealment in Images and Video:** Data hiding for error concealment has been proposed by Robie and Mersereau [20]. The data required for error correction is embedded as watermark and transmitted to the video decoder. The decoder uses this data in conjunction with error concealment technique to recover from channel error.

1.2 Watermarking Requirements

A watermark must be robust, secure and imperceptible. The design of a watermarking algorithm usually involves a trade-off among the above requirements. The requirements of an algorithm are described below.

1. **Imperceptibility:** Many application involving multimedia hosts such as audio, images and video required unobtrusive watermarks. We require that the watermarked host and the original host be perceptually indistinguishable.

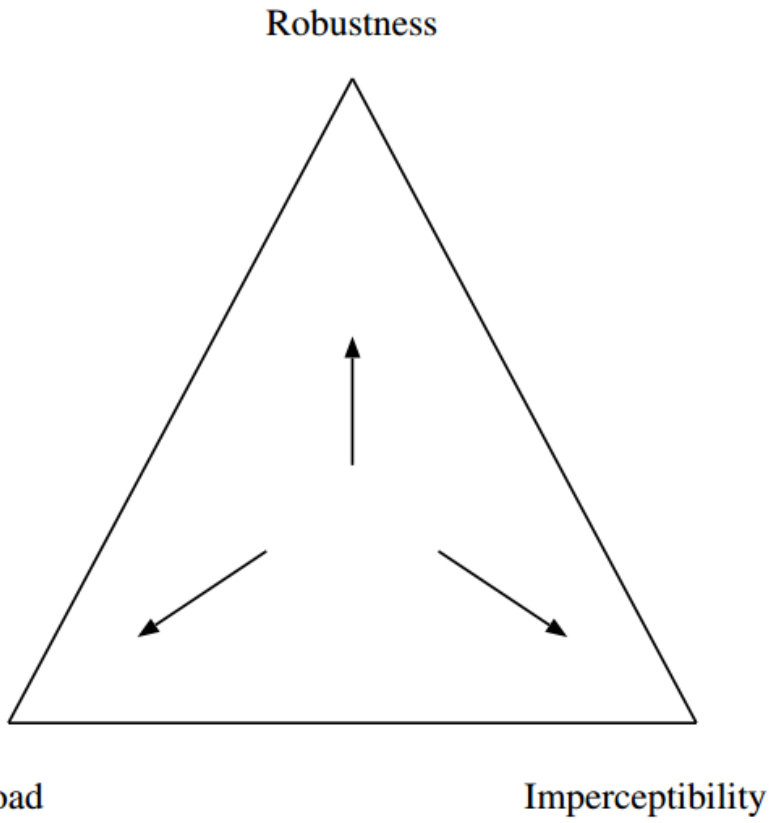

Figure1.2: Figure illustrating the relationship between imperceptibility, robustness and payload. It is clear that there is a trade-off between the three basic requirements.

This Requirement arises due to need to perceptual quality and consequently the commercial value of the multimedia host.

Podilchuk and Zeng [19] have developed a method that exploits the properties of the Human Visual System (HVS). The idea [19] is to limit the changes in this host due to embedding to just below the threshold of perception to render the watermark invisible. Podilchuk and Zeng calculated the threshold of visual perception by computing the Just Noticeable Difference (JND). The paper by Vleshchouwer et. al [22] overviews research in perceptual watermarking.

2. **Robustness:** Robustness refer to the ability of the watermark to be resilient to the interference. The information must be extracted reliably from a corrupted watermark. The interference may be either malicious or unintentional. The unintentional interference may arise from content processing such as compression, filtering, re-sampling, digital-to-analog(D/A) and analog-to-digital (A/D) conversion. On the other hand, a knowledgeable attacker may

intentionally try to destroy the watermark. Examples of intentional attacks include collusion and averaging attacks wherein an un-watermarked host is constructed from several copies of the watermarked host [21, 23].

Attack modelling for digital watermarking is an active research area. We refer the reader to the paper by Voloshynovskiy et. al [23] for a tutorial on attack modelling and countermeasures.

3. **Payload:** the amount of information to be conveyed by a watermark depends on intended application. Applications such as data authentication and finger-printing require only one bit of embedded information since it is required only to verify whether a given watermark is present or not. Applications such as copyright protection and meta-data tagging required more than one bit of information to be embedded in the host. Some have suggested that at least 128 bits of information are needed for efficient copyright protection [15].

4. **Oblivious and Non-Oblivious watermarking:** In some applications such as broadcast monitoring, it is possible for the watermark extraction algorithm to have a copy of the original (un-watermarked) signal. This scenario is referred to as informed or non-blind watermarking. Oblivious or blind watermarking refers to the case wherein the information is extracted without the knowledge of the original host signal.

Figure 2. Illustrates the relationship between robustness, imperceptibility and payload requirements of a watermarking system. It is difficult to simultaneously meet these conflicting requirements in a watermarking system, the design of a watermarking algorithm involves a trade-off between these requirements.

1.3 Ownership Authentication

Software developers are trying to find methods to stop the piracy of software's but, unfortunately no single technique is currently strong enough to protect the piracy of software's. Therefore software piracy is one of the main direct threat to software industry and may affect the interests of software developers or providers. It directly affects the revenue of software vendors. One of the most promising attempts to protect intellectual property rights includes software watermarking, which is a kind of a watermark based on human perception.

1.4 Watermarking

Watermarking is the process of hiding additional information within software codes, digital data (such as image, audio, and video) and documents in such a way that it is nearly undetectable. Digital watermarking technique refers to the process of embedding the given watermark information (such as ownership information, name, logo, signature, etc.) in the protective information (such as picture, audio, video, or text) and picking the given watermark information from the protective information, which is not perceived by human perceptual system**.** In other words, watermarking is a process of embedding a digital watermark or signal containing information unique to the copyright owner in the object (text, image, audio, or video) which is needed to be protected. A digital watermark is defined as a visible or invisible identification code that is permanently embedded in the data, to transmit hidden data. It remains present in the data even after the decryption process. It usually provides copyright protection of intellectual property. The watermark is later used to identify the actually copyright owner of the object. Watermarking techniques can be divided into various types as shown in the **fig. 1.3** [20].

According to the human perception, the digital watermarks can be divided into Three different types as follows.

- Visible Watermark
- Invisible-Robust Watermark
- Invisible-Fragile Watermark

From application point of view digital watermark could be

- Source Based or
- Destination Based.

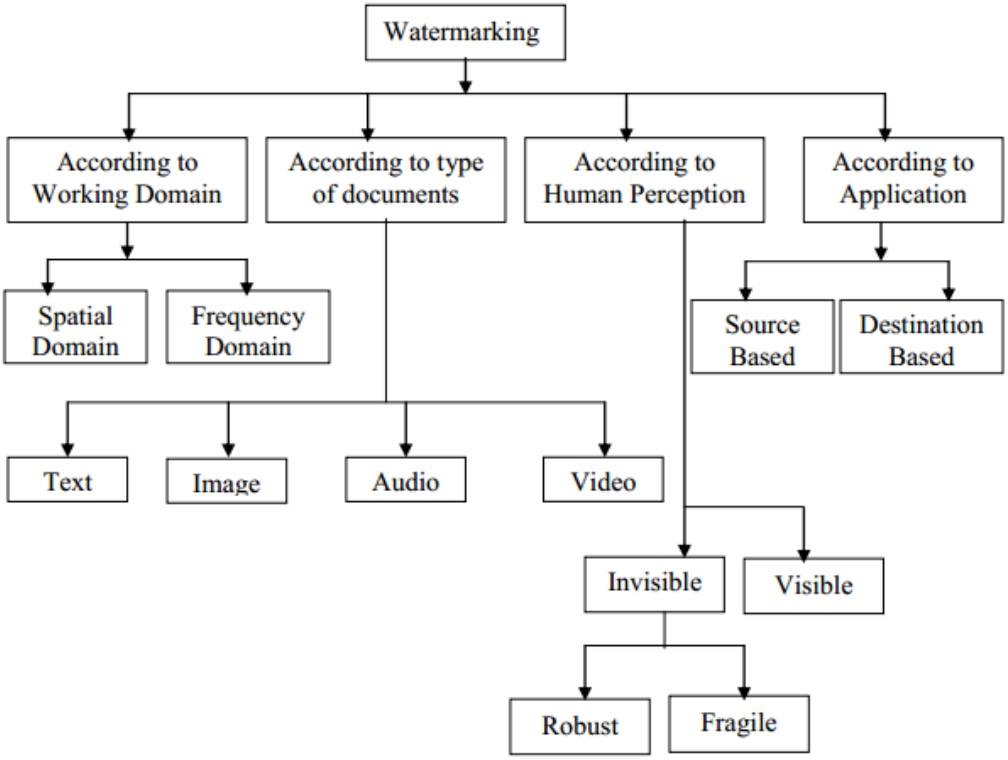

Figure 1.3: Types of Watermarking Techniques

Source-based watermarks are desirable for ownership identification or authentication where a unique watermark identifying the owner is introduced to all the copies of particular software being distributed. A source-based watermark could be used for authentication and to determine whether received software or other electronic data has been tampered with. The watermark could also be destination based where each distributed copy gets a unique watermark identifying the particular buyer. The destination based watermark could be used to trace the buyer in the case of illegal reselling

1.4.1 **Visible and Invisible watermark**

The visible watermark is a secondary translucent overlaid into the primary software codes. The watermark appears visible to a casual viewer on a careful inspection.

The invisible-robust watermark is embedded in such a way that alternations made to the software codes are perceptually not identified. The invisible-fragile watermark is embedded in such a way that any manipulation or modification of the software codes would alter or destroy the watermarks.

