A
Major Project-II Report
On

CONTENT CHARACTERISTIC BASED ROBUST DATABASE WATERMARKING

Submitted in Partial Fulfilment of the Requirement for the Degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

RITESH GOEL
2K14/CSE/15
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 **"Content Characteristic Based Robust Database Watermarking"** submitted by **Ritesh Goel, Roll No. 2K14/CSE/15** 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 "**Content Characteristic Based Robust Database Watermarking**" 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.

Ritesh Goel
2K14/CSE/15

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.

Ritesh Goel
University Roll no: 2K14/CSE/15
M.Tech (Computer Science & Engineering)
Department of Computer Engineering
Delhi Technological University
Delhi – 110042

ABSTRACT

With the booming of internet, the availability of the web based services including database services has become easy and economical to all. Digital data such as text, relational database, audio, video and software are intellectual property of owners but there outsourcing through the internet brings different threats like coping, modifying and then redistributing them. With wide spread use of relational database in many real-life applications, maintaining integrity and providing copyright protection is gaining keen interest of the researchers of the relational database. To cope with these issues the watermarking techniques playing a vital role in the last decade. This thesis reports the state-of-art watermarking techniques, highlights the shortcomings of these techniques. Then the proposed solution is given for dealing with some of the limitations of the previously presented techniques. The field of relational database watermarking is gaining interest and its becoming mature time by time. But the work done on relational database, having numerical values is more than non-numeric, so the proposed technique deals with non-numeric relational database. Considering the limitation of the previously proposed techniques, the given technique does not select any of tuple, portion (subset) from the database but uniformly distribute the watermark or consider the database as a whole while watermark embedding process. So the chances of subset selection decrease automatically. The proposed system takes into consideration a sample database and applies watermark to its tuples by selecting a set of attributes based on random bit selection and relevance/dependency of attributes of database. This report is concluded by providing results of several experiments performed in support of the effectiveness of proposed technique.

Keyword: database, watermarked database, primary key.

List of Figures

List of Tables

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

1.1 Motivation

Due to the cost-free availability and efficiency of Internet, sharing digital data such as text, databases, images, videos and software has become very common around the world and with no time. Data may be outsourced by a researcher for educational, commercial or official purposes. On the other hand it can be quickly got and misused by another person. The attacker can make changes to that data and redistribute it across Internet pretending being the owner of that data. So protecting the data from such kind of misuse and claiming ownership of actual owner has become very difficult. It may affect the owner of digital products financially.

Protecting such kind of threats is a crucial issue all over the world and the owners of digital assets furious about the integrity of their data. So the researchers are mainly concerned about the finding security issues and solving them. Tackling the intellectual properties of data such as authentication, copyright protection, integrity, piracy, plagiarism, ownership claiming and confidentiality is a major problem. The creators of digital resources may not always be the owners. For example in an organization an employee creates some digital asset but the employer of that company owns it and he keep the authority of digital assets with himself. Also there may be multiple owners or creators. So it is a critical to prove ownership on such digital contents. To cope with this critical situation, the idea of digital watermarking was given in the early 1990s. By watermarking the actual owner of the data can be identify easily and efficiently.

Watermarking techniques have historically been used to ensure security in terms of ownership protection and tamper proofing for a wide variety of data formats. This includes images, audio, video natural language processing software, relational databases, and more. Reversible watermarking techniques can ensure data recovery along with ownership protection. Fingerprinting, data hashing, serial codes are some other techniques used for ownership protection. Fingerprints also called transactional watermarks are used to monitor and identify digital ownership by watermarking all the copies of contents with different watermarks for different recipients. Primarily this type

of digital watermarking tries to identify the source of data leakage by tracing a guilty agent. In hashing, digital contents can be saved by performing a one-way hash function whereby the data contents do not change. If the hash of the original and tampered data is the same, data authenticity can be verified but ownership cannot be proved easily. Serial or classification codes are used for filtering of inappropriate contents over the Internet and are mainly applicable to images, audio and video. Watermarking has the property that it can provide ownership protection over the digital content by marking the data with a watermark unique to the owner. The embedded watermark can subsequently be used for proving and claiming ownership.

Database systems are using now a days in many real life applications. Sometimes, authors need to outsource their databases across Internet. As the time goes, the need of preserving copyright protection is becoming crucial. Many techniques were suggested to assure to integrity of relational databases. The researchers in the database fields import the idea of watermarking for the copyright protection of their relational database. The techniques used for multimedia assets were not suitable for marking Relational databases. The idea was given for the first time in 2002 by Rakesh Agarwal et al. [1]. Watermarking multimedia objects successfully dealt with problem of piracy and copyright violation. As watermarking multimedia object is comparatively mature field and a lot of material is available in this area. Many techniques are available which is dealing with copyright issue of multimedia products efficiently. But the algorithms proposed for images, videos, texts, audio and software are not useful for watermarking relational database watermarking due to difference in nature of the database from other digital assets. Therefore, watermarking relational database was a big challenge for the researchers. After Rakesh Agarwal et al. [1] idea for watermarking relational database for the first time, many people gave their own algorithm for the preserving relational database integrity. Some suggested using an image as watermarks [2], some tried to introduce ‘fake’ tuples among the original tuples [6]. Our technique deal successfully reduced the chances of subset selection attack as for embedding the watermark we have considered the entire database.

So the motivation behind uniformly distribute33d watermark system is that current techniques handles different attacks but with little probability of survival against strong subset attacks. Proposed technique provides a better way of protecting watermarked database from subset attacks. Most of the algorithms presented by the authors are

distortion based, which introduced errors to the original data. In some application the maintenance of the quality is of high priority. As our technique doesn't actually insert *marks* in the data physical so it's distortion-based. The extraction of watermark does not need the actual table so its blind system.

1.2 Information Hiding

Before introducing directly our technique, providing a broad view of the Information Hiding is worth mentioning. Different Information hiding methods are used for Intellectual Property Right Protection (IRP) .Information Hiding is communication of information by concealing and retrieving from any digital media [14]. Information Hiding includes techniques like Cryptography, Steganography and Watermarking.

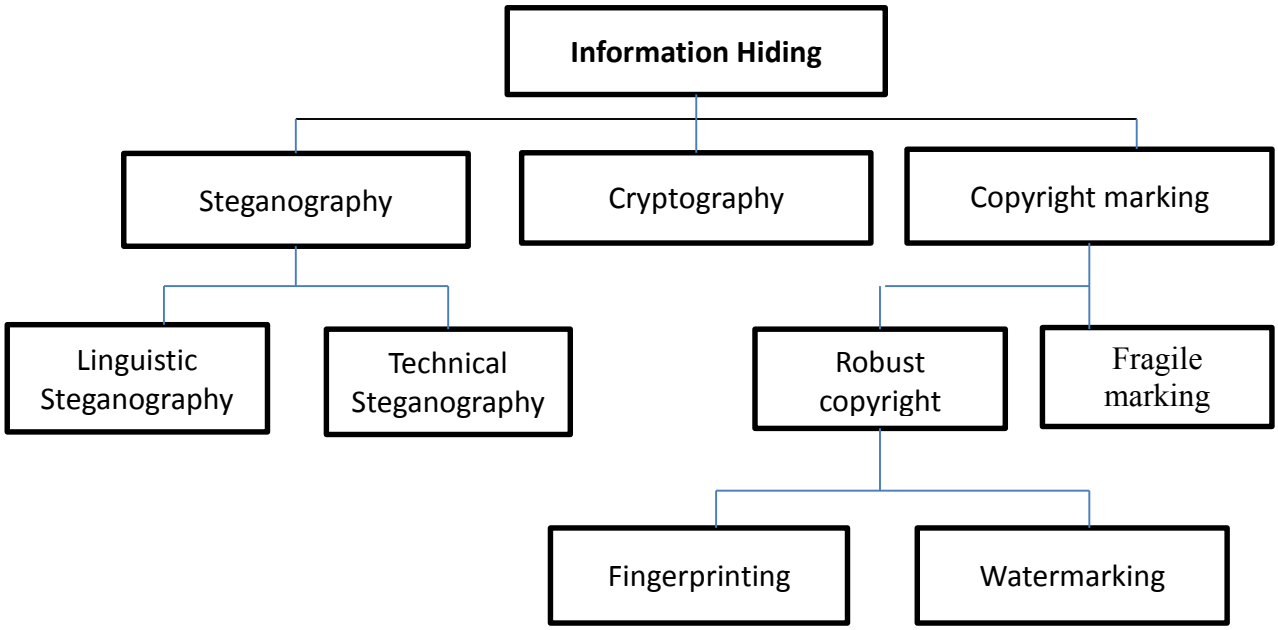


Figure 1-1 Overview of Information Hiding Techniques

1.2.1 Cryptography

Cryptography is an information hiding technique in which the contents (not the existence) of the information are concealed from the intruder. It is a two phase process i.e. encryption and decryption. In encryption phase, message is encoded at sender side and then passed through a communication line such that contents of the message cannot be readable for public. While in decryption phase, the encoded message is decoded or decrypted at the receiver end to the simple readable message again. For this purpose a secret key is used at both of the sender and receiver system.

If a message is encrypted by using an encryption secret key then its only can be decrypted by a person who has the decryption secret key. To prevent the message to be revealed by the data hackers, the communication line should be made secured and the encryption algorithm should be strong enough so that the eavesdropper cannot break it.

1.2.2 Steganography

Steganography is a derived from a Greek word "steganos" and "graphia", means "concealed" and "writing" respectively. In Steganography, the hidden information is placed secretly within apparently inoffensive carriers. Steganography hide the contents as well as the existence of the information. Unlike the cryptography, the steganography goal is to avoid drawing suspicion about the presence of secret information in the carrier message.

1.2.3 Digital Watermarking

Digital watermarking is a process in which secret copyright information ("marks") is inserted in the host content like text, image, database, video, audio and software in so that it is not possible for an intruder to change or remove it. Watermarking process consists of two processes, watermarking embedding process and watermarking insertion process. The watermark embedding process involves inserting the watermark using a secret key (known only to the owner of the database) into the data which is intended to outsource. When the same watermarked database is shared on the internet or network and if some kind of malicious attacks is suspected then the watermark insertion process is to be performed to proof ownership.

Digital watermarking of multimedia content is more commonly known. Particularly image watermarking—a derivative of Steganography is an age-old practice allowing covert transmission of messages from one party to another by exploiting redundancy in common image formats. However the basic process of multimedia watermarking is very different from that used to watermark relational databases because of a fundamental difference in the properties of the data. Multimedia data is highly correlated and continuous whereas relational data is independent and discrete. With the advent of modern copyright protection and information hiding techniques, database watermarking can be used to enforce ownership rights of relational data. However a major drawback of these techniques is that they modify the data to a very large extent

which often results in the loss of data quality. There is a strong need to preserve the data quality in watermarked data so that it is of sufficiently high quality and fit for use in decision making as well as in planning processes in different application domains. Data quality can be defined as the appropriateness of data for its intended applications.

1.3 Watermarking Relational Database

The increasing use of databases in many real life applications has brought the need of authentication and integrity of relational database systems. Watermarking image, text, video, audio and software inspired the researcher of copyright protection of relational databases and they brought the idea of watermarking for relational database. In this regard the first publication came in 2002 by Rakesh Agarwal et al. [1].

As the nature of Relational database is not same as that of the multimedia objects, so the algorithms proposed for multimedia so far cannot be used for watermarking relational database [1].