1.4.2 Dual Watermark

Dual watermark is a combination of a visible and an invisible watermark [20][29]. In this type of watermark an invisible watermark is used as a backup for the visible watermark as is clear from the **Fig. 1.4.**

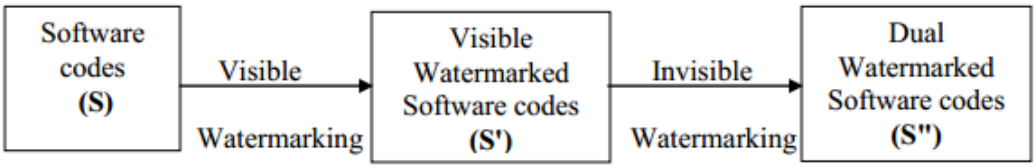

Figure 1.4: Schematic Representation of Dual Watermarking

1.4.3 Robust and Fragile Software Watermark

Robust software watermarks can be extracted even if they have been subjected to adversarial or casual semantics-preserving or near-semantics preserving code translation. Such watermarks are used in systems that prevent unauthorized uses or prevention and in systems that make public claims to software ownership [37]. Fragile software watermarks will always be destroyed when the software has been changed. Such watermarks are used in integrity verification of software and in systems that allow limited change and copy

1.5 Software Watermarking

Software watermarking is a method of software protection by embedding secret information into the text of software. We insert such secret information to claim ownership of the software. This enables the copyright holders to establish the ownership of the software by extracting this secret message from an unauthorized copy of this software when an unauthorized use of this software occurs. Software watermarking can be regarded as a branch of digital watermarking, which started from 1954 [25]. Since the publication of a seminal work by Tanaka et al. in 1990 [24], digital watermarking has made considerable progress and become a popular technique for copyright protection of multimedia information. Research on software watermarking started in the 1990s. Authors of papers [31] have given brief surveys of software watermarking research

The software watermarking is a new research area that aims at providing copyright protection of commercial software. It is relatively new software protection technique, which appeared in recent decade and whose basic principle is to embed secret information as the evidence to identify an ownership. This technique is also used in other kinds of protection and enforcement of intellectual property rights such as text, digital images, digital audio, digital video etc. The software can be protected by two main approaches namely hardware-based and software-based [3].

In the hardware-based technique, the developers or providers used additional hardware components such as a specific CD, smart card etc. to execute the software. It is impossible to execute the software without the presence of a trusted hardware component. The hardware-based technique could provide comparatively harder protection so that usually it is difficult for the crackers to completely crack the software protection. On the flip side, this technique faces some tough issues like hardware compatibility. In addition, it is not convenient and user friendly and will definitely add to the cost.

In the software-based technique, the developers or providers used earlier registration codes, license files, selling of the codes and many other methods, which protect the software merely via the software itself. The most common implementation technique is to put the registration on the client. It requires a legal token to give the user permission to use. The token may be a license key, a license file or an activation code and so on. The software codes are copied by most of the people due to the following reasons [7].

- Software is intangible and/or non-exclusive
- Everyone does it
- It is very easy to copy software codes
- It does not harm anyone
- The low quality of software
- Software is expensive
- The risk is minima

1.6Techniques of Copyright Protection

Various techniques of copyright protection of software codes have been defined. These techniques are categorized as static watermarks and dynamic watermark

1.6.1 Static Watermarks

A static watermark is stored inside program code section in a certain format and it does not change during the program execution. According to the representation of static watermark information, this can be represented by two types: data watermarks and code watermarks.

A data watermark stores watermark information as program data and can be stored anywhere inside a program such as in comments or in variables. A code watermark is represented by choosing a particular sequence of instructions, in cases (and these are common) where more than one sequence of instructions has an equivalent effect. A static code watermark may also be stored in "dead code" (which is never executed); any sequence of instructions may be used with equivalent effect in a dead-code area. For example, in a Java program, a particular order of cases in a switch statement can be used to represent a watermark number. The first static watermarking technique for software codes was developed by reordering its basic blocks [3]. The other kind of static watermark for application software are static graph-based algorithms, register allocation algorithm, function dependency oriented sequencing [15], Asymmetric key for RDBMS [20].

1.6.2 Dynamic Watermarks

The dynamic watermarks are generated during program execution and stored in the program execution states. The dynamic watermarking hides the watermark in data structures that are built specially for embedding purpose during execution of the program. Such type of watermark is extracted based on the output of program for a given series of input data and extracted at run time [5][6]. The dynamic watermarks are B+-trees, R-trees and linked lists [25], Graph colouring [17] and Character Segmentation Method[21]. There are three kinds of dynamic watermarks: Easter Eggs, Execution Trace Watermarks and Dynamic Data Structure Watermarks [48].

1.6.3 Easter Egg Watermarks

Easter Egg Watermarks are revealed at the user interface, by a secret sequence of input. These secret messages may be copyright information or an unexpected text or image.

1.6.4 Execution Trace Watermarks

In Execution Trace Watermarks when program is run, the watermark information will be presented in the instruction or data-reference trace of program execution sequence. A special sequence of inputs may be required. This watermark can be extracted by analysing the address trace and/or the sequence of operations of data-reference trace special sequence a watermarked program.

1.6.5 Data Structure Watermarks

In Data structure watermarks, program is executed with a special input sequence and watermark information will be left within the state of the program such as the heap or the stack. This watermark can be retrieved by analysing the runtime memory state of the program. The Dynamic Graph Watermark (DGW) [25][48] belongs to data structure watermarks category.

1.6.6 Other Watermarks

Another categorization of software watermarking and more generally for watermarking methods is based on whether the original code is needed or not. The methods which require the original code (without watermark data) to extract watermark are called "Informed" methods. Methods which can extract watermark using only the watermarked code are called "canopy" methods.
The software watermarking methods can be categorized into three major categories for hiding the watermark data. The methods of first category always declare all variables of the codes in the form of which start with first letter of their data types. Such types of watermarks will not change the original codes of program. The methods of second category add some line of codes (dummy blocks) or reorder the mathematical expression of the source code. The lines of codes are put in places such as in an if-then-else statement. In the statement, if the if condition is always false then the line of codes will never run, these lines of codes can hide the watermark. The reordering of an equation uses changing the order of equation operands. This method will not change the source codes. These methods work on programming languages like C, C++ etc. These methods simply add new codes to program and do not change the original code of program [13].
The methods of third category work on low level languages such as assembly

or Java bit code. They try to change the order of the registers used for operations and hide information in the registers used. These methods work on a very low level language and only change the register ordering algorithm. The method explained in low level language is HDL- based IP module protection [18], Compiler based infrastructure [16] etc.

1.7 **Watermarking Properties**

An effective watermark should have several properties [20], whose importance will vary depending upon the application. There are a number of papers that have discussed the characteristics of watermarks [31- 33]. Some of the properties discussed here are robustness, imperceptibility, security, un-detectability and capacity. In practice, it is not possible to design a watermarking system that excels at all of these. Every watermarking technique makes trade-offs between them, taking into account the application domain.

In the following subsections we describe each of the properties mentioned above and discuss how its importance and definition vary with application. The fundamental properties of a watermark are as follows:

1.7.1 Robustness

The watermarked contents may undergo various attacks before the watermark is retrieved, where the attack is defined as any alteration of contents that can damage the watermark [34 - 35]. Resistance against attacks is the key issue when designing a watermarking system. The watermarking system should be resistant against any intentional or unintentional processing of the watermarked contents that can be an image, audio, video, or text. This attribute of a watermarking system is called robustness (aka fidelity). If the watermark can survive in the object even if it was tampered, it is called a robust watermark.

Robustness also means that it must be nearly impossible to defeat a watermark without degrading the marked contents to a large extent, to the extent that the contents remain no longer useful and valuable. While designing a watermarking technique, it is necessary to bear in mind the intended application and the corresponding set of conceivable attacks. Secondly, we must strive to achieve robustness, making it resistant against attacks.

1.7.2 Security

Another property of any watermarking scheme is security. The authorship information (watermark) must remain hidden from unauthorized detection. Security of a watermarking system means that the watermark, its existence and the payload must remain secret. Unauthorized parties should not be able to detect the existence of a watermark and its size.

1.7.3 Imperceptibility

It is the fundamental requirement of watermarking meaning thereby that the watermark must be embedded in the object imperceptibly. To preserve the quality of watermarked contents, the watermark should not noticeably distort the original content.

The original and the watermarked objects should look similar and ideally, the original content and watermarked content should be perceptually identical. The watermark should not be perceived by the viewer or the watermark should degrade
The quality of content.

1.7.4 Capacity

Capacity means the number of bits the technique can encode in a unit amount of time. It indicates the upper limit of watermark length. Generally, capacity of any watermark scheme should be high. However, different applications have different capacity requirements for example, in case of broadcast monitoring very high capacity is required.

1.8 **Thesis Organization**

Chapter 1 Introduction: Describes the introductory part of watermarking, software watermarking, Software Piracy, Ownership Authentication, Fragility, Possible Attack etc.

Chapter 2 Review of Literature: Describes the related work done in this field.

Chapter 3 Proposed Work: In this chapter we have described what watermark techniques we used

Chapter 4 Result: In this chapter we have described what Result we are getting when we perform watermarking

Chapter 5 Conclusions and future scope of research work

Chapter 2

LITERATURE REVIEW

This chapter deals with the explanation of various existing watermark embedding and extracting techniques for various types of software's such as structural, modular, RDBMS, web based online applications and images.

Watermarking is a method of protection of images, audio, video, software and data by embedding secret information into them. We insert such secret information to claim ownership for them. This enables the copyright holders to establish the ownership of the software by extracting this secret message from an unauthorized copy of this software when an unauthorized use of this software occurs. The watermarking technique used for protecting images, audio software and data video is called digital watermarking and watermarking technique used for protecting software and data is called software watermarking which is a branch of digital watermarking.

Software watermarking started in 1954 [38] but the publication of a seminal work by Tamada et al. came in 1990 [24]. Digital watermarking has made considerable progress and become a popular technique for copyright protection of multimedia information. Research on software watermarking started in the 1990s.

The patent by Y. Hao G. Bao and H. Zhang, [49] presented the first published software watermarking algorithm. The preliminary concepts of software watermarking also appeared in paper [38] and patents [24] [28]. Collberg et al. presented detailed definitions for software watermarking [38]. Unlike other fields of digital watermarking such as multimedia watermarking, software watermarking has not received sufficient attention yet. Authors of papers [30 - 31] have given brief surveys of software watermarking research.

Various research methodologies presented by different researchers for digital, software and database watermarking has also been discussed here along with. The research work currently available for software, digital and database watermarking.

2.1 **Software Watermarking Algorithm**

In this section we have described major software watermarking algorithms which are defined by different researcher. The algorithms are R+ tree data structure method [25], Basic block reordering algorithms [33], Semblance Based Disseminated

Software Watermarking Algorithm [37], Register allocation algorithms[36], Spread-spectrum algorithms[42], Mobile agent watermarking algorithms [41], Threading algorithm, Abstract interpretation algorithm, Metamorphic algorithm, Dynamic path algorithm and Graph-based algorithms [44]. We especially focus on the CT algorithm for software watermarking and the constant encoding technique [33] for protecting the software.

2.1.1 R+ tree data structure method

Ibrahim Kamel and Qutaiba Albluwi [25] describes about robust software watermarking through R+ tree data structure method. The R-trees are extension of the B+-tree for multidimensional objects. In R+ tree they uses data in a two dimensional space; however, an R-tree and its variants work for any number of dimensions. A geometric object is represented by its minimum bounding rectangle (MBR) in **Fig. 2.1**. Non-leaf nodes contain entries of the form (ptr,R) where ptr is a pointer to a child node in the R-tree; R is the MBR that covers all rectangles in the child node as depicted in **Fig. 2.2**. Leaf nodes contain entries of the form (obj-id, R) where obj-id is a pointer to the object description and R is the MBR of the object. R-tree allow nodes to overlap. This way, the R-tree can guarantee at least 50% space utilization and at the same time remain balanced.

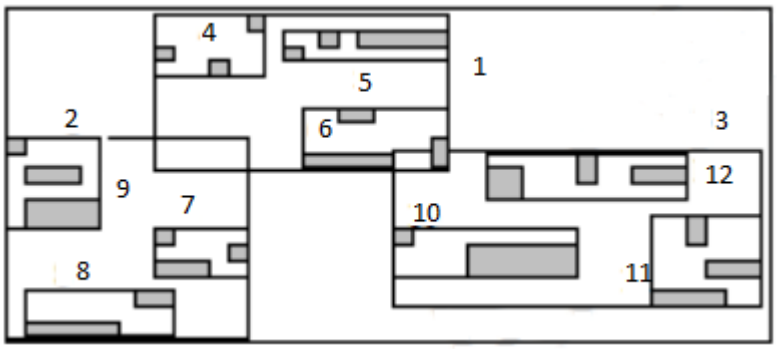

Figure 2.1: Geometric Object

Each node in the tree (except the root) contains between m (minimum number of entries per node) and M (maximum number of entries per node) entries, where m = M/2. At the same time, each non-leaf node (except the root node) has between m and M child nodes. R-tree is a balanced tree; meaning all leaves appear on the same level. The maximum height of the tree can be calculated using the formula h = | $\log_n$ N – 1 |

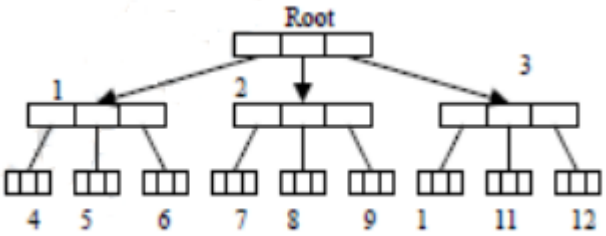

Figure 2.2: R-tree

2.1.2 Basic Block Reordering Algorithm

Y. Hao, G. Bao and H. Zhang [49] published the first software watermarking algorithm. It embeds a watermark into a program by reordering the basic blocks of the program. In a program, a basic block is a set of sequential instructions with a single entry point and a single exit point. The Davidson-Myhrvold algorithm first chooses a group of basic blocks in an executable, then reorders them to form a watermark with some special feature. This reordering needs to maintain the original flow of execution so that the function of the original program is unchanged. Checking the order of this group of blocks enables us to extract the inserted watermark. It is easy to attack software watermarks inserted by this algorithm; if we use this algorithm to watermark the watermarked program again, the original watermark will completely be destroyed. This algorithm can be enhanced by using opaque predicate to establish false dependencies among basic blocks, making it difficult to remove them.

2.1.3 Semblance Based Disseminated Software Watermarking Algorithm

Zeeshan Pervez, Noor-ul-Qayyum, Yasir Mahmood and Hafiz Farooq Ahmad [2] have describes the software watermarking techniques for the virtual environment like JVM (Java Virtual Machine). This is based on java based applications. Semblance Based Disseminated Software Watermarking (SBDSW) is designed to encode the secret information which is added to the program after compilation. This encoding of secret information within the program is achieved by adding dummy instructions which are hard to identify and replace. The SBDSW algorithm is divided into following steps [2].

- Watermark Encoding.
- Dictionary Mapping.
- Instruction Embedding.
- Watermark Recognizer.

2.1.3.1 Watermark Encoding

Watermark encoding for JVM works by manipulating the classes of programs. Each class of programs are represented by another smaller representation which was calculated with hashing or by manipulating the local variables or instruction of a methods

2.1.3.2 Dictionary Mapping

Dictionary mapping maps to set of possible dummy instructions or variables which are inserted in the program to encode watermark.

2.1.3.3 Instruction Embedding

Instruction Embedding explain various techniques for watermarking like watermark the whole class, every method or selected methods.

2.1.3.4 Watermark Recognizer

Watermark Recognizer recognizes the method to extract the watermark by tracing the dummy instructions. The dummy instructions are scrutinized by scanning the program and each time an instruction counterparts to the one in dictionary its corresponding binary is recorded. The recorded binaries are used to reveal the Key to unmask the legitimate buyer.

2.1.4 Register Allocation Algorithms

Kang Hui, Liu Jing, Zhu Xiao-dong and Zhang Xiao-xu, [47] have discussed some techniques to watermark solutions to constraint problems such as the graph-colouring problem. The QP algorithm is one of them. It aims to watermark solutions to the graph-colouring problems to protect their intellectual properties. The graph colouring problem concern allocating as fewer colours as possible to the vertices of a graph so that no vertices connected by an edge in the graph receive the same colour.

Myles and Collberg implemented the QP algorithm for the first time for software watermarking through register allocation [38]. They pointed out that the QP algorithm has a serious flaw since it does not permit reliable recognition. Myles and Collberg proposed a new version of the QP algorithm, the QPS algorithm, which allows robust extraction of watermark in the absence of attacks. Unfortunately, after extensive

evaluations, they concluded that the QP algorithm, as well as the QPS algorithm, is unsuitable for software watermarking of architecture neutral codes in the presence of determined attackers.

Kang Hui, Liu Jing, Zhu Xiao-dong and Zhang Xiao-xu, [47] discussed certain misunderstandings in the QP and QPS algorithms and determined the un-extractability of the QP and QPS algorithm through examples. They went on to propose an improvement for the QP algorithm and introduced some potential topics for further research.

2.1.5 Spread-spectrum Algorithms

The spread-spectrum watermarking method was originally developed for watermarking digital media [35]. It represents the data of a document as a vector and modifies each component of the vector with a small random amount. This small amount is called a watermark. Such watermarks can be recognized by correlation with the extracted watermark signal. The spread-spectrum software watermarking procedure consists of three steps: representation extraction, watermark insertion and watermark testing.

D. Curran, M. O. Cinneide, N. Hurley and G. Silvestre [27] have proposed a spread-spectrum software watermarking method which uses call graph depth as a signal. In this algorithm, at first, a vector from a running program is extracted. The call graph depth is measured at distinct points during the execution of the program to be watermarked on certain particular input. In the end, the program code is modified so that its call graph depth is changed, such that it expresses the watermark when this input is given to the program. J. P. Stern, G. Hachez , F. Koeune and J. J. Quisquater [15] have proposed the SHKQ algorithm to apply the spread-spectrum watermarking method to software watermarking. In the SHKQ algorithm, code is viewed not as a set of sequential instructions, but as a statistical object. What it really marks is the frequency counts of sets of consecutive instructions. It extracts a group of representation of the code by the Vector Extraction Paradigm proposed by Stern et al. and then applies the spread spectrum techniques to insert watermarks.

C. Collberg and T. R. Sahoo [43] have implemented the SHKQ algorithm in the software watermarking research tool Sand Mark. They introduced method overloading to increase the frequency of patterns in cases where code insertion and code

substitution are not enough to achieve an acceptably strong watermark signal. Furthermore, they experimented with various attacks on this algorithm.

2.1.6 Stern-based Collusion-Secure Software Watermarking Algorithm

Jieqing Ai, Xingming Sun, Yunhao Liu, Cox, I.J., Guang Sun and Yi Luo [25] have proposed a collusion-secure software watermark algorithm and implemented it in MSIL by simplifying the difficulty of realization of Stern algorithm. Their solution has the ability of resisting many attacks by semantics-preserving program transformations, including compilation, optimization, obfuscation and dead-code removal, etc.

In this algorithm they define the following embedding and extraction procedure:

2.1.6.1 Embedding and Extraction procedure

1. Compute the number of occurrences of each element in code book V to form the vector.
2. Input secret key k and user information W, using a security pseudo-random number compute the watermark w = (w_0, w_1, w_2, …….. w_n)
3. Rearrange the order of elements in vector c to form c' by the secret key k
4. Modify the code in such a way to make sure that the new extracted vector b from watermarked application is equal to c' +w.

Modification the occurrences of the instruction groups are done in an iterative manner. A sophisticated modification is done to the code while preserving its correctness and the new occurrence vector is computed. If the new vector is closer to b, the process is continued, else the modification is refused.

The Watermark Extraction algorithm compares [37] the occurrence vector extracted from the original code with watermarked code. If they differ by approximately w. we conclude that the watermark is present.