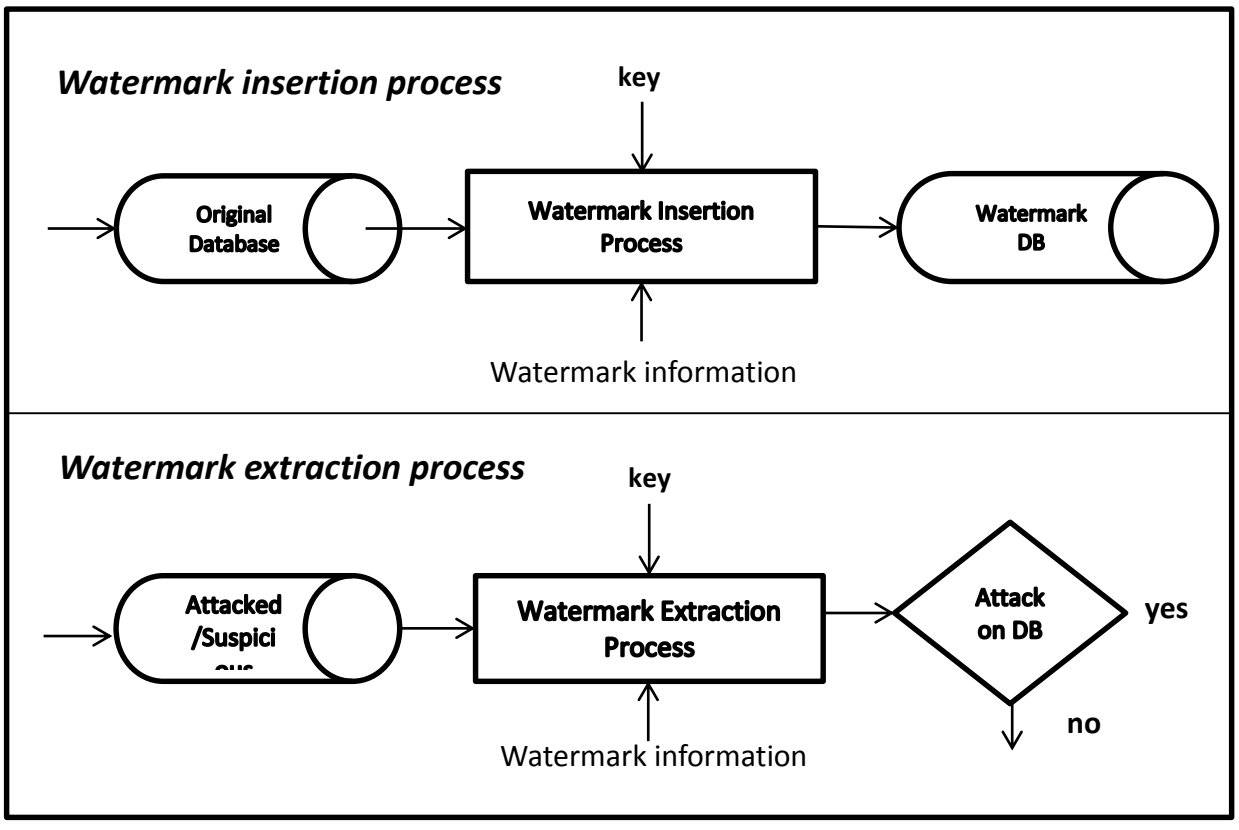


Figure 1-2 Basic Watermark System

1.3.1 Characteristics of Watermarking

A system of watermarking relational database should poses the desirable characteristics illustrated below.

1.3.1.1 Detectability

Watermark insertion is inserted in the original database but to proof ownership on data it should be possible for the watermark algorithm to extract the watermarks from the attacked database. Without the detectability property of the database it would not be possible to proof ownership on the data.

1.3.1.2 Blindness

A watermark system is called blind if at the time of extraction of watermarks there is no need of the original database and the watermarks which were provided for watermark insertion i.e. neither the un-watermarked database nor the watermarks are needed for the watermark extraction.

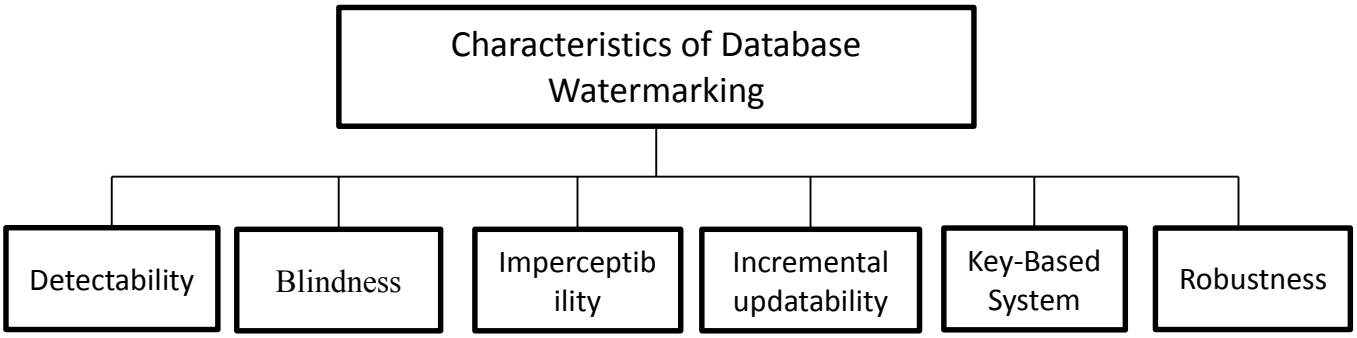


Figure 1-3 Characteristics of Database Watermarking

1.3.1.3 Robustness

The watermark should be strong enough against various kinds of malicious attacks such that an eavesdropper cannot change or delete the marks easily from the watermarked database.

1.3.1.4 Incremental Updatability

As the relational database is continuously updated i.e. new recodes can be added, the previous deleted or the existent one is changed continuously. The watermarking

system should be able to mark the updated portion without disturbing the previous marks. This property is very important in especially marking large databases.

If a watermarking technique rare this ability then the whole databases will have to be re-watermarked, which may result costly and time consuming.

1.3.1.5 Imperceptibility

In most of the cases we cannot compromise on the quality of the data and Its maintain the quality of the database. So there need an algorithm which does not introduce distortion to the database.

1.3.1.6 Key-Based System

The watermarking system should use such methods in which there is a private key known only to the owner of the database and assumed that the method of inserting the mark should be public.

1.3.2 Types of Attacks on the Database

Agarwal et al. [1] described some common attacks against watermarking relational databases. According to him the database may experiences the following types of attacks.

1.3.2.1 Benign Updates

There are some cases when a person is not aware that the underlying database is watermarked. He may perform some update e.g. deletion, insertion, updation. This may either add some more watermarks or remove the already existed watermarks from the database which in turn affect the embedded watermark extraction process. Although these updates are performed unintentionally but still a watermarking technique should be robust to resist such kind of updates because it greatly affects the watermarked database.

1.3.2.2 Malicious Attacks

In most of the case the attacker is aware that the database has been secured by inserting the watermarks and attempts to alter the marks by the different kinds of malicious attacks. These attacks are intentionally done by the attacker.

1.3.2.2.1 Value Modification Attacks

a. Bit Attack

If the attacker tries to alter the marks within the data simply by changing one or more of its bits such kind of attack is called bit attack. For removing mark completely an attacker have to change all of the bits of the mark which may leads to error in the data. The success rate of this attack can be increase if he has some information about the bit positions. More the bits change in the data more the distortion will be produce in the data and hence the data will become of no use for the attacker.

b. Randomization Attack

In bit attack when values of bits are attempted to set by a random number then it is called Randomization attack. This include zero- out attack which change the value of bit position to zero. Another in this category is the bit flipping attack in which the attacker inverts the bit value.

c. Rounding Attack

In this type of attack the attacker tends to round the values of the attribute. The rounding attack may not be successful because the attacker has to guess exactly the number bit position involved in the marks.

1.3.2.2.2 Subset Attack:

Subset selection is a type of piracy attack in which the intruder attempts to temper a subset of database. It is of three types:

a. ***Subset Selection Attack*:**

In this type of attack, the attacker selects a subset from the watermarked database with a probability that the subset does not contain watermark information.

b. ***Subset Deletion Attack:***

In subset deletion attack, attacker deletes a subset from watermarked database with probability that the deleted subset contains greater amount of watermark information.

c. ***Subset Addition Attack:***

In subset addition attack, attacker adds his own subset to watermarked database, containing his watermark information thus distorting original watermark information.

1.3.2.2.3 Collusion Attack:

a. ***Mix-and-match Attack:***

Mix and match attack: In this attack, Mallory creates his own relation by taking disjoint tuples from multiple relations containing similar relations.

b. ***Majority Attack:***

The attacker takes many copies of the same database and find out matching patterns among them and finally he create his own database.

1.3.2.2.4 False Claim of Ownership

In this type of attack the attacker tries to misguide the court about the ownership of the data. He can do this by adding some more tuples and watermarks to already watermarked database.

a. ***Additive Attack***

An attacker may add his own marks to the database and claim ownership.

b. ***Invertible Attack***

In this type of attack the intruder attempts to discover the marks in the already watermarked database besides the original done by the owner of the database. If he successfully found any pattern of marks in the database so he can claim ownership.

1.3.2.2.5 Brute Force Attack

In this kind of attack the malicious attacker tries to guess the secret key.

1.4 Reversible watermarking

Reversible watermarking tries to overcome the problem of data quality degradation by allowing recovery of original data along with the embedded watermark information. This paper proposes one such reversible watermarking technique that keeps the data useful for knowledge discovery. Data modifications are allowed to such extent that the quality of the data before embedding watermark information and after extracting is acceptable for knowledge extraction process. Consequently, knowledge discovery becomes successful in decision support systems where high quality data recovery is imperative.

Reversible watermarking techniques are already available in literature; however, to the best of our knowledge, no work has been conducted on overcoming the problems of reversible watermarking techniques in the presence of malicious attacks. Achieving robustness (attack resilience) in the presence of reversibility (ability to recover the watermark and the original data) is a challenging task. These two features may be potentially conflicting because a reversible watermark string also makes it an easier target for attack. Therefore, we try to find the most appropriate watermark bandwidth that ensures maximum watermark robustness without significant loss of information that may result by watermarking. To this end, we model the bandwidth optimization as a constraints optimization problem.

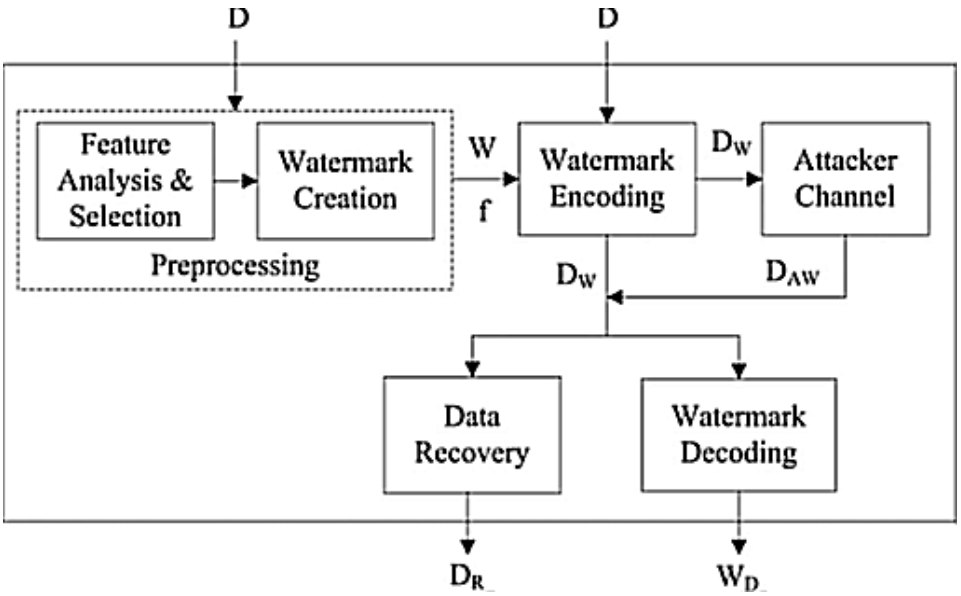


Figure 1.4 Main architecture of RRW

Constrained optimization (CO) allows one to optimize a single or multiple objectives with respect to certain variables that are bounded by some constraints. Our motivated problem is also a CO problem, whereby the research communities want to share databases over the public Internet or a cloud environment for their knowledge discovery processes. Ownership rights of these databases need to be protected from malicious recipients; in the presence of data quality constraint.

Recent research studies enunciate that computational intelligence techniques, such as genetic algorithm (GA) and particle swarm optimization (PSO) are a promising branch of evolutionary computation that model hard constrained optimization problems [17] using biological inspired computing algorithms. Digital watermarking can also be modelled as an

optimization problem as demonstrated by some recent research works [18] and [19] that use PSO for watermarking different data formats and the results are quite encouraging.

GA—an optimization algorithm is employed in the robust and reversible watermarking technique (RRW) proposed in this paper to achieve an optimal solution that is feasible for the problem at hand and does not violate the defined constraints.
An optimal watermark value is created through the GA and inserted into the selected feature of the relational database in such a way that the data quality remains intact.

In RRW, mutual information is used to select a suitable (candidate) feature from the database for watermarking. According to literature, existing reversible watermarking techniques, do not take into account the mutual information measure for determining relative importance of features. In RRW, the knowledge of mutual information for every candidate feature is also employed to compute the watermark information. Thus, it is ensured that the data quality will not be affected. Consequently, RRW provides a robust solution for data recovery that is reversible and resilient against heavy attacks.

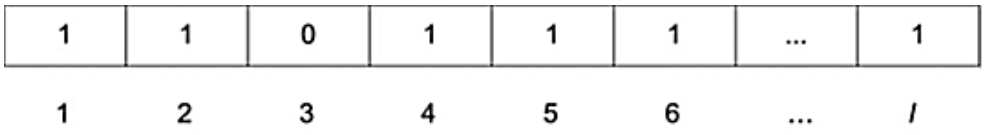
RRW mainly comprises a

(1) Data pre-processing phase,
(2) Watermark encoding phase,
(3) Attacker channel,
(4) Watermark decoding phase
(5) Data recovery phase.

In data pre-processing phase, secret parameters are defined and strategies are used to analyse and rank features to watermark. An optimum watermark string is created in this phase by employing GA—an optimization scheme—that ensures reversibility without data quality loss. In the watermark encoding phase, the watermark information is embedded in the selected feature(s). Two parameters, be the optimized value from the GA and hr a change matrix are used in the watermark encoding and decoding phases.

During watermark creation phase, we employed the following major steps of the GA for getting optimal watermark information:

- Initial random population of binary strings called chromosomes is generated. A Gene value of each chromosome represents l-bit watermark string as shown in Fig. below.
- Fitness of each chromosome is evaluated by employing a constrained optimized fitness function as discussed.
- Tournament selection mechanism is applied to get the most appropriate individuals as parent chromosomes.
- Genetic operations of crossover and mutation are performed on parent chromosomes to create off-springs. A single point crossover operator is applied to evolve high quality individuals, inheriting parental characteristics, by exchanging information between two or more chromosomes. A uniform mutation operator is applied to bring diversity in population through small random changes in gene values of binary chromosomes. The values of crossover fraction and mutation rate are set empirically.
- Elitism strategy is applied to hire two individuals with best fitness value; as elites to the next generation without genetic changes.
- Remaining population of the next generation is created by replacing less fit individuals of the previous generation with the fittest newly created off-springs.
- Steps 2 to 6 are repeated until MIO and MIW reach approximately equal values for a certain number of generations.
- Both, optimal watermark information string and best fitness value (b) is returned after the fulfilment of the termination criteria.



Finally, the watermarked data for intended recipients is generated. The attacker channel comprises subset alteration, subset deletion and subset insertion attacks generated by the adversary. These malicious attacks modify the original data and try to degrade its quality. In the watermark decoding phase the embedded watermark is decoded from the suspicious data. In order to achieve this, the pre-processing step is performed again, and decoding strategies (feature selection on the basis of MI, b the optimized value from the GA and hr the change matrix) are used to recover the watermark. Semi-blind nature of RRW is used mainly for data reversibility in case of heavy attacks (attacks that may target large number of tuples).

Original data is recovered in data recovery phase, through post processing steps for error correction and recovery.

The major contributions are:

- The design of an intelligent reversible watermarking technique for relational data that ensures data recovery without compromising data quality
- A robust data recovery scheme that is resilient against subset alteration, subset deletion and subset insertion attacks. RRW detects the watermark fully and recovers the original data.

In RRW, all the tuples of the selected feature can be marked thanks to the selection of a low distortion watermark; therefore, the attacker will have to attack all the tuples to corrupt the watermark to mitigate the effect of the majority voting scheme. Attacking all the tuples is not a viable option for the attacker because he has no knowledge of the original data or the usability constraints and that would completely compromise its usefulness.

Moreover, since RRW can afford to embed watermark bits in all or a large fraction of the tuples of the selected feature; it achieves high robustness against heavy attacks. However, marking all tuples is not a requirement. RRW is configurable in that the data owner can choose a fraction for watermarking if it is required. RRW outperforms existing state of the art reversible watermarking techniques including DEW, GADEW and PEEW. These techniques embed the watermark in partitions of the data to ensure minimum distortion; therefore, recover original data with degraded data quality and lack robustness. RRW has overcome drawbacks of these techniques and is also resilient against heavy attacks.
The robustness of RRW is evaluated through attack analysis, considering the attacker channel. It is worth mentioning that sound watermarking techniques exploit redundancy in the data to embed the watermark in a manner that it does not impact the overall size. For example consider the case of image watermarking that embeds information in the least significant bit (LSB) of pixel values which is imperceptible as well as it does not affect the overall image size. This is desirable because changes in the size of the original data can compromise the presence or absence of a watermark. Similarly one other contribution of our technique is that it avoids detection by keeping the size of the original relational data unaltered.

1.5 Symmetric key Algorithm

The shared secret key is the single key which is used in Encryption-Decryption process. This is the fundamental concept of symmetric key algorithm in cryptography. In symmetric key algorithm the same key will be used for encryption and decryption process. Here two type of symmetric algorithm, block and stream cipher. A block cipher used to encrypt the key into cipher text that has same size of image after encryption. For example we take the size to 1024x1024 bmp image or size of bmp image is 2.5 MB as an input for encryption than cross ponding output must be same size. Blowfish, Data Encryption Standards (DES), Triple DES, and IDEA are example of symmetric block cipher. The symmetric key algorithm uses a single key for encryption and decryption process.

1.6 Asymmetric Key Algorithm

The issue with secret keys is trading them over the Internet or a vast system while keeping them from falling into the wrong hands. Any individual who knows the secret key can decrypt the message. One answer is asymmetric encryption, in which there are two related keys - a key pair. A public key is made unreservedly accessible to any individual who may need to send you a message. A second, private key is kept secret, so that just you know it.

Any message (text, binary files, or documents) that are encrypted by utilizing people in public key must be decrypt by applying the same algorithm, yet by utilizing the coordinating private key. Any message that is encrypting by utilizing the private key must be decoded by utilizing the coordinating public key. This implies that you don't need to stress over ignoring public keys the Internet (the keys should be public). An issue with asymmetric encryption, in any case, is that it is slower than symmetric encryption. It requires significantly additionally handling energy to both encryption and decryption the substance of the message.

1.7 Thesis Structure

The thesis contains five chapters. Chapter 1 introduces watermarking, various types of watermarking, why do we need them etc. This chapter also explains the motivation of Research and focus on important goal, scope and objective of study. Here describe the symmetric key algorithm.

Chapter 2 describes the work that has been done in past. It explains database watermarking and fundamental of relational database necessary to know for watermarking a database. In this chapter also describe about the cryptography and some encryption algorithm like Advance encryption Standard, Data Encryption standard which have been used to calculate hash function.

Chapter 3 describes the proposed work of database watermarking with every step.

Chapter 4 thesis contains outputs of database watermarking. This chapter contains the simulation of attacks and analysis of result.

Chapter 5, describe the conclusion and future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Watermarking Relational database is comparatively young field as compare to Watermarking multimedia objects. The first literature came to the world in 2002 when the pioneer of the Database watermarking Rakesh Agarwal et al. [1] suggested that the Database copyright protection can be achieved by inserting hidden 'marks' into the Relational database.

This research though had some limitations but got the attention of the researchers all over the world. Since then the field have continued to flourish and many authors proposed their own ideas for securing the database from copyright threads using watermarking. Still Watermarking of Relational Database is comparatively new and challenging idea so the literature in this field is very limited. Different authors since have proposed different techniques on the basis of their own analysis. In this chapter we have reviewed the current state-of-art techniques presented to date.

2.2 Classification of Watermarking Techniques:

The state-of-art techniques for watermarking databases can be classification as under:

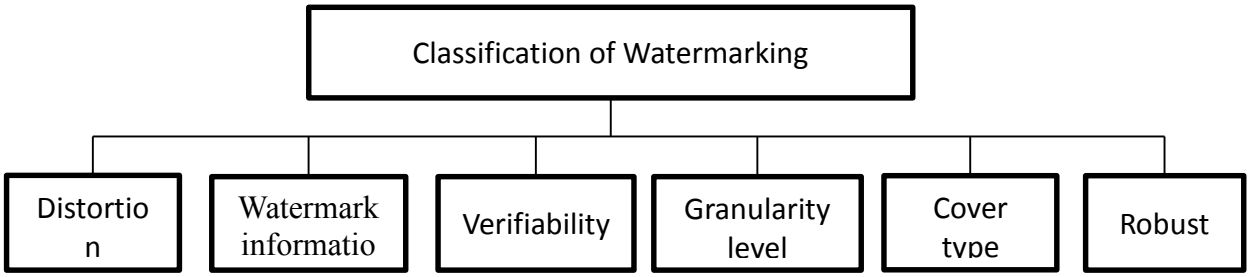


Figure 2-1 Classification of Relational Database Watermark

2.2.1 Distortion

Database watermarking schemes can be broadly divided distortion-based or distortion free depending upon the watermark insertion method. Some watermarking schemes are distortion-based (marking introduced errors in the underlying data) while the other is distortion free (embedding process does not add errors to the original date).

2.2.2 Watermarks Information

During watermark embedding phase *watermarks* are inserted in the database. Different researcher in the database watermark used different techniques for inserting marks such as bits, texts, image, speech etc.

2.2.3 Verifiability

Watermarking detection can be private or public.

2.2.4 Granularity Level

Insertion of watermarks in the database can be bit-level, character level, attributes level, tuple level or record level etc.

2.2.5 Cover Type

It deals with the embedding on the basis of attributes whether numerical, string, categorical or any.

2.2.6 Fragile or Robust

Watermarking schemes can be one of the two types i.e. Robust or fragile. Robust watermarking Scheme is basically designed for copyright protection and fragile for temper detection and control. We check this on the basis of intention. Fragile watermarking scheme is highly disturbed in case of pin point changing or modification made to the contents of database. On the other hand Robust watermarking Scheme ensures that embedded watermark should not be washed out by slight modifications of watermarked content.

2.3 Classification of the State Of Art Techniques

2.3.1 Distortion-Based Watermarking Techniques

2.3.1.1 Inserting bits as watermarks

Agarwal et al. [1] proposed a technique to hide watermarking information by changing the Least Significant Bits (LSBs) of numerical values. This technique depends on secret key. He assumes that the marked attributes can tolerate changes of the values. The basic idea is to ensure that some bit positions of some for the attributed of some of the tuples contain hidden information. 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 watermarks. As the sorting of tuples does not required here so given technique is computationally efficient. The technique inserts only a random bit flow into the data. This algorithm only tells about the existence of the watermark in the data. It doesn't tell about what actually the watermark is. This algorithm marks numeric data only.

2.3.1.2 Image Based Watermarking

Hu et al. [2] proposed a technique in which an image is embedded into the relational data as copyright information. The image is considered as a sequence of 0's and 1's. The marks of 0 and 1 represent integrated copyright information. The given method of using image is assumed to be correct, feasible and robust and it is consider supporting easy watermarking identification. This method tells about what the watermark information is but it has also some flaws. The author selects the tuples for watermark in an un-uniform way.