2.1.7 Dynamic Graph Watermark

Yang-Xia Luo, Jian-Hua Cheng and Ding-Yi Fang [44] have proposed the AB algorithm to not only compensate for the two deficiencies of the CT algorithm, but also to prevent an attacker to access to the original watermark through a thorough analysis of running stack. The shortfall of the AB algorithm is that sharing algorithm will expand

watermark data and relatively reduce the ability to accommodate the watermark information [45].

Jianqi Zhu, YanHeng Liu and KeXin Yin [29] have proposed two dimensional IPPCT (2D_IPPCT) and a multi-dimensional IPPCT (MD_IPPCT). 2D_IPPCT structure has greater data rate and it is proved that under the circumstances of the same number of leaf nodes, the data rate of MD_IPPCT will increase to a limit with dimension number n increasing. They proved that there exists a two dimensional area, when the data rate is in this area, the DGW system will have a higher overall performance. In other words, 2D_IPPCT structure not only inherits the stability of PPCT structure, but also has higher encoding rate.

Yin Ke-xin, Yin Ke and Zhu Jian-qi [35] have proposed a robust software watermarking based on Shamir threshold scheme and chaotic encryption. In this approach they firstly split the watermark into pieces using the Shamir threshold scheme, which helps to retrieve the original watermark with partial information & increase resilience and then the pieces are run through 1-D logistic chaotic encryption before embedded into the dynamic branch structure of program that improves robustness further. By this technique the error-correction can be used to make it resilient against various attacks

XiaoJiang Chen, DingYi Fang, JingBo Shen, Feng Chen, WenBo Wang and Lu He [46] and B. Horne, L. Matheson, C. Sheehan, R.Tarjan [37] have proposed a novel approach of dynamic graphic software watermark. In this method, they created many fake watermarks through encoding multi-constant and enhance the stealth & anti-attack ability to keep software from damage. Meanwhile, a detailed analysis in theory is made in terms of the principle, feasibility and merits of this approach. A series of tests are made on the basis of the prototype system, analysing the two sub-systems separately about their validity, robustness and performance overload caused by watermark embedding. They also analysed bit-rate of the three graph watermark structures in the system theoretically. Graphically it is shown in **Fig. 2.3**.

2.1.8 Dynamic Path Algorithm

C. Collberg, E. Carter, S. Debray, H. Huntwork, J. Kececioglu, C. Linn and M. Stepp [45] have inserts watermarks in the runtime branch structure of a program to be watermarked. This algorithm is based on the observation that the branch structure is an essential part of a program and that it is difficult to analyse the branch structure

completely because it captures so much of the semantics of the program. The implementation of this algorithm has three stages. In the first one, the tracing stage, we determine the dynamic behaviours of the un-watermarked program by tracing its execution path on a particular input sequence. Then suitable points to insert the watermark must be found. Next, in the embedding stage (modifying the sequence of branches taken and not taken, on the secret input sequence) embeds the watermark into the program. Lastly, during extracting stage, we trace the program again using the secret input sequence and check the branch sequence to extract the watermark.

2.1.9 The Threading Algorithm

F. Hartung and M. Kutter [11] proposed a threading software watermarking algorithm and implemented it for Java bytecode. In this study we call it the NT algorithm. This algorithm takes advantage of the intrinsic randomness for a thread to run in a multithreaded program. Since it is very hard to analyse such a program, this algorithm claims resilience. In the NT algorithm, the process of embedding watermarks is divided into two steps. Firstly, creating multiple threads of execution – the number of possible execution paths through the program is increased. Inserting suitable locks in certain positions maintains the semantics of the old program. Secondly, locks are added to ensure that only small subsets of the possible paths are actually executed by the watermarked program. The watermark is embedded in those execution paths.

2.1.10 The Abstract Interpretation Algorithm

F. Hartung and M. Kutter [11] devised the abstract interpretation algorithm to embed the watermark in values assigned to designated integer local variables during program execution. These values can be determined by analysing the program under an abstract interpretation framework, enabling the watermark to be detected even if only part of the watermarked program is present.

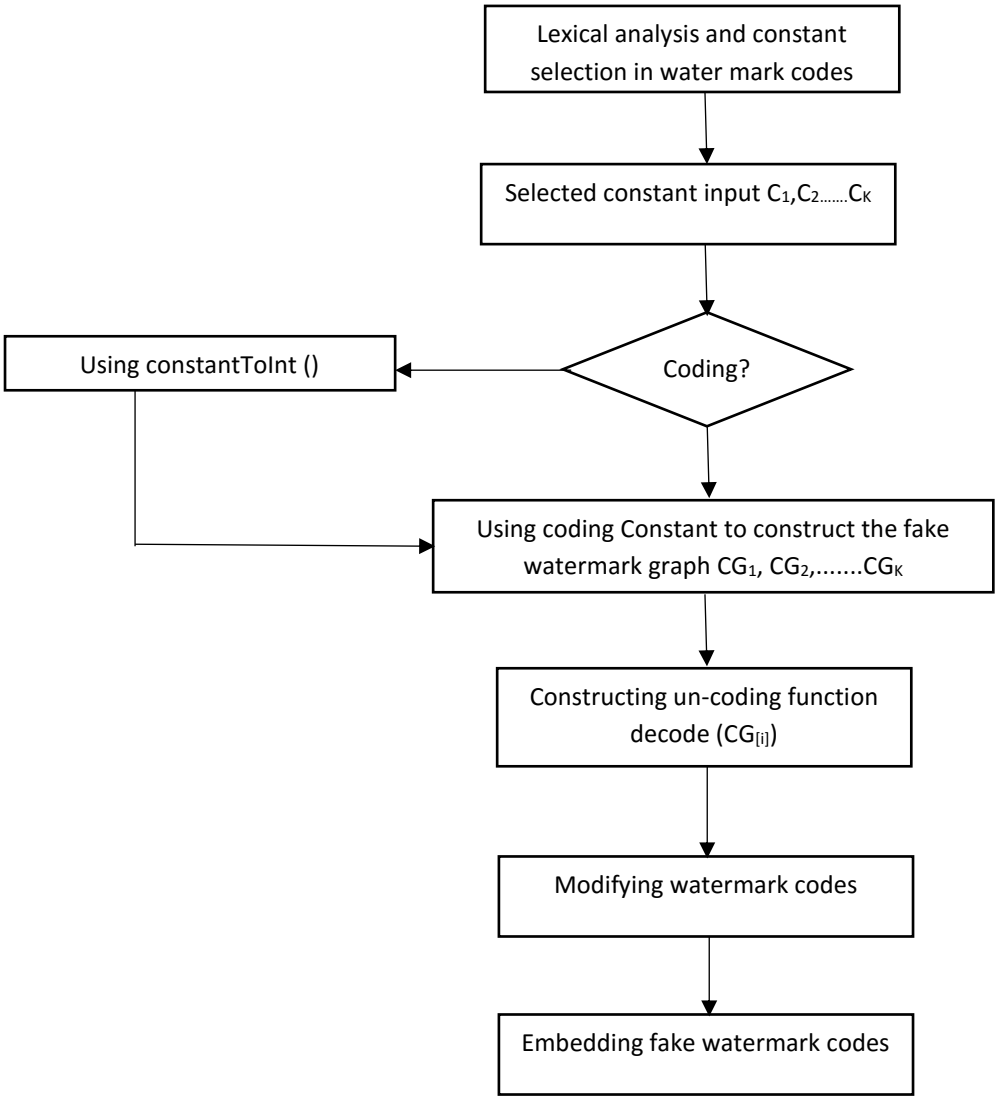

Figure 2.3 The Workflow of multi-constant DCW

2.2 Digital Watermarking Algorithms

Many algorithms have been proposed for digital watermarking algorithms for images [37-40], audio and video [36]. Before going to define the digital watermark the first question we need to ask with any watermarking or stenographic system [23 – 27][30] is that what form will the embedded message take? The most straight-forward approach would be to embed text strings into an image, allowing an image to directly carry information such as author, title, date etc. However, the drawback of this approach is that ASCII text in a way can be considered to be a form of LZW compression, with each letter being represented with a certain pattern of bits. By compressing the watermark-object before insertion, robustness suffers.

Due to the nature of ASCII codes, a single bit error because of an attack can entirely change the meaning of that character and thus the message. It would be quite easy for even a simple task such as JPEG compression to reduce a copyright string to a random collection of characters. Rather than characters, why not embed the information in an already highly redundant form, such as a raster image? Not only do images lend themselves to image watermarking applications but the properties of the HVS can easily be exploited in recognition of a degraded watermark in **Fig. 2.4**:

Figure 2.4: Ideal Watermark-Object vs Object with 25% Additive Gaussian Noise

2.2.1 Least Significant Bit Modification

The most straight-forward method of watermark embedding would be to embed the watermark into the least-significant-bits of the cover object [35]. Given the extraordinarily high channel capacity of using the entire cover for transmission in this method, a smaller object may be embedded multiple times. Even if most of these are lost due to attacks, a single surviving watermark would be considered a success.

LSB substitution however despite its simplicity brings a host of drawbacks. Although it may survive transformations such as cropping, any addition of noise or lossy compression is likely to defeat the watermark. An even better attack would be to simply set the LSB bits of each pixel to one fully defeating the watermark with negligible impact on the cover object. Furthermore, once the algorithm is discovered, the embedded watermark could be easily modified by an intermediate party.

An improvement on basic LSB substitution would be to use a pseudo-random number generator to determine the pixels to be used for embedding based on a given “seed” or key [37]. Security of the watermark would be improved as the watermark could no longer be easily viewed by intermediate parties. The algorithm however would still be vulnerable to replacing the LSB‟s with a constant. Even in locations that were not used for watermarking bits, the impact of the substitution on the cover image would be negligible. LSB modification proves to be a simple and fairly powerful tool for

stenography, however lacks the basic robustness that watermarking applications require.

2.2.2 Correlation-Based Techniques

Another technique for watermark embedding is to exploit the correlation properties of additive pseudo-random noise patterns as applied to an image [39]. A pseudo-random noise (PN) pattern $W(x, y)$ is added to the cover image $I(x, y)$, according to the Eqn. 2.1 given below.

$$I_W(x, y) = I(x, y) + k*W(x, y) \quad (2.1)$$

In Eqn. (2.1), k denotes a gain factor and IW the resulting watermarked image. Increasing k increases the robustness of the watermark at the expense of the quality of the watermarked image.