The authors have used MD5 which is not much secure according to the recent researches [15]. This method also adds errors to the actual data this may be not bearable for some categorical data. This algorithm also marks numeric data only. This method adds errors to the actual data that may be not desirable in case of some categorical data.

Odeh et al. [7] proposed a database watermarking algorithm based on inserting binary image watermarks in non-numeric multi-word attributes of selected database tuples. A large bit capacity is available to hide the marks in the database. Using image as watermarks introduces errors to the data. Peng et al. [8] introduced a technique which used image as watermark. This technique supports easy watermark identification. But it has also flaws for example, whether the watermarked data is still usable according to their method is still unwarranted because they reset the whole decimal fraction and this kind of alteration may be so insignificant that normal application of data will be affected. Li et al. [9] Authors put forward a method of watermarking database using image.

The authors tried to identify the watermarking information more easy. The robustness of the proposed algorithm was verified against a number of database attacks such subset deletion, subset addition, subset alteration and subset selection attacks. Using this method for marking whether the watermarked data is still usable according to their method is unwarranted because they reset the whole decimal fraction and this kind of alteration may be insignificant that normal application of data will be affected.

2.3.1.3 Embedding Speech Signals as Security Information:

Cui et al. [3] used Biometric feature speech and information of the copyright holder to generate the watermark information. Speech signals are introduced as watermarks to be embedded to the relations. The authors demonstrate the feasibility of introducing speech signals as watermark. Here the author presented a technique to code the signals to generate the watermark.

The technique is supposed to be unique (since everyone's speech has a distinct feature compared with others), stable (feature of speech don't change with time), universal and measurable. The authors did not uniformly distribute the marks in the data so as discussed previously the chances of subset selection attack is maximum.

2.3.1.4 Characteristic Code

Dong et al. [5] proposed the method to secure the database against invertibility attack. According to their proposed algorithm, by calculating the characteristic code of the original database on the contents of the database, the relationship between the characteristic code and the database was constructed. And the characteristic code was embedded as watermark information into the original database.

The algorithm uses characteristics code for inserting watermarks in the data and the characteristic code need the original database. Similarly in detection algorithm also need the original database. This technique lacks a very important desirable property, the blind system property.

2.3.1.5 Fake Tuple Insertion:

Pournaghshband et al. [6] Proposes a technique which embeds watermark information in the form of fake tuples. These tuples are erroneously inserted into the database. They assumed that this technique is robust against different forms of malicious attacks as well as benign updates to the data. The author has discussed his approach's robustness not quantitatively but discussed it analytically.

Keeping the track of the "fake" tuples needs a lot of memory and safe storage. If the user has set the database to not permit overwriting so the problem of primary key collision may occur. In this case this approach would fail since it would occasionally not let the real tuple to be inserted to the relation.

2.3.2 Distortion-Free Watermarking Techniques:

2.3.2.1 Inserting Virtual Marks:

Prabhakar et al. [10] presented a method which inserts a "virtual" mark through adjusting the distribution of data in each selected subset. To adjust the data's distribution is time-consuming. The method is heavily dependent on a single attribute, when the watermarked attribute is partitioned out, the whole watermark will be removed. Keeping the information about the marks required a lot of memory space as in some situations it may exceed the original database storing space.

2.3.2.2 Rearranging Order of Tuples

Cortesi et al. [4] introduced a distortion free invisible watermarking technique for relational databases. This scheme first partitioned tuples with actual attribute values. Then they applied hash function on the top of this grouping and get a watermark as a permutation of tuples in the original table. They assumed that as the changing order does not affect the original database, so the given technique is distortion free.

The proposed system by authors has embedded the marks in the statistical property of the tuples. The scheme did not add the watermark information in the data itself. These kinds of algorithms are time-consuming, computationally-intensive and keeping the information requires a lot of memory space. Sometimes it takes more space than the amount of space required for storing the original database. One other problem with this technique is that it lacks mathematical formulations. The requirements of safe storage as well as the space needed are the problems which must be solved for such techniques.

2.4 Encryption Algorithms

2.4.1 Advance Encryption Standard

AES block cipher developed by the Jon Daeman and Vicent Rijmen. Advance encryption standard (AES) support the any combination of image with key size 128, 192 and 256 bits. In AES algorithm 128 bit data divided into four basic operation block. This block are maintain by 4x4 matrices for the decryption process these 128 bit data passed through different number of round like 10,12,14. This round maintains by following transformation:

2.4.1.2 Sub Byte Transformation

Sub byte Transformation is S Substation table (S box) having properties of nonlinear substitution which is made by multiplicative inverse and affine transformation. The figure show that sub byte transformation.

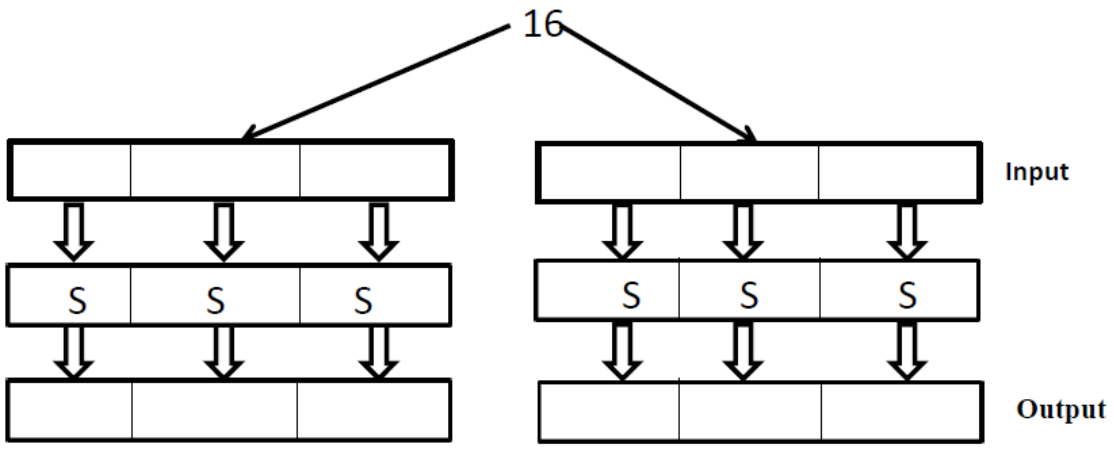


Figure 2.2: Block Diagram Substitution

2.4.1.3 Sift row Transformation

The Shift Rows step works on the lines of the state; it consistently moves the bytes in every line by a certain offset. For AES, the first column is left unaltered. Every bytes of the second line is moved one to one side. Likewise, the third and fourth columns are moved by counterbalances of two and three separately. For squares of sizes 128 bits and 192 bits, the moving example is the same. Column n is moved left round by n-1 bytes. Along these lines, every section of the yield condition of the Shift Rows step is made out of bytes from every segment of the information state. (Rijndael variations with a bigger piece size have marginally distinctive balances). For a 256-bit hinder, the first line is unaltered and the moving for the second, third and fourth column is 1 byte, 3 bytes and 4 bytes separately—this change applies for the Rijndael figure when utilized with a 256-bit obstruct, as AES does not utilize 256-bit piece.

2.4.1.4 Mix columns Transformation

This is process of matrix multiplication of the states. Every Colum multiplied by the constant matrix. It means bytes are treated as polynomial instead of a number. This stage (known as

Mix Column) is fundamentally a substitution however it makes utilization of mathematics of GF (28).

Every segment is worked on independently. Each byte of a section is mapped into another quality that is an element of every one of the four bytes in the section. The change can be controlled by the accompanying lattice increase on state.

Every component of the item matrix is the whole of results of components of one line and one segment. For this situation the individual augmentations and increases are performed in GF (28). The Mix Columns change of a solitary segment j ($0 \leq j \leq 3$) of state can be communicated right.

2.4.1.5 Add round Transformation

This is process of XOR operation of round state and round key. This transformation having the property of own inverse. In this stage (known presently) 128 bits of state are bitwise XORed with the 128 bits of the round key.

The operation is seen presently operation between the 4 bytes of a state segment and single word of the round key. This change is presently conceivable which helps in proficiency however it additionally impacts all of state.

2.4.1.6 AES Key Expansion

The AES key expansion takes right now 4-word key and produces a linear array of 44 words. Every round uses 4 of these words right. Each word contains 32 bytes which implies each sub key is 128 bits in length**.**

The function g consists of the following sub functions Rot Word performs a one-byte circular left shift on a word. This means that an input word [b0, b1, b2, b3] is transformed into [b1, b2, b3, b0]. SubWord performs a byte substitution on each byte of its input word, using this-box described earlier. The result of steps 1 and 2 is XORed with round constant, Rcon[j].

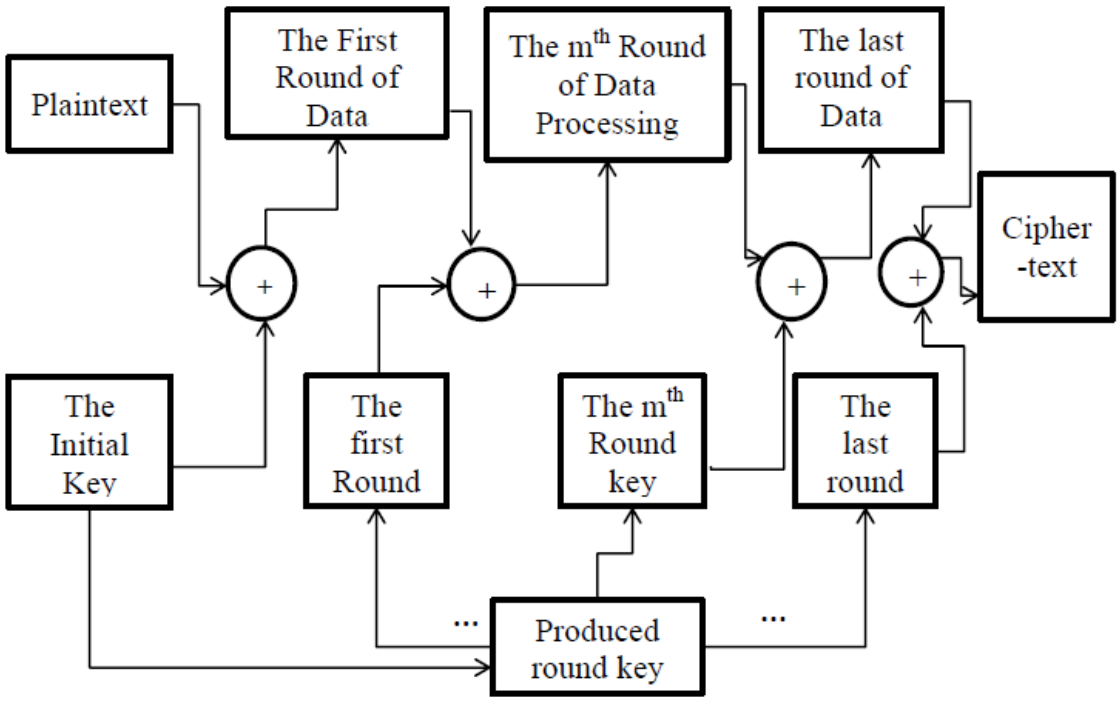


Figure 2.3 : Encryption Process of Advance Encryption Standard(AES)

The encryption and decryption have several steps. First add round, a round function work on data block related all sub operation like sub bytes, shift row, mix column and add round key will be perform. This operation will perform many times which is depending on key length. The decryption step same as perform encryption in reverse order. This Advance encryption algorithm has key size 128 bits.

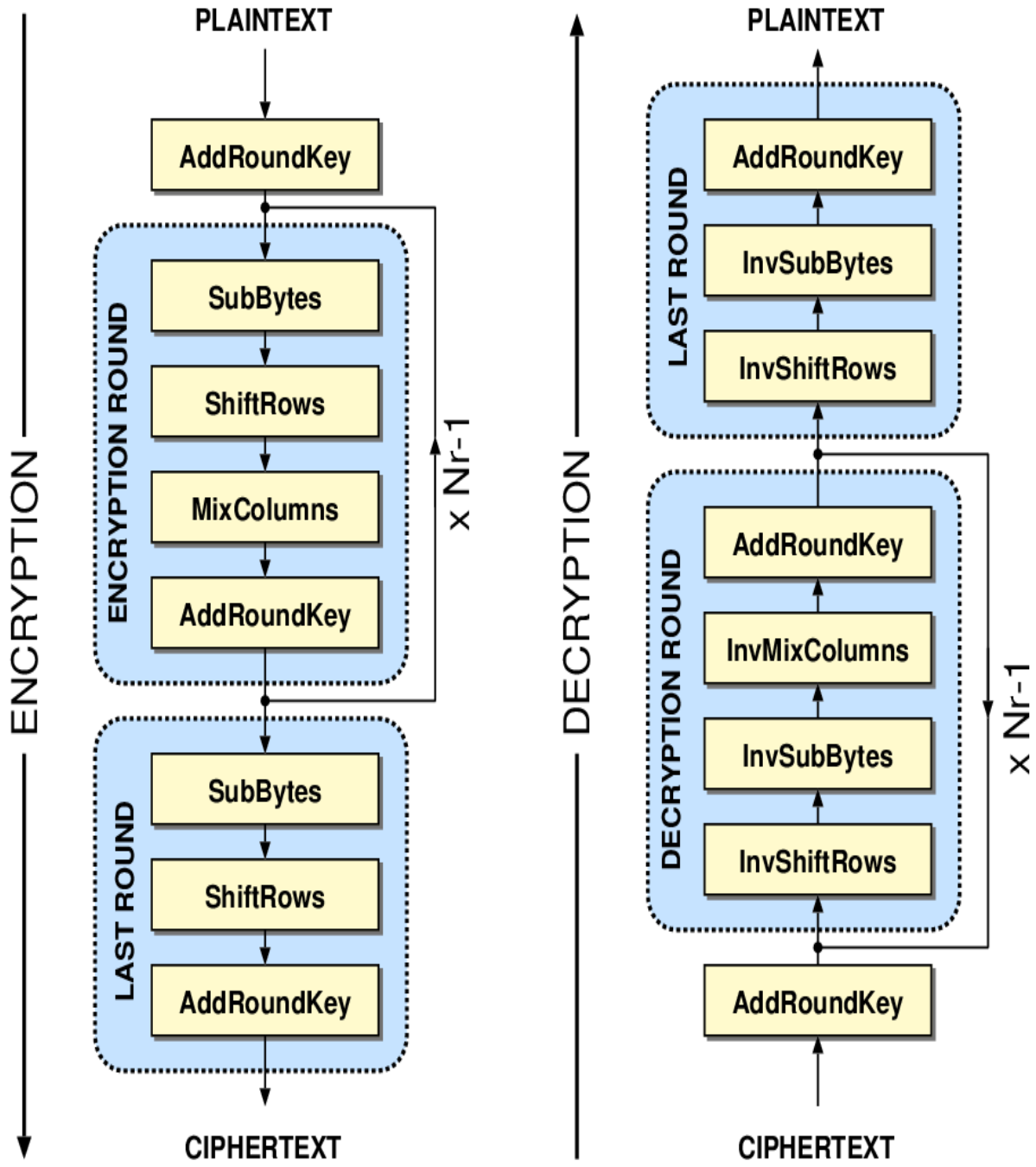


Figure 2.4: Block Diagram of Advance Encryption Standard (AES)

2.4.2 Data Encryption Standard (DES)

There are two functions in DES cryptography system. First confusion and second one is diffusion. DES has 16 number of operation. A round is combined step of confusion and diffusion step, the process of diffusion to obtain the relation between original image and

cipher image. The process of confusion to make the relationship encrypted image and key to reduce the predictability of discovering the key. DES used the 56 bit of key for 64 bit of data. The rest of 8 bit used for parity checking. The encryption process has two permutations, intimal and final permutations. In every round different 48 bits is used to produce the cipher image.

The 64 bit key of DES reduces the size of 56 bit just left every 8 bit for parity checking. The main function of parity checking to ensure that key is free from bugs. The next step is from 56 bit every 48 bits generated for each round of Des function. After that DES algorithm is computation of DSE function. This DES function applied on 48 bits key to right most 32 bits. P expansion box use for the expand 32 bit to 48 bits. 4 bit become 6 bit with repeating first and fourth bit. S box performed the compress key and expandable block. The s box has property nonlinear and gives the maximum security for DES algorithm. In decryption process all step will perform in reverse order.

Encryption Process in DES

Encryption process having two input first one image (plain image) and second one is encryption key. The encryption key select randomly form function generate secret (). It is very common that the size of image is larger than text. Now here the first step is image convert into byte of array and byte array convert into string object. The second step is defined the method for encryption and decryption and key by some awt classes. Next step is array byte of image convert into string for input as a DES algorithm. DES takes only 64 bit length of data for encryption so rest of string will pass by loop for encryption. The header of image is excludes from encryption process. Only which element start from next of header will be encrypted.

Decryption process in DES

The encrypted image divides into same block length of DES algorithm, First of 64 bit block and same key used for decryption process with reverse order of encryption algorithm. The next step decrypted text convert into same string and this string convert into byte array. By help of byte array we can get the original image.

2.5 Topological Ordering

In the field of computer science, a **topological sort** or **topological ordering** of a directed graph is a linear ordering of its vertices such that for every directed edge *uv* from vertex *u* to vertex *v*, *u* comes before *v* in the ordering. For instance, the vertices of the graph may represent tasks to be performed, and the edges may represent constraints that one task must be performed before another; in this application, a topological ordering is just a valid sequence for the tasks. A topological ordering is possible if and only if the graph has no directed cycles, that is, if it is a directed acyclic graph (DAG). Any DAG has at least one topological ordering, and algorithms are known for constructing a topological ordering of any DAG in linear time.

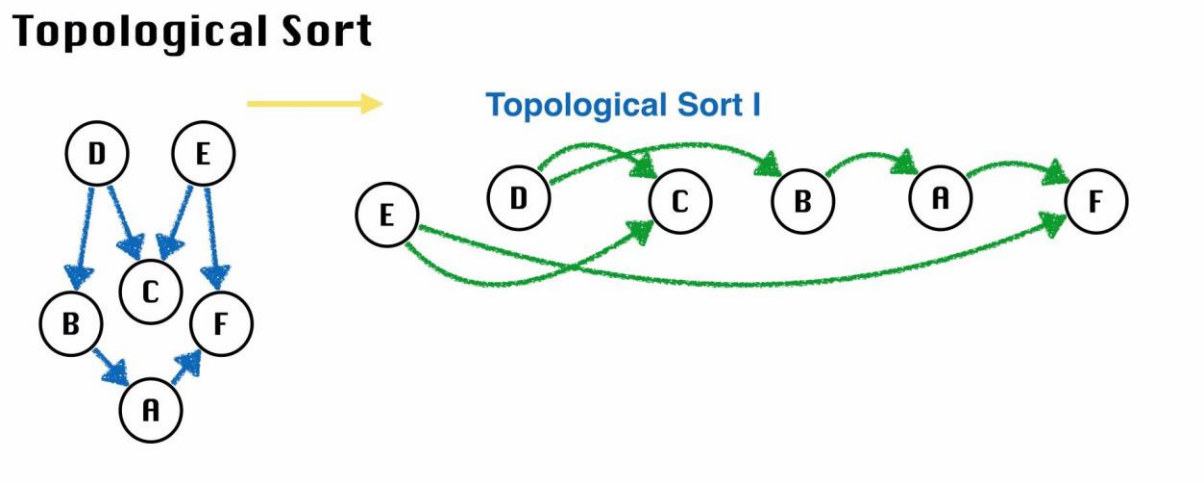


Figure 2.5 Topological sort

The canonical application of topological sorting (topological order) is in scheduling a sequence of jobs or tasks based on their dependencies; topological sorting algorithms were first studied in the early 1960s in the context of the PERT technique for scheduling in project management. The jobs are represented by vertices, and there is an edge from *x* to *y* if job *x* must be completed before job *y* can be started (for example, when washing clothes, the washing machine must finish before we put the clothes in the dryer). Then, a topological sort gives an order in which to perform the jobs.

In computer science, applications of this type arise in instruction scheduling, ordering of formula cell evaluation when re-computing formula values in spread sheets, logic synthesis, determining the order of compilation tasks to perform in make files, data serialization, and

resolving symbol dependencies in linkers. It is also used to decide in which order to load tables with foreign keys in databases.

Here, we use modified topological ordering algorithm to find a set of optimal attributes to apply watermark to them.

1. **Compute the in-degrees of all the vertices of graph**
2. **Find a vertex *U* with minimum in-degree and add it to output**
3. **Remove *U* and all its edges (*U,V*) from graph.**
4. **Update the in-degrees of all the remaining vertices**
5. **Repeat steps 2 through 4, until there are no vertices remaining**

Chapter 3

PROPOSED WORK

Problem statement

Given a database, say D, we need to produce a watermarked database D_w corresponding to D. That said, there are various other features desired in our algorithm to be taken care of, that are:

- Watermarks should not temper the actual data beyond a threshold.
- Calculating optimal hash function for hashing primary key.
- Implement various approaches to watermark database and compare their results. Various approaches are:
    - Single bit update attack
    - Multiple bit update attack
    - Mapping bits based on a secret image
    - Genetic algorithm based approach
- Testing the sample database using various attacks on database and mimicking the real time attacks that happen on attacker channel. Various attacks would be:
    - Insertion attack: The attacker may try to attack the database by inserting illegal tuples in database.
    - Update attack: The attacker may try to update the existing data in the database that can hamper the integrity of database.
    - Deletion attack: The attacker may try to delete some crucial data in database.
- The approach should preferably be easy to extend to various types of data.
- The watermarked DB should be robust.
- Finally, tabulating the results.

Our focus is to develop an information model through a statistical measure that identifies such features that do not have a significant effect on the decision making process. Mutual Information, a well-known information theory (concept), statistically measures the amount of information that one feature contains about the other features in a database.

Given the original database D having numerical data, we need to produce watermarked database where a bit of selective attributes of tuples are modified to represent a watermark. This bit is later used to check if that tuple has been tampered or not.

Single bit Algorithm for watermarking database

- ***for each tuples r of R do***
    - ***If F(r.pk) mod x equals 0 then***
        - ***Encrypt r.PK with k ,i.e. calculate H(pk,k)***
        - ***Choose 3 random bits from H(pk ,k) and select that column***
        - ***Choose another 3 bits from H(pk ,k) and select that bit of above column value***
        - ***Modify the above bit to 1.***

Choosing attributes to apply watermark is also a big concern. The database may contain several attributes and each attribute has its own relevance value. This value can be determined on basis of its dependency on other attributes. The most dependent attribute is least relevant. So, considering this we have taken a topological ordering of attributes. Based on this order, we watermark the set of least relevant attribute.