To retrieve the watermark, the same pseudo-random noise generator algorithm is seeded with the same key and the correlation between the noise pattern and possibly watermarked image computed. If the correlation exceeds a certain threshold T, the watermark is detected and a single bit is set. This method can easily be extended to a multiple-bit watermark by dividing the image up into blocks and performing the above procedure independently on each block.

This basic algorithm can be improved in a number of ways. First, the notion of a threshold being used for determining a logical "1" or "0" can be eliminated by using two separate pseudo-random noise patterns. One pattern is designated a logical "1" and the other a "0". The above procedure is then performed once for each pattern and the pattern with the higher resulting correlation is used. This increases the probability of a correct detection, even after the image has been subject to attack [44].

2.2.3 Frequency Domain Techniques

An advantage of the spatial techniques discussed above is that they can be easily applied to any image; regardless of subsequent processing (whether they survive this processing however is a different matter entirely). A possible disadvantage of spatial techniques is that they do not allow for the exploitation of this subsequent processing in order to increase the robustness of the watermark.

In addition to this, adaptive watermarking techniques are a bit more difficult in the spatial domain. Both the robustness and quality of the watermark could be improved if the properties of the cover image could similarly be exploited. For instance, it is generally preferable to hide watermarking information in noisy regions and edges of images, rather than in smoother regions. The benefit is twofold; Degradation in smoother regions of an image is more noticeable to the HVS and becomes a prime target for lossy compression schemes.

Taking these aspects into consideration, working in a frequency domain of some sort becomes very attractive. The classic and still most popular domain for image processing is that of the Discrete-Cosine-Transform (DCT) [46]. The DCT allows an image to be broken up into different frequency bands [35], making it much easier to embed watermarking information into the middle frequency bands of an image. The middle frequency bands are chosen in such a way so that they have minimize to avoid the most visual important parts of the image i.e low frequencies without overexposing themselves to removal through compression and noise attacks (high frequencies) [37].

2.2.4 Wavelet Watermarking Techniques

Another possible domain for watermark embedding is that of the wavelet domain [36]. The Discrete Wavelet Transform (DWT) separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components. The process can then be repeated to computes multiple "scale" wavelet decomposition.

2.2.5 Singular Value Decomposition

Although SVD works for any N×M matrix I, but without loss of generality [101], our discussion will be limited for the N×N matrix. The SVD of the N×N matrix I is

$$I=USV^T \qquad (2.2)$$

Where U and V $\in R^{NxN}$ are unitary and S $\in R^{NxN}$ is a diagonal matrix and the superscript T denotes matrix transposition. The diagonal elements of S, denoted by σ_i^S are called the singular values of I and are assumed to be arranged in decreasing order

$\sigma_i > \sigma_{i+1}$. The columns of U denoted by U_i's are called the left singular vectors while the columns of V denoted by V_i's are called the right singular vectors of I. It is easy to see that σ_i, V_i and U_i satisfy:

$$IV_i = \sigma_i U_i \quad (2.3)$$

$$U_i^t I = \sigma_i V_i^t \quad (2.4)$$

2.2.6 Singular Value Decomposition (SVD) -Based Watermarking

Liu and Tan [15] proposed an SVD-based watermarking scheme for rightful ownership protection. Without loss of generality, I and W are assumed to be N×N square matrices. Their algorithm consists of the following three steps:

- Perform SVD on the original un-watermarked image

$$A = \bigcup \Sigma V^T \quad (2.5)$$

- Add the watermark image W to S and obtain the reference watermark Sn as

$$S_n = S + \beta W \quad (2.6)$$

Then perform SVD on the reference watermark Sn as

$$S_n = S + \beta W = U_W S_W V_W^T \quad (2.7)$$

- Obtain the watermarked image I_W as

$$I_W = U S_W V^T \quad (2.8)$$

Here, β is a scale factor that controls the strength (energy) of the embedded watermark.

To extract the watermark from a possibly distorted watermarked image I_W*, their algorithm proceeds as follows:

- Perform SVD on the possibly distorted watermarked image I_W* as

$$I^*_W = U^* S^* V^{*T} \quad (2.9)$$

- Use U_W, V_W as obtained from Eqn. (7) to obtain

$$S^* = U_W S^*_W V_W^T \quad (2.10)$$

- Get the possibly distorted watermark W* as

$$W^* = \frac{1}{\beta}(S^* - S) \quad (2.11)$$

This algorithm requires U_W, S and V_W to be available for detection. C. Collberg, G.R. Myles and A. Huntwork, [43] have shown that this algorithm is fundamentally flawed. This is because it only embeds the diagonal matrix S_W. The detection algorithm simply extracts a possibly distorted diagonal matrix S^*_W. After that, the detection algorithm utilizes (does not extract) the singular vectors of the reference watermark (U_W and V_W). **"**Comments on an SVD-based watermarking scheme for protecting rightful ownership" [39] have shown that, by using the reference watermark SVD pair (U_W, V_W) in the detection stage, false-positive detection will have a probability of one. In other words, using the singular vectors of any fake watermark in the detection stage, one can always claim that this watermark was the embedded one. Hence, he can claim ownership of the watermarked image. Ahmad A. Mohammad, Ali Alhaj, Sameer Shaltaf proposed a variation of [38]. We propose a variation on Mohammad, Alhaj & Shaltaf by embedding text also [22]. As opposed to their algorithm, the proposed algorithm overcomes the problem of false positive detection. Also if first watermark degraded or destroyed due to some reason we can detect second one. In addition, the proposed algorithm is robust and noninvertible.

2.2.7 Other

Xuesong Zhang, Fengling He, Wanli Zuo [3] described the watermarking techniques based on hash functions by using special hash function in which the watermark is embedded. Let the watermark W to be embedded is mapped into a large integer at first and divided into many small parts w_1, w_2, …. w_n, based on these parts, a hash function is constructed in the following form:

$H(x) = w_i$ where i = 1, 2, ..n

The hash function H(x) is embedded into the program being protected. Some of the shortcomings of this method defined by authors are as follows:

- Here the hash function is equivalent to a lookup table. Hash function return values are stored in the code section of the software directly.

- The hash function is not called in the software. It is a part of the run-time code.

C. Collberg, G.R. Myles and A. Huntwork, [43] have described the architecture of a fully automated evaluation tool for digital a watermarking scheme which is the logical continuation of the early benchmark. They explained the new benchmark which is operated on a piece of code that is provided by the user through a library and uses an object-oriented language to make multimedia handling quite simple. It also relies on pre-defined evaluation profiles (configuration files), allowing testing of different types of watermarking schemes automatically to different levels of assurance.

C. Collberg, S. Jha, D. Tomko and H. Wang [45] have presented a secure watermark verification scheme based on zero knowledge protocol and public-key encryption scheme. In this paper they explained copyright proving without revealing any information to remove the watermark. This leads to a significant improvement of watermark verification scheme in terms of security and validity.

Wenfa Qi, Xiaolong Li, Bin Yang [33] and Sion R., Atallah M.J., and Prabhakar S [31] have presented an approach for Chinese/English character segmentation and language discrimination. Their method has high performance and consists of server procedures based on the statistical characteristic data through multiple phases, without depending on the character recognition results at all. Especially it can deal with touching or overlapping of Chinese characters themselves or their components under much noises, even with additional long lines. The main advantage of this method is that it is less time consuming. Their method can be effectively used in pre-process of text watermarking scheme to get correct and synchronous segmented results. This method can even be implemented effectively with hypothesis that there is no change of the font size in one single text line. If it wants to segment italic Chinese character correctly, the

italic-direction projection should be conducted. It can‟t deal with completely touched English characters.

Wenfa Qi, Xiaolong Li and Bin Yang [33] have presented the scheme for watermarking Scalable Vector Graphic (SVG) graph data, which is independent to the watermark embedding algorithm. In this paper the authors explained watermarked SVG data that may be encrypted or signature furthermore so as to strengthen the security of storage and exchanging. The authors also explained future work which is focused on developing watermark embedding program component or environment and novel robust embedding algorithm for polygonal line and parameter curve.

Lee, S. and Jung, S. [2] have presented the scheme based on the software birthmark presented in [10], They have introduce FnGS as the birthmark and proposed software zero-watermarking by incorporating the idea of Shamir‟s secret sharing and image zero-watermarking which is a higher validity, credibility and resilience from experiments. Their embedding and extraction procedure are shown below in **Fig. 2.6** and **Fig. 2.7** respectively.