Testing

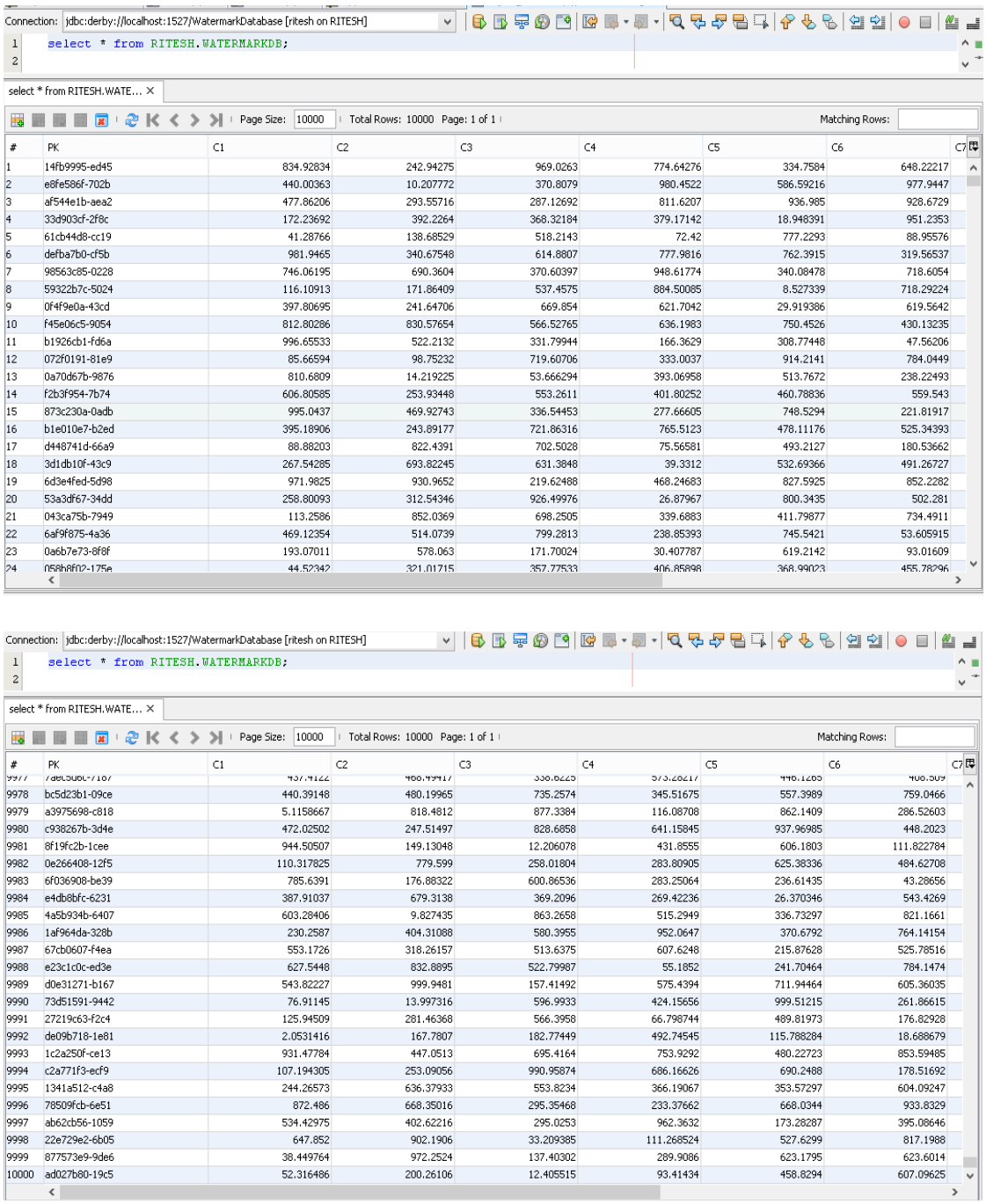
To test the above approaches, we perform two kind of attacks i.e. insertion and updating.

- In case of insertion attack, the attacker tries to insert data into database.
- In case of updating attack, the attacker tries to update the data by altering values of various attributes of tuples.

Chapter 4

Implementation, Testing and Result Analysis

We employed java to implement the algorithms and took a sample database with numerical data in it to test the database watermarking implementations.



Connection: jdbc:derby://localhost:1527/WatermarkDatabase [ritesh on RITESH]

```
1  select * from RITESH.WATERMARKDB;
2
```

select * from RITESH.WATE... ×

Page Size: 10000 | Total Rows: 10000 | Page: 1 of 1 | Matching Rows:

#	PK	C1	C2	C3	C4	C5	C6	C7
1	14fb9995-ed45	834.92834	242.94275	969.0263	774.64276	334.7584	648.22217	
2	e8fe586f-702b	440.00363	10.207772	370.8079	980.4522	586.59216	977.9447	
3	af544e1b-aea2	477.86206	293.55716	287.12692	811.6207	936.985	928.6729	
4	33d903cf-2f8c	172.23692	392.2264	368.32184	379.17142	18.948391	951.2353	
5	61cb44d8-cc19	41.28766	138.68529	518.2143	72.42	777.2293	88.95576	
6	defba7b0-cf5b	981.9465	340.67548	614.8807	777.9816	762.3915	319.56537	
7	98563c85-0228	746.06195	690.3604	370.60397	948.61774	340.08478	718.6054	
8	59322b7c-5024	116.10913	171.86409	537.4575	884.50085	8.527339	718.29224	
9	0f4f9e0a-43cd	397.80695	241.64706	669.854	621.7042	29.919386	619.5642	
10	f45e06c5-9054	812.80286	830.57654	566.52765	636.1983	750.4526	430.13235	
11	b1926cb1-fd6a	996.65533	522.2132	331.79944	166.3629	308.77448	47.56206	
12	072f0191-81e9	85.66594	98.75232	719.60706	333.0037	914.2141	784.0449	
13	0a70d67b-9876	810.6809	14.219225	53.666294	393.06958	513.7672	238.22493	
14	f2b3f954-7b74	606.80585	253.93448	553.2611	401.80252	460.78836	559.543	
15	873c230a-0adb	995.0437	469.92743	336.54453	277.66605	748.5294	221.81917	
16	b1e010e7-b2ed	395.18906	243.89177	721.86316	765.5123	478.11176	525.34393	
17	d448741d-66a9	88.88203	822.4391	702.5028	75.56581	493.2127	180.53662	
18	3d1db10f-43c9	267.54285	693.82245	631.3848	39.3312	532.69366	491.26727	
19	6d3e4fed-5d98	971.9825	930.9652	219.62488	468.24683	827.5925	852.2282	
20	53a3df67-34dd	258.80093	312.54346	926.49976	26.87967	800.3435	502.281	
21	043ca75b-7949	113.2586	852.0369	698.2505	339.6883	411.79877	734.4911	
22	6af9f875-4a36	469.12354	514.0739	799.2813	238.85393	745.5421	53.605915	
23	0a6b7e73-8f8f	193.07011	578.063	171.70024	30.407787	619.2142	93.01609	
24	058b8f02-175e	44.52342	321.01715	357.77533	406.85898	368.99023	455.78296	

Connection: jdbc:derby://localhost:1527/WatermarkDatabase [ritesh on RITESH]

```
1  select * from RITESH.WATERMARKDB;
2
```

select * from RITESH.WATE... ×

Page Size: 10000 | Total Rows: 10000 | Page: 1 of 1 | Matching Rows:

#	PK	C1	C2	C3	C4	C5	C6	C7
9978	bc5d23b1-09ce	440.39148	480.19965	735.2574	345.51675	557.3989	759.0466	
9979	a3975698-c818	5.1158667	818.4812	877.3384	116.08708	862.1409	286.52603	
9980	c938267b-3d4e	472.02502	247.51497	828.6858	641.15845	937.96985	448.2023	
9981	8f19fc2b-1cee	944.50507	149.13048	12.206078	431.8555	606.1803	111.822784	
9982	0e266408-12f5	110.317825	779.599	258.01804	283.80905	625.38336	484.62708	
9983	6f036908-be39	785.6391	176.88322	600.86536	283.25064	236.61435	43.28656	
9984	e4db8bfc-6231	387.91037	679.3138	369.2096	269.42236	26.370346	543.4269	
9985	4a5b934b-6407	603.28406	9.827435	863.2658	515.2949	336.73297	821.1661	
9986	1af964da-328b	230.2587	404.31088	580.3955	952.0647	370.6792	764.14154	
9987	67cb0607-f4ea	553.1726	318.26157	513.6375	607.6248	215.87628	525.78516	
9988	e23c1c0c-ed3e	627.5448	832.8895	522.79987	55.1852	241.70464	784.1474	
9989	d0e31271-b167	543.82227	999.9481	157.41492	575.4394	711.94464	605.36035	
9990	73d51591-9442	76.91145	13.997316	596.9933	424.15656	999.51215	261.86615	
9991	27219c63-f2c4	125.94509	281.46368	566.3958	66.798744	489.81973	176.82928	
9992	de09b718-1e81	2.0531416	167.7807	182.77449	492.74545	115.788284	18.688679	
9993	1c2a250f-ce13	931.47784	447.0513	695.4164	753.9292	480.22723	853.59485	
9994	c2a771f3-ecf9	107.194305	253.09056	990.95874	686.16626	690.2488	178.51692	
9995	1341a512-c4a8	244.26573	636.37933	553.8234	366.19067	353.57297	604.09247	
9996	78509fcb-6e51	872.486	668.35016	295.35468	233.37662	668.0344	933.8329	
9997	ab62cb56-1059	534.42975	402.62216	295.0253	962.3632	173.28287	395.08646	
9998	22e729e2-6b05	647.852	902.1906	33.209385	111.268524	527.6299	817.1988	
9999	877573e9-9de6	38.449764	972.2524	137.40302	289.9086	623.1795	623.6014	
10000	ad027b80-19c5	52.316486	200.26106	12.405515	93.41434	458.8294	607.09625	

Figure 4.1 Screenshots of sample database

No. of rows inserted during attack	Total no. of rows	No. of violations detected	Violation Percentage(%)
1000	11000	460	4.18
2000	12000	944	7.86
3000	13000	1422	10.93
4000	14000	1881	13.43
5000	15000	2363	15.75

Table 1: Insertion attack statistics

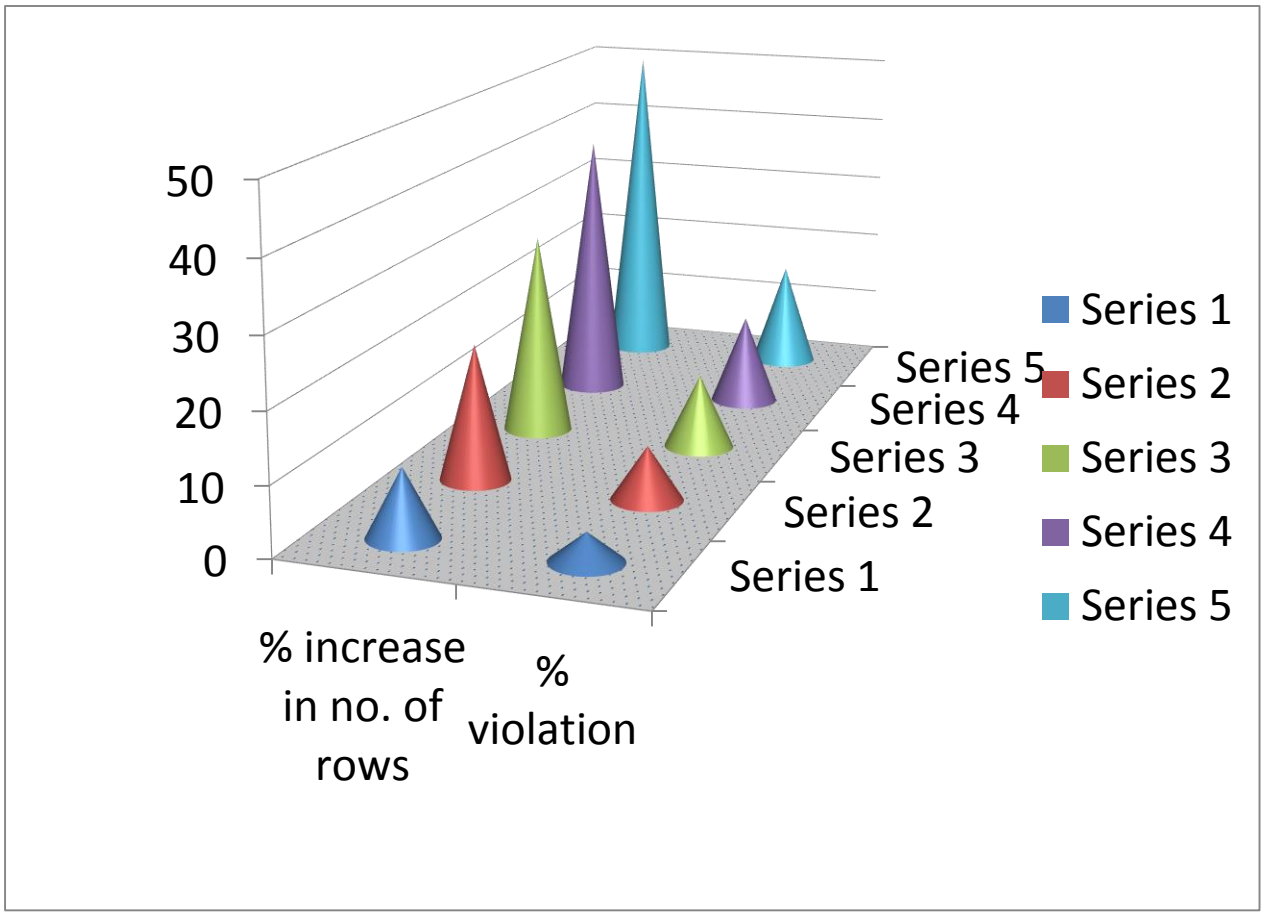


Figure 4.2: Graphical representation of insertion attack

Updating Attack

No. of rows updated during attack	No. of violations detected	Violation Percentage(%)
1000	54	.54
2000	71	.71
3000	157	1.57
4000	210	2.1
5000	279	2.79

Table 2: Violations percentage corresponding to 1 attribute update

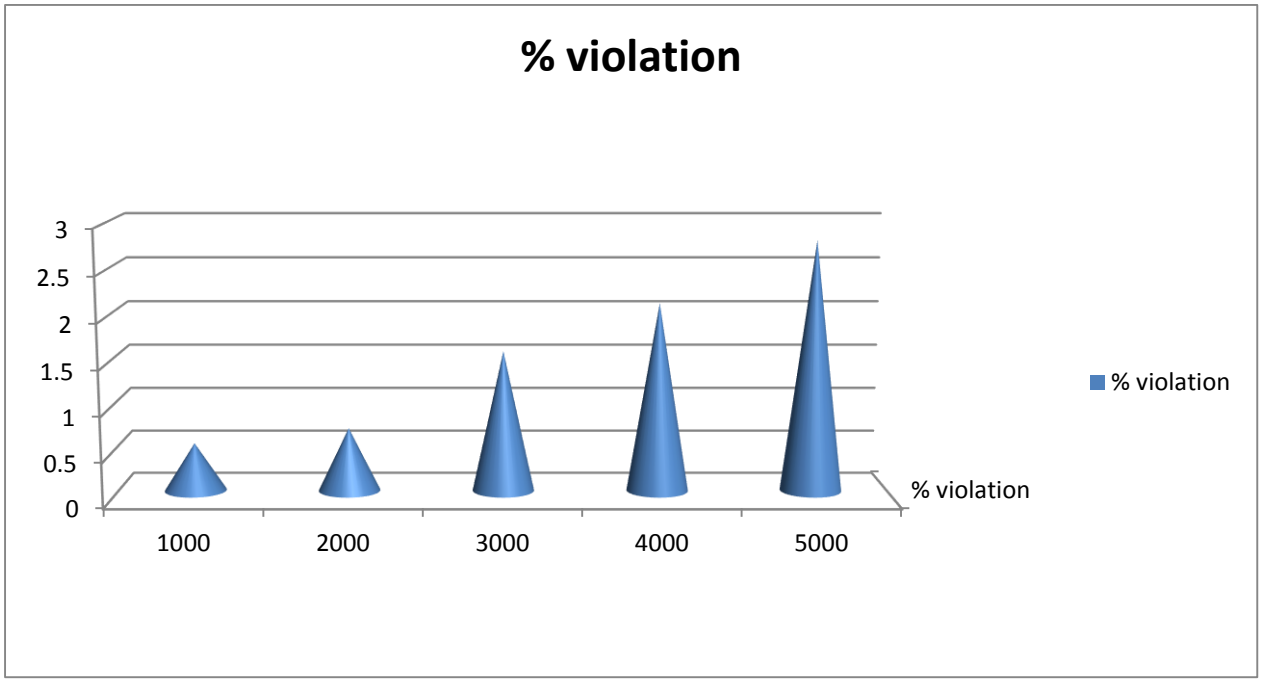


Figure 4.3 Graphical representation of update attack on 1 attribute

No. of rows updated during attack	No. of violations detected	Violation Percentage(%)
1000	72	.72
2000	92	.92
3000	187	1.87
4000	262	2.62
5000	314	3.14

Table 3 : violations percentage corresponding to 2 attribute updating

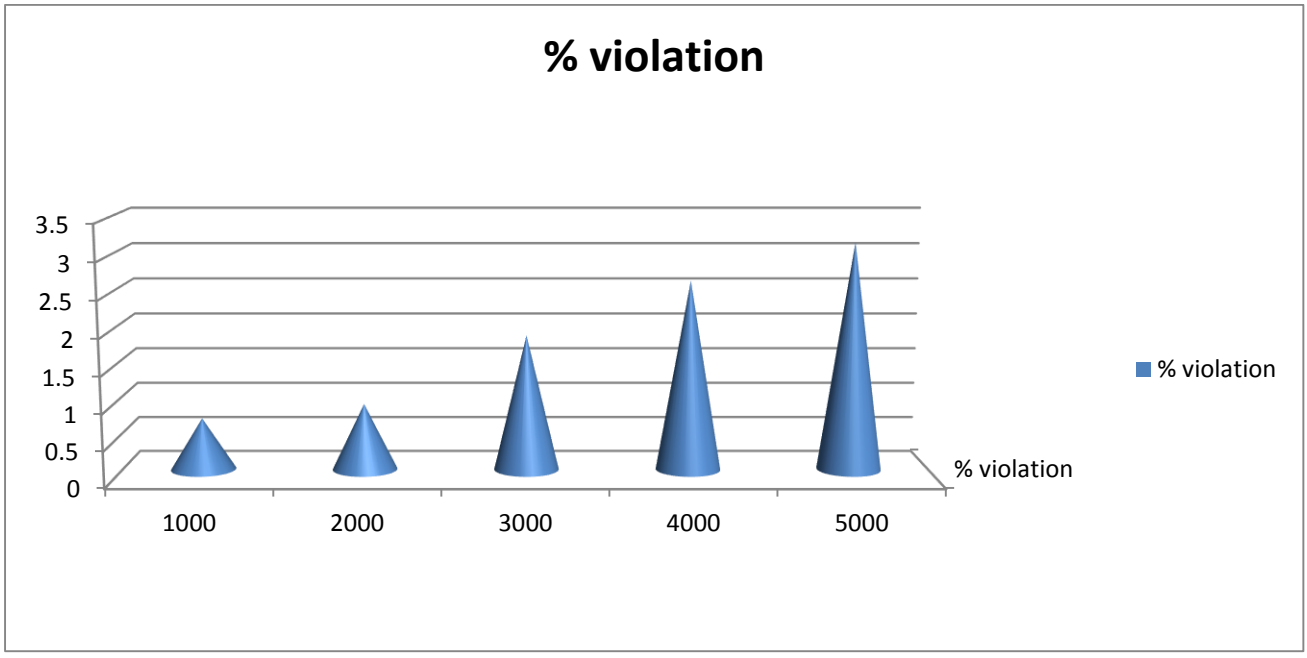


Figure 4.4 Graphical representation of update attack on 2 attributes

No. of rows updated during attack	No. of violations detected	Violation Percentage(%)
1000	92	.92
2000	101	1.01
3000	196	1.96
4000	299	2.99
5000	330	3.3

Table 4: violations percentage corresponding to 3 attribute updating

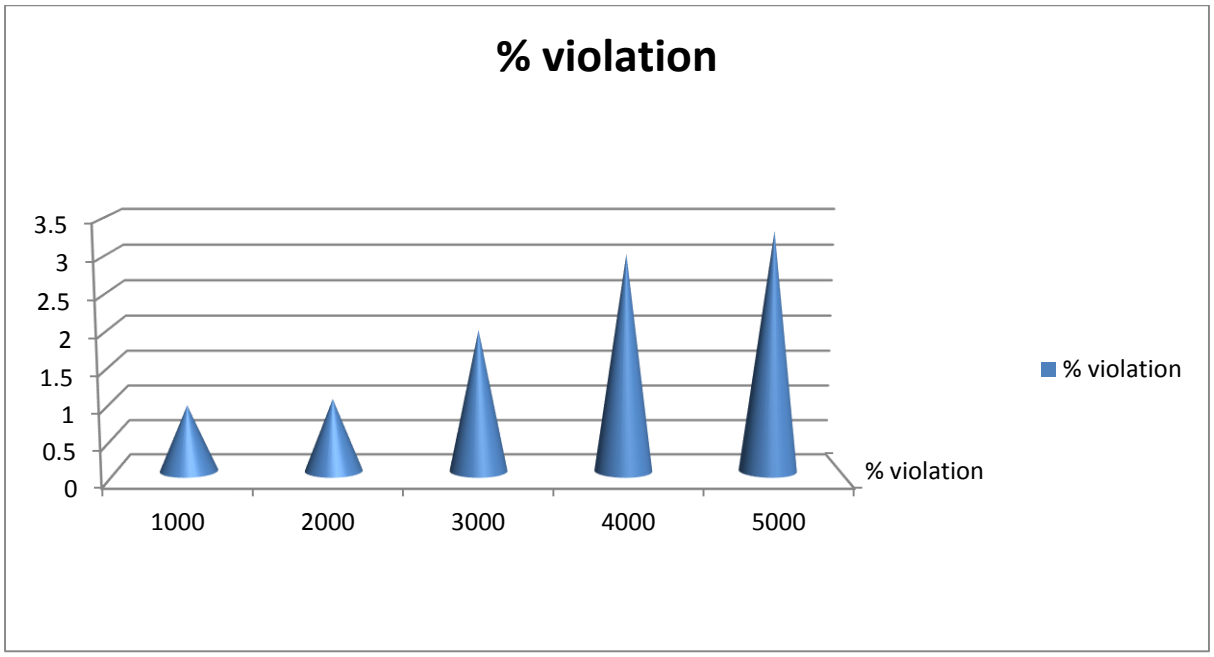


Figure 4.5 Graphical representation of update attack on 3 attributes

No. of rows updated during attack	No. of violations detected	Violation Percentage(%)
1000	96	.96
2000	141	1.41
3000	236	2.36
4000	317	3.17
5000	364	3.64