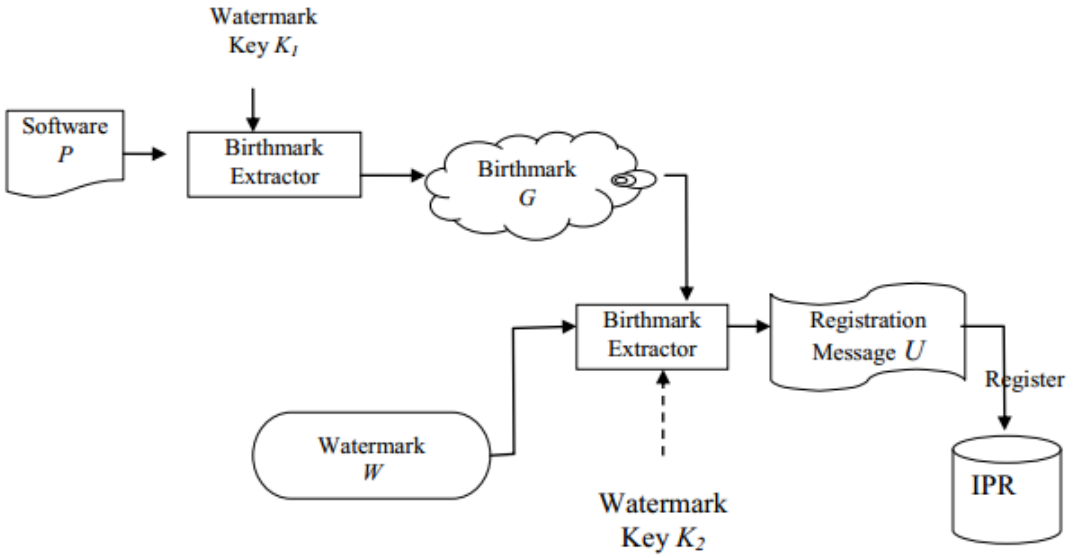

Figure 2.5: The Embedding Procedure of Zero Watermarking

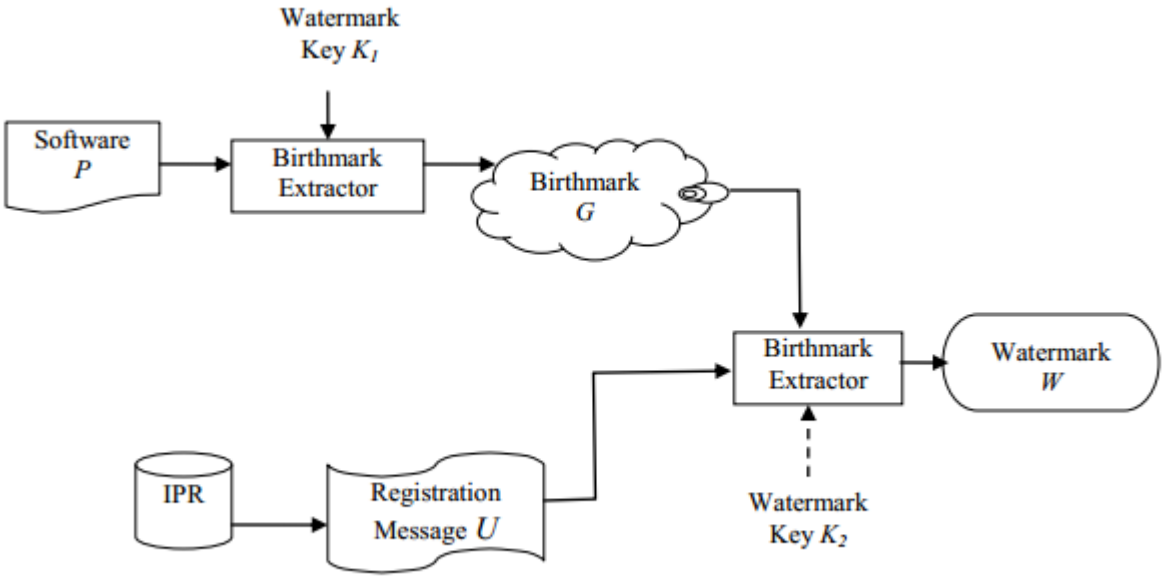

Figure 2.6: The Extraction Procedure of Zero Watermarking

Y. Hao, G. Bao and H. Zhang [49] have presented a framework to improve the security of traditional watermarking schemes watermark detection process. Since most of the implementations of TPM is constructed by hardware, so the operations in TPM are secure and cannot be tampered with anyone including the owner of TPM and the data stored in TPM protected locations are unseen by anyone except for TCG protected capabilities. This method binds the secret message with the special program to ensure that not only the watermark detection can be public, but also the secure sensitivity messages are protected as well [26]. Kang Hui, Liu Jing, Zhu Xiao-dong and Zhang Xiao-xu [48] have proposed a scheme based on the fingerprint watermark and tried to introduce biometrics in the watermark system. They wish to integrate the digital watermarking technology with the fingerprint identification technology. The scheme depends upon the spatial domain, DCT domain of multi-bits embedded watermark methods to embed and extract the information of the fingerprint characteristic and it is better to extract the simplicity in the robustness and embedded method. They have also shown that the scheme is feasible and effective.

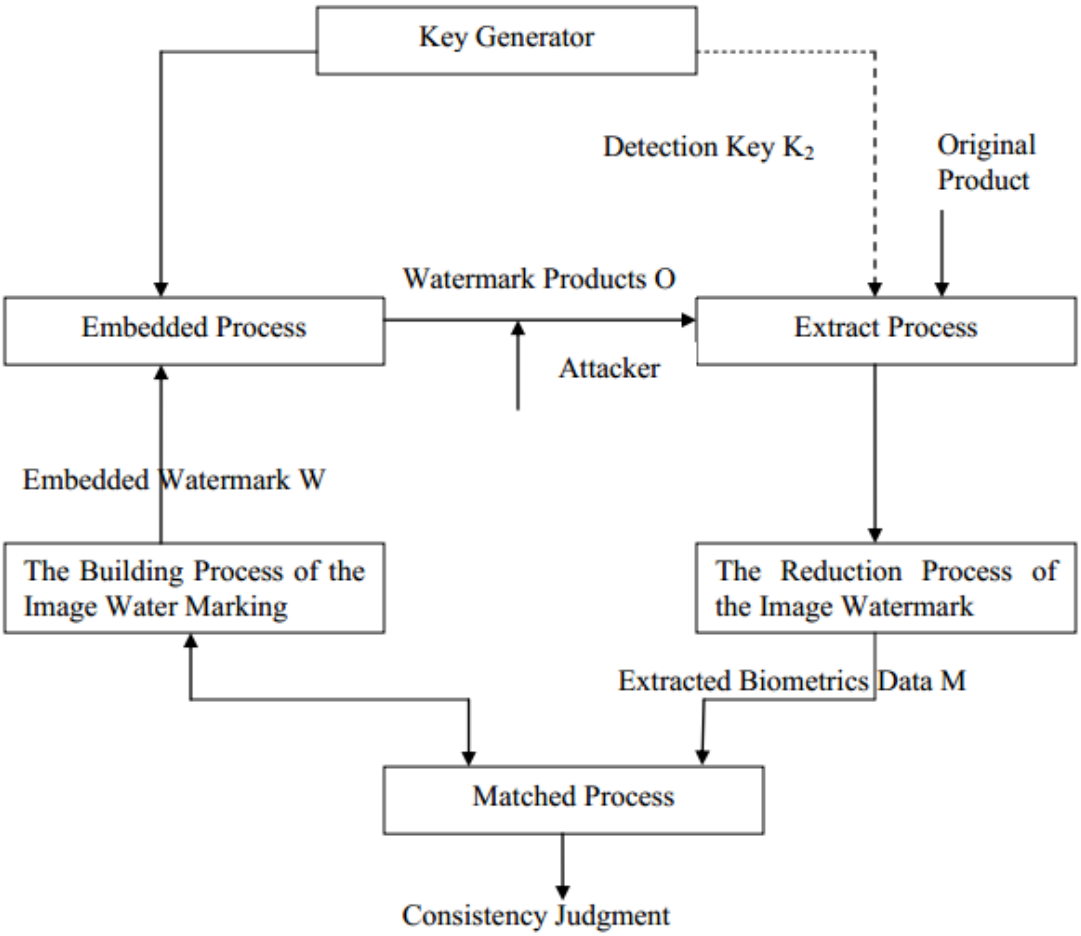

Figure 2.7: System Model Based on the biological characteristics of the digital watermarking

Hang Su, Chuqing Lv, Yanbing Ji and Yulin Wang [47] have proposed a new mechanism of watermarking based on the combination of rotation method and water-marking technology to fulfil the digital-image-bill authentication. This watermarking method is a modified block-based scheme for DCT-domain. They have also shown that this watermark is vulnerable for geometrical attacks, such as cropping, cutting, compression, rotation and embedded data with intra-block relations rather than inner-block relations.

2.3 Watermarking Algorithm for Relational Database

The relational database watermarking techniques consist of watermark embedding and watermark verification. During watermark embedding process, a private key K is used to embed the watermark W into the original relational database. The watermarked database is then made publicly available. To verify the ownership of a suspicious relational database, the verification process is performed where the suspicious database is taken as input and by using the private key K (the same which is used during the embedding) the embedded watermark (if present) is

extracted and compared with the original watermark information. **Fig. 2.9** depicts the basic relational database watermarking technique.

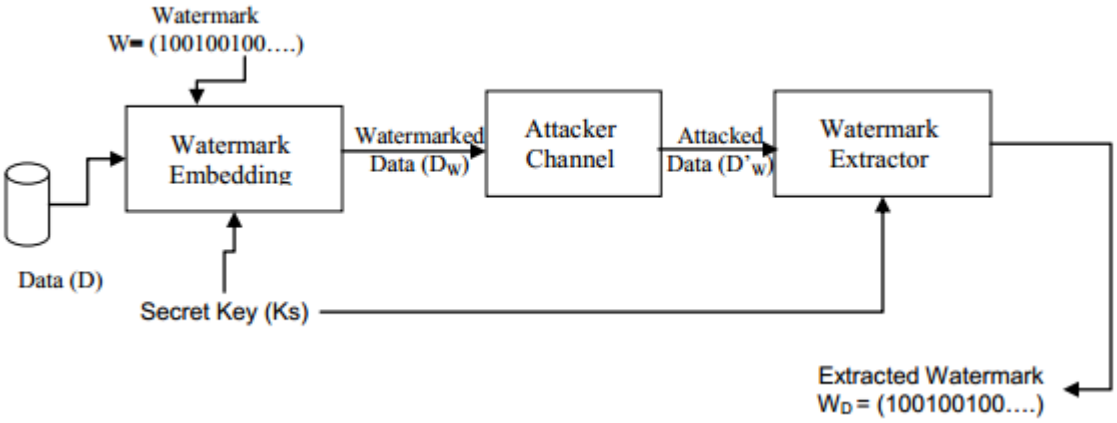

Figure 2.8: Basic Watermarking Technique for RDBMS

The relational data defers from multimedia data in the following ways:

2.3.1 Few Redundant Data

In this case multimedia objects consists of large number of bits providing large cover to hide watermark, whereas the database object is a collection of independent objects, called tuples. The watermark has to be embedded into these tuples.

2.3.2 Out-of-Order Relational Data

In this case the relative spatial/temporal positions of different parts or components in multimedia objects do not change, whereas there is no ordering among the tuples in database relations as the collection of tuples is considered as set.