Table 5: violations percentage corresponding to 4 attribute updating

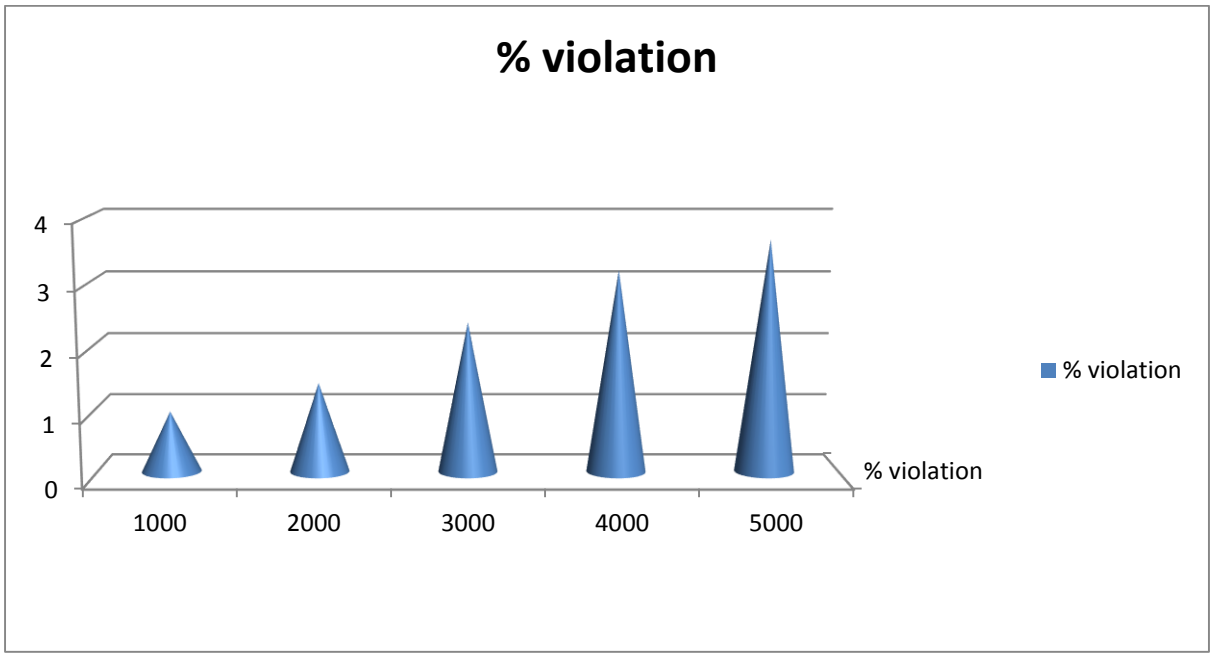


Figure 4.6 Graphical representation of update attack on 4 attributes

No. of rows updated during attack	No. of violations detected	Violation Percentage(%)
1000	110	1.1
2000	171	1.71
3000	283	2.83
4000	392	3.92
5000	430	4.3

Table 6: violations percentage corresponding to 5 attributes updating

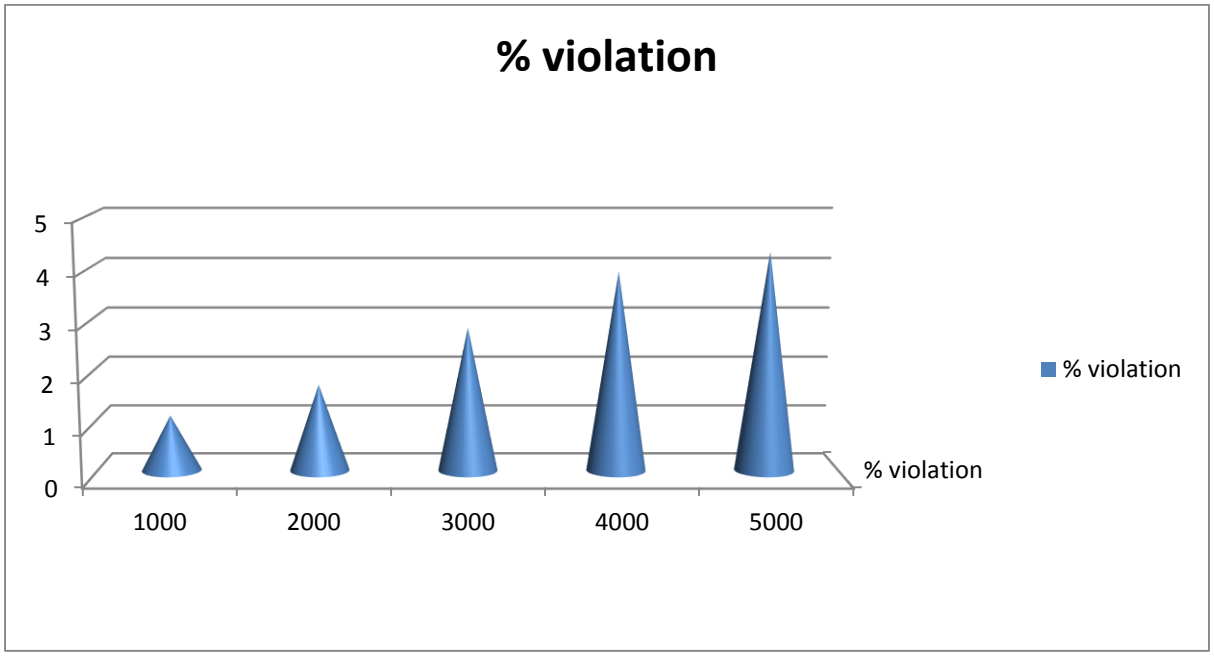


Figure 4.7 Graphical representation of update attack on 5 columns

Chapter 5

Conclusion

The rapid change in the field of information technology and computer science made possible the cost-free distribution of data via internet. It is benefiting the mankind but on the other hand it has made difficult to maintain security, privacy and ownership claiming. To resolve the issue of the Intellectual Property Right Protection (IPR) through Watermarking is gaining attentions globally. However the use of relational database in many real-life applications made owners of the relational database curious about the security, authentication of the database. For this purpose watermarking relational database playing a vital role in maintaining authentication, copy right protection, privacy and integrity from the last one decade. However for taking full advantages, the watermarking techniques have yet to undergo through more challenges. In the proposed scheme the whole database is considered in the watermark insertion process, reducing the chances of subset selection attack. It is 100% distortion-free technique because it *virtually* inserts the watermarks in the data but the concept of zero-*watermarking* is applied. As far as the concept of relational database watermarking is introduced, a lot of literature can be seen on numeric database. The user/owner of data can set thresholds for accepted violations and can detect if the database has been tampered or not.

Future work

The proposed technique is restricted to numeric data only. In future, we are expecting to extend it by cause of other attacks i.e. insertion and updating attack is considered but the proposed technique can be used as protection against other attacks too. Increasing the number of more than one secret key is also in future plan combining the same method with non-numeric also.

REFERENCES

[1] Rakesh Agarwal,Jerry Kiernan *watermarking Relational databases*,Proceedings of the 28th VLDB Conference, Hong Kong, China,2002.

[2] Zhongyan Hu,Zaihui Cao,Jianhua Sun *An image Based Algorithm For Watermarking Relational Database*,2009 International Conference on Measuring Technology and Mechatronics Automation IEEE, 2009.

[3] Haiqing Wang,Zinchn Cui and zaihui Cao *A speech Algorithm for Watermarking Relational Databases* ,2008 International Symposiums on Information Processing,IEEE,2008.

[4] Sukriti Bhattavcharya ,Agostino Cortesi *A Distortion Free Watermarking Framework For Relational Databases* ,2008 .

[5] Xiaomei Dong,Xiaohua Li,Ge Yu,Lei Zheng *An Algorithm Resistive to Invertibility attack in Watermarking Relational Databases,* IEEE, 2009.

[6] Vahab Pournaghshband) *A New Watermarking approach for Relational Data* ,ACM , 2008.

[7] Ali Al-Haj and Ashraf Odeh *Robust and Blind Watermarking of Relational Database Systems*,Journal of Computer Science ,2008.

[8] Xiang Zhou,Min Huang and zhiyoung Peng *An Additive-proof watermarking Mechanism for Databases' Copyrights Protection Using Image*,ACM , 2007.

[9] Zhihao Zhang, Xiaoming Jin, Jianmin Wang, Deyi Li *Watermarking Relational Database Using Image* ,Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August IEEE,2004.

[10] R.Sion, M.Atallah, and S.Prabhakar *On Watermarking Numeric Datasets*

[11] Julien LAFAYE*An Analysis of Database Watermarking Security*,Third International Symposium on Information Assurance and Security IEEE ,2007.

[12] Hsien-Chu Wu, Fang-Yu Hsu, and Hwang-Yu *Chen Tamper Detection of Relational Database Based on SVR Predictive Difference,* Eighth International Conference on Intelligent Systems Design and Applications,2008.

[13] Xiaoyun wang ,yiqun Lisa and Hongbo Yu*Finding Collisions in the Full SHA-1*,2005.

[14] Zunera Jalil , *Copyright Protection of Plain Text Using Digital Watermarking , 2010* . *http://prr.hec.gov.pk/Thesis/718S.pdf*

[15] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn*Information Hiding|A Survey,* 1999

[16] zhang yong, niu xia-mu , wu di, zhao liang, li jun-cao, xu wei-jun *a method of Verifying Relational Databases Ownership withImage Watermark* , 2004. [17] Muhalim Mohamed Amin, Subariah Ibrahim, Mazleena Salleh , Mohd Rozi Katmin *Information Hiding Using Steganography* , 2003.

[18]Raju Halder, Shantanu Pal ,Agostino Cortesi *Watermarking Techniques for Relational Databases:Survey, Classification and Comparison*, 2010.

[19] Ch Arathi , Literature Survey on Distortion based Watermarking Techniques for Databases, *International Journal of Computer Science & Communication Networks, Vol 2(4), 456-463*,2010.

[20]Erik Sonnleitner, A robust watermarking approach for large databases, *Institute for Application Oriented Knowledge Engineering, Johannes Kepler University Linz Altenbergerstr. 69, 4040 Linz, Austria, 2012.*

[21] DBLP Bibliography Database: DBLP-Citation-network V5, http://arnetminer.org/citation.

[22] Y. Zhang, B. Yang, and X.-M. Niu, "Reversible watermarking for relational database authentication," J. Comput., vol. 17, no. 2, pp. 59–66, 2006.

[23] G. Gupta and J. Pieprzyk, "Reversible and blind database watermarking using difference expansion," in Proc. 1st Int. Conf. Forensic Appl. Tech. Telecommun., Inf., Multimedia Workshop, 2008, p. 24.

[24] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in Proc. IEEE Int. Conf. Image Process., 2003, pp. I–501,vol. 1.

[25] G. Gupta and J. Pieprzyk, "Database relation watermarking resilient against secondary watermarking attacks," in Information Systems and Security. New York, NY, USA: Springer, 2009, pp. 222–236.

[26] J.-N. Chang and H.-C. Wu, "Reversible fragile database watermarking technology using difference expansion based on SVR prediction," in Proc. IEEE Int. Symp. Comput., Consum. Control, 2012, pp. 690–693.

[27] K. Jawad and A. Khan, "Genetic algorithm and difference expansion based reversible watermarking for relational databases," J. Syst. Softw., vol. 86, no. 11, pp. 2742–2753, 2013.

[28] M. E. Farfoura and S.-J. Horng, "A novel blind reversible method for watermarking relational databases," in Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl., 2010, pp. 563–569.

[29] D. M. Thodi and J. J. Rodriguez, "Prediction-error based reversible watermarking," in Proc. IEEE Int. Conf. Image Process. 2004, vol. 3, pp. 1549–1552.

[30] D. M. Thodi and J. J. Rodriguez, “Reversible watermarking by prediction-error expansion,” in Proc. 6th IEEE Southwest Symp. Image Anal. Interpretation, 2004, pp. 21–25.

[31] D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730, Feb. 2007.

[32] M. E. Farfoura, S.-J. Horng, J.-L. Lai, R.-S. Run, R.-J. Chen, and M. K. Khan, “A blind reversible method for watermarking relational databases based on a time-stamping protocol,” Expert Syst. Appl., vol. 39, no. 3, pp. 3185–3196, 2012.

[33] X. Li, B. Yang, and T. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524– 3533, Dec. 2011.

[34] E. Sonnleitner, “A robust watermarking approach for large databases,” in Proc. IEEE First AESS Eur. Conf. Satellite Telecommun., 2012, pp. 1–6.

[35] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan, “Biased minimax probability machine for medical diagnosis,” AMAI, 2004.

[36] K. Bache and M. Lichman. (2013). UCI machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml

[37] A. Reiss and D. Stricker, “Introducing a new benchmarked dataset for activity monitoring,” in Proc. IEEE 16th Int. Symp. Wearable Comput., 2012, pp. 108–109.

[38] M. Mitchell, An introduction to genetic algorithms. Cambridge, MA, USA: MIT Press, 1996.

[39] S. Bhatia, P. Prakash, and G. Pillai, “Svm based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features,” in Proc. World Congr. Eng. Comput. Sci., 2008, pp. 22–24.