2.3.3 Frequent Updating

In this case any portion of multimedia objects is not dropped or replaced normally, whereas tuples may be inserted, deleted, or updated during normal database operations.

2.3.4 Other

There are many psycho-physical phenomena based on human visual system and human auditory system which can be exploited for mark embedding. However, one cannot exploit such phenomena in case of relational databases.

Due to these differences between relational and multimedia data, there exists no image or audio watermarking method which is suitable for watermarking of relational databases. These differences give rise to many technical challenges in database watermarking as well.

2.4 Evaluation Criteria of Software Watermark

In software watermarking people generally use four criteria for evaluating their quality [38]. The criterion of data rate involves the ratio of the size of the watermark to that of the watermarked program. Resilience is the ability to resist against semantics-preserving translations, whereas stealth concerns lack of statistically distinct visible features between the un watermarked and the watermarked program. Finally, performance criterion is the ratio of the size and the execution time of the watermarked program to that of the original program.

2.5 Research Platforms for Software Watermarking

The following four systems dominate the research platforms for software watermarking:

JavaWiz [44], Hydan [43][46], UWStego [45] and SandMark [43].

JavaWiz is a software watermarking system developed at Purdue University. It can watermark Java source programs and it is written entirely in Java. This system has implemented the CT algorithm.

Hydan is a software watermarking system developed at Columbia University. It is used to watermark an executable.

UWStego is software watermarking research tool developed at the University of Wisconsin for experimenting and testing various software watermarking techniques and has a toolset for developing new software watermarking algorithms.

Lastly,

SandMark is a comprehensive research tool for software watermarking and obfuscation developed at the University of Arizona. It can be used to measure the effectiveness of software watermarking algorithms. Like JavaWiz, this platform is also written in Java.

Chapter 3

Proposed Work

In the rich body of literature on watermarking multimedia data, most of the techniques were initially developed for still images [26] and later extended to video and audio sources [27]. These methods do not apply in the context of relational data because an important parameter in their operating lies in the fact that multimedia and software objects are of value only when they are entire: it is not possible to maintain the usefulness of the objects if parts are arbitrarily removed from them or added to them. In the case of databases, the insertions, deletions and updates of tuples constitute the more familiar processes in the framework of their operation. Also, in a database relation every tuple is a separate object (entity) and should be protected independently.

The fundamental objective of watermarking methods for relational data is to deliver efficient performance with respect to the following important metrics:

- The storage cost for the maintenance of the secret keys and other useful information (if any) that are required to maintain secrecy for the detection phase,
- The time cost required to embed the watermark and detect it, subsequently, in a suspicious relation,
- The ability of a relation to remain watermarked after modification operations (insertions, updates and deletions of database tuples), and,
- The sensitivity of the method to malicious attacks.

Perhaps the most well-known robust watermarking scheme for relational data is the one proposed in [28] whereby a small portion of numeric data is changed according to a secret key in such a way that this change can be detected for the purpose of ownership proof. Since the method just embeds a meaningless watermark, so that it can only determine whether the database is indeed watermarked, it cannot be used for meaningful embedding information. Another drawback is that a very high or very low percentage of marks has to be detected in a suspicious database to verify ownership, otherwise the method cannot decide whether the “unlike” watermark is a result of an

attack or because no certain watermarks exists. This work has been extended in [29, 30] to allow meaningful multiple-bit watermarks to be embedded as well.

Another popular robust multibit watermark scheme for numeric data is proposed in [31] in which the tuples are securely divided into nonintersecting subsets. A single watermark bit is embedded into each subset, by modifying the distribution of tuple values. However, the capacity of the watermark is limited and the method has to record an extra subset classifying information, which is much larger than the size of the watermark, and safe storage, as well as space needed, are at question. Also, the scheme is not suitable for database relations that need frequent updates, since frequent data modifications may destroy the watermark and it is very expensive for the watermarking method to re-watermark modified database relations. Reference [32] extended this work, making it resistant to modifications and alteration attacks, however the subset information that needs to be stored and be given as input to the watermark detection process remains high.

In our approach we take a database D_B and produce a Watermarked database W_{DB}. In this for each tuple $r \in R$ encrypt the primary key P_K using AES as

$$E_{PK} = \text{Encrypt}\ (P_K, K) \tag{3.1}$$

After that select 3 bit from E_{PK} to choose one of the m attribute same thing done for other attribute then select 8 least significant bit of both attributes and derive a number from it and name it X and Y, if difference of X and Y is between 80 to 170 then tuple is suitable for watermarking else not then select another attribute if that tuple is suitable, make last bit of this attribute to 1. Select another bit from above attribute and use as watermark.

- For each tuple $r \in R$
  - $E_{PK} = \text{Encrypt}\ (P_K, K)$
  - Select three bit from E_{PK} to choose one of the m attribute
  - Select another three bit from E_{PK} to choose one of the m-1(except above) attribute
- Select 8 least significant bit of both attributes and derive a number from it as X and Y

- If difference of X, Y is between 80 to 170 then tuple is suitable for watermarking else not.
- Select another attribute if that tuple is suitable make last bit of this attribute to 1 and hide in X and Y and make it X' and Y'
- With the help of X' and Y' we can reverse the watermark.

Our one condition is select suitable tuple in between 80 to 170 and other is not suitable so we can check it for different value as shown in table below

Mod	Suitable Tuple	Not Suitable Tuple
5	**1438**	**362**
8	**924**	**326**
10	**764**	**236**
15	**493**	**173**
20	**372**	**128**

Figure 3.1: Selected Tuple

Chapter 4

Result

In this we perform various attack on our database and we get the result as shown in figure below. In fig 4.1 we perform insertion attack

No. of rows inserted during attack	Total no. of rows	No. of violations detected	Violation Percentage (%)
1000	11000	520	4.72
2000	12000	990	8.25
3000	13000	1561	12.00
4000	14000	1973	14.09
5000	15000	2564	17.09

Figure 4.1: Insertion Attack

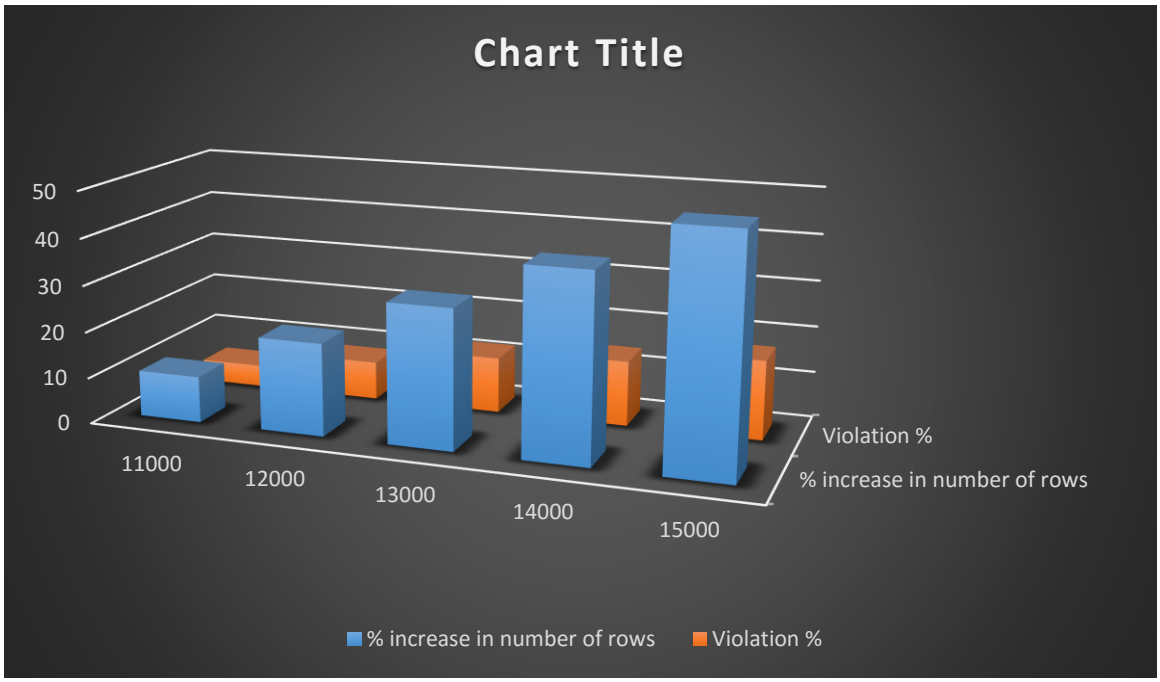

Figure 4.2: Graphical representation of insertion attack

In the Fig 4.3 shows the updation attack on database

No. of rows updated during attack	No. of violations detected	Violation Percentage (%)
1000	61	.61
2000	88	.88
3000	173	1.73
4000	232	2.32
5000	299	2.99

Figure 4.3: Updation attack on 1 column of database

Figure 4.4: Graphical Representation of updation on 1 column

No. of rows updated during attack	No. of violations detected	Violation Percentage (%)
1000	96	.96
2000	116	1.16
3000	205	2.05
4000	289	2.89
5000	343	3.43

Figure 4.5: Updation attack on 2 column of database

Figure 4.6: Graphical Representation of updation on 2 column

No. of rows updated during attack	No. of violations detected	Violation Percentage (%)
1000	103	1.03
2000	126	1.26
3000	215	2.15
4000	313	3.13
5000	359	3.59

Figure 4.7: Updation attack on 3 column of database

Figure 4.8: Graphical Representation of updation on 3 column

No. of rows updated during attack	No. of violations detected	Violation Percentage (%)
1000	116	1.16
2000	153	1.53
3000	252	2.52
4000	329	3.29
5000	373	3.73

Figure 4.9: Updation attack on 4 column of database

Figure 4.10: Graphical Representation of updation on 4 column

No. of rows updated during attack	No. of violations detected	Violation Percentage (%)
1000	121	1.21
2000	183	1.83
3000	296	2.96
4000	403	4.03
5000	446	4.46

Figure 4.11: Updation attack on 5 column of database

Figure 4.12: Graphical Representation of updation on 5 column

Chapter 5

Conclusion

Irreversible watermarking techniques make changes in the data to such an extent that data quality gets compromised. Reversible watermarking techniques are used to cater to such scenarios because they are able to recover original data from watermarked data and ensure data quality to some extent. However, these techniques are not robust against malicious attacks particularly those techniques that target some selected tuples for watermarking. In this paper, a novel robust and reversible technique for watermarking numerical data of relational databases is presented. The main contribution of this work is that it allows recovery of a large portion of the data even after being subjected to malicious attacks. The results of the experimental study show that, even if an intruder deletes, adds or alters up to 50 percent of tuples, our approach is able to recover both the embedded watermark and the original data.

REFERENCES

[1] J. A. Bloom, I. J. Cox, T. Kalker, J.Linnartz, and M. L. Miller. “Copy protection for dvd video”. Proceedings of the IEEE, 87(7):1267-1276, July 1999.

[2] Lee, S. and Jung, S. (2001)."A survey of watermarking techniques applied to multimedia” In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE ’01)*, pages 272–277, Pusan, South Korea. IEEE Press.

[3] Potdar, V. M., Han, S., and Chang, E. (2005)" A survey of digital image watermarking techniques". In *Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN ’05)*, pages 709–716, Peth, Australia. IEEE Press.

[4] Abdel-Hamid, A. T., Tahar, S., and Aboulhamid, E. M. (2004). "A survey on ip watermarking techniques". *Design Automation for Embedded Systems*, 9(3):211–227.

[5] Hacigumus, H., Iyer, B., and Mehrotra, S. (2002)." Providing database as a service". In *Proceedings of the 18th International Conference on Data Engineering (ICDE ’02)*, pages 29–38, San Jose, California, USA. IEEE Computer Society.

[6] Agrawal, R. and Srikant, R. (2000). "Privacy-preserving data mining". *ACM SIGMOD Record*, 29(2):439–450.

[7] Hu, J. and Grefen, P. (2002)." Component based system framework for dynamic b2b interaction". In *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC ’02)*, pages 557–562, Oxford, England. IEEE Computer Society.

[8] Khanna, S. and Zane, F. (2000)." Watermarking maps: hiding information in structured data". In *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms (SODA ’00)*, pages 596–605, San Francisco, California, United States. Society for Industrial and Applied Mathematics.

[9] Agrawal, R. and Kiernan, J. (2002). "Watermarking relational databases". In *Proceedings of the 28th international conference on Very Large Data Bases (VLDB ’02)*, pages 155–166, Hong Kong, China. VLDB Endowment.

[10] J. A. Bloom, I. J. Cox and M. L. Miller. “Watermarking application and their properties”. In International Conference on Information Technology: Coding and Computing, pages 6-10, April 2000.

[11] F. Hartung and M. Kutter. “Multimedia watermarking technique”. Proceeding of the IEEE, 87(7): 1079-1107, July 1999.

[12] U. Kohl, J. Lotspiech, and M. A. Kaplan. “Safeguarding digital library content and users: Protecting documents rather than channels”. http://www.dlib.org/dlib/september97/ ibm/09]otspiech.html,1997.

[13] J. Lacy, S. Quackenbush, A. Reibman, and J. Snyder. ”Intellectual property protection and digital watermarking”. In Information Hiding, Second International Working Proceeding, volume 1525, pages 158-168, April 1998.

[14] G. C. Langelaar, I. Setyawaan, and R. L. Lagendijk. “Watermarking digital image and video data: A state-of-the-art overview.” IEEE Signal Processing Magazine, 17(5):20-46, September 2000.

[15] C-Y. Lin and S. F. Chang. “A robust image authentication algorithm surviving jpeg compression”. In SPIE: Storage and Retrieval Image/Video Databases, January 1998.

[16] C-Y. Lin and S. F. Chang. “A robust image authentication algorithm surviving jpeg compression”. In SPIE: Security and Watermarking of Multimedia Contents, January 1999.

[17] J. Linnartz, T. Kalker, and J. Haitsma. Detecting electronic watermarks in digital video. In Proc. IEEE ICASSP, pages 2071-2074, March 1999.

[18] C. I. Podilchuk and E. J. Delp. “Digital watermarking: Algorithm and Application”. IEEE Signal Processing Magazine, 18(4):33-46, July 2001.

[19] C. I. Podilchuk and W. Zeng. “Image-adaptive watermarking using visual models.” IEEE Journal on Selected Areas in Communications, 16(4):525-539, May 1998.

[20] D. L. Robie and R. M. Merserean. ”Video error correction using steganography.” In Proc. IEEE ICIP, pages 207-226, October 2001.

[21] H. Stone. “Analysis of attacks on image watermarks with randomized coefficients.” NEC Technical Report, 1996.

[22] C. De Vleeschouwer, J. F. Delaigle, and B. Macq. “Invisibility and application functionalities in perceptual watermarking- an overview.” Proceeding of the IEEE,90(1):64-77,January 2002.

[23] S. Voloshynoviskiy and S. Pereira and I. T. Pun. “Attack modelling: Towards a second generation watermarking benchmark. Signal Processing, 81:1177-1212,2001.

[24] R. B. Wolfgang and E. J. Delp. “Fragile Watermarking using the vw2d watermark.” In Proc. Electonic Imaging 99, pages 204-213, January 1999.

[25] Ibrahim Kamel and Qutaiba Albluwi, “A robust software watermarking for copyright protection”, Computers & Security, 28(6), Publisher Elsevier Ltd, pp. 395 - 409, 2009.

[26] Langelaar G.C., Setyawan I., and Lagendijk R.L.: "Watermarking digital image and video data: a state-of-the-art overview". IEEE Signal Processing Magazine, Vol.17, pp.20-46, 2000.

[27] Boney L., Tewfik A.H., and Hamdy K.N.: "Digital watermarks for audio signals". Proceedings of the International Conference on Multimedia Computing and Systems, pp.473-480, 1996.

[28] Agrawal R., Haas P.J., and Kiernan J.: "Watermarking relational data: framework, algorithms and analysis". The VLDB Journal, Vol. 12, pp 157-169, 2003.

[29] Li Y., Swarup V., and Jajodia S.: "Constructing a virtual primary key for fingerprinting relational data". Proceedings of the ACM Workshop on Digital Rights Management, pp. 133–141, 2003.

[30] Li Y., Guo H., and Wang S.: "A multiple-bits watermark for relational data". Proceedings of the Principle Advancements in Database Management Technologies, pp.1-22, 2010.

[31] Sion R., Atallah M.J., and Prabhakar S.: "Rights protection for relational data". IEEE Trans. Knowl. Data Eng. Vol.16,No.12,pp.1509-1525, 2004

[32] Shehab M., Bertino E., and Ghafoor A.: "Watermarking relational databases using optimization-based techniques". IEEE Trans. Knowl. Data Eng. (TKDE), Vol. 20, No.1, pp.116-129, 2008.

[33] Wenfa Qi, Xiaolong Li and Bin Yang, "A Character Segmentation Method without Character Verification", International Symposium on Intelligent Information Technology Application Workshops, pp. 581 -584, December 21 - 22, IITAW 2008. 130

[34] Jonathan Bailey, "Why Your Copyright Protection is Second Rate", Available: http://www.plagiarismtoday.com/2008/01/11/why-your-copyright-issecond-rate/

[35] Michael Heyward, "Changing book import rules will hurt Australian writers", Available: http://textpublishing.com.au/news/post/changingbook-import-rules-will-hurt-australian-writers

[36] Cedric Lam, Janet Wong and Grace Wong, "Protecting copyright online", Available: http://www.managingip.com/Article/2460316/Protectingcopyright-online.html

[37] "Berne Convention for the Protection of Literary and Artistic Works" Available: http://www.wipo.int/treaties/en/ip/berne/index.html

[38] "European copyright law", Available: http://eurlex.europa.eu/LexUriServ/LexUriServ.do ?uri=CELEX:32001L0029:EN:HTM L

[39] "Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS)", Available: http://www.wto.org/english/docs_e/legal_e/27- trips_03_e.htm

[40] "Anti-Counterfeiting Trade Agreement", Available: http://trade.ec.europa.eu/doclib/html/142039.htm

[41] "The World Intellectual Property Organization", Available: http://www.wipo.int/portal/index.html.en

[42] "International Intellectual Property Alliance", 2009, Special 301 Report, Available: http://www.iipa.com/special301.html

[43] C. Collberg, G.R. Myles and A. Huntwork, "SandMark - A tool for software protection research", IEEE Magazine of Security and Privacy, pp. 40-49, 2003.

[44] Jen Palsberg, Sowmya Krishnaswamy, Minseok Kwon, Di Ma, Qiuyun Shao and Yi Zhang, "Experience with Software Watermarking", Epstein J,et al., eds. Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000) [C], New Orleans: IEEE Computer Society Press, pp. 308-316, 2000.

[45] C. Collberg, S. Jha, D. Tomko and H. Wang, "Uwstego: A general architecture for software watermarking", Technical Report TR04–11, August 31, 2001.

[46] R. El-Khalil and A. Keromytis, "Hydan: Embedding secrets in program binaries", in http://www.andrew.cmu.edu/user/dgao/InfoHiding/binarystego.pdf, August 14, 2004.

[47] Hang Su, Chuqing Lv, Yanbing Ji and Yulin Wang, "A Watermarking Enote Technique against Geometric Attacks", 2nd International Conference on Mechanical and Electronics Engineering (ICMEE), pp. V1-446 - V1-449, August 1 – 3, 2010.

[48] Kang Hui, Liu Jing, Zhu Xiao-dong and Zhang Xiao-xu, "Study on implementation of a fingerprint watermark", International Conference on Computer Science and Software Engineering, pp. 425 – 428, December 12-14, 2008.

[49] Y. Hao, G. Bao and H. Zhang, "Secure Public Digital Watermarking Detection Scheme", Congress on Image and Signal Processing, CISP '08, pp. 725 – 729, May 27-30, 2008.