A
Dissertation
On

Database Watermarking Using Elliptic Curve Cryptography (ECC)

Submitted in Partial Fulfillment of the Requirement

For the Award of the Degree of

Master of Technology

in

Computer Science and Engineering

by

Archana Saxena
University Roll No. 2K13/CSE/03

Under the Esteemed Guidance of

Manoj Kumar
Associate Professor
Computer Science & Engineering Department, DTU

2014-2015

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

DELHI TECHNOLOGICAL UNIVERSITY

DELHI – 110042, INDIA

ABSTRACT

The piracy of software, images, video, audio, and text has long been a concern for owners of these digital assets. Protection schemes are usually based upon the insertion of digital watermarks into the data. The watermarking software introduces small errors into the object being watermarked. These intentional errors are called *marks*, and all the marks together constitute the *watermark*. The marks are chosen so as to have an insignificant impact on the usefulness of the data and are placed in such a way that a malicious user cannot destroy them without making the data significantly less useful.

Although watermarking does not prevent illegal copying, it deters such copying by providing a means for establishing the original ownership of a redistributed copy.

The increasing use of databases in applications beyond "behind-the-firewalls data processing" is creating a similar need for watermarking databases. The Internet is exerting tremendous pressure on data providers to create services that allow users to search and access databases remotely. Although this trend is a boon to end users, it exposes the data providers to the threat of data theft. Providers are therefore demanding technology for identifying pirated copies of their databases. So database watermarking becomes a basic requirement of database owners.

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Manoj Kumar, Associate Professor, Computer Science & Engineering Department, DTU 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 like to acknowledge Mr. R. K. Yadav, Assistant Professor, Computer Science & Engineering Department, DTU for his constant support and I would also like to acknowledge the Computer Science & Engineering Department and 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, spouse, daughter and friends for constantly encouraging me during the completion of work.

Archana Saxena

University Roll no: 2K13/CSE/03

M.Tech (Computer Science & Engineering)
Department of Computer Science & Engineering

Delhi Technological University

Delhi – 110042

Computer Science & Engineering Department
Delhi Technological University
Delhi-110042
www.dtu.ac.in

CERTIFICATE

This is to certify that the dissertation titled **"Database Watermarking using Elliptic Curve Cryptography (ECC)"** is a bonafide record of work done by **Archana Saxena, Roll No. 2K13/CSE/03** at **Delhi Technological University** for partial fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

(Manoj Kumar)
Associate Professor & Project Guide
Department of Computer Science & Engineering
Delhi Technological University

Date: __ __ ____

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ECC	Elliptic Curve Cryptography
EMC	Encrypted Mark Code
MD	Message Digest
LSB	Least Significant Bits
OWL	Ontology Web Language
ECM	Elliptic Curve Method
PKC	Public-Key Cryptography
RSA	Asymmetric Encryption Method
MAC	Message Digest Function
HASH	Hash Function
RRW	Robust and Reversible Watermarking
GA	Genetic Algorithm
PDA	Personal Digital Assistant
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie Hellman
ECIES	Elliptic Curve Integrated Encryption System
DSA	Digital Signature Algorithm
SEP	Stable Election Protocol
IES	Integrated Encryption Scheme
EC	Elliptic Curve
MSB	Most Significant Bit
GB	Giga Bits
RAM	Random Access Memory
GHz	Giga Hertz
CPU	Central Processing Unit

CHAPTER 1

INTRODUCTION

Due to increasing popularity of Internet, data piracy has become a big issue where data owners are more concern for their digital assets. To protect these digital assets, a digital watermark can be inserted in to the data, where data can be any software, image, video, audio or text. To create a watermark in to an object, watermarking software creates an intentional small error in to the object that is called *mark* and group of all these marks into the object collectively is called a *watermark*.

Now-a-days use of database is increasing very rapidly in every application. The Internet is creating many applications where users can access and search the data remotely so data providers are creating the services for users. So it also required some watermarking techniques that can insert watermark into the database without changing the meaning and value of any field of the database. It is the biggest challenge of database watermarking techniques to insert a watermark into a database without changing the value of any attribute. Data providers are demanding for the database watermarking technologies to identify the pirated copies of their databases.

Many watermarking techniques are based on different watermark information; most of these techniques are designed for numerical database and are distortion based. There are almost similar steps to identify attribute then tuple and then marking position for the watermark. Most of these techniques used a single attribute of a tuple to embed a watermark. So, this work will be extended towards embedding the watermarks at different attributes at different places. Therefore, it will be difficult for attacker to remove watermarks from different places from the database. Most of these techniques are also depend on presence of primary key.

To insert a watermark into the database, a basic technique is used. Where watermark information is first converted into a binary digital watermark and then inserted into the original database through a user's key in such a way that attacker will not be able to identify the inserted watermark. If attacker makes any changes into watermarked database, like delete, update or insert tuples, watermark extraction method will detect the inserted watermark bits through the user's key. The technique is shown in figure 1.1.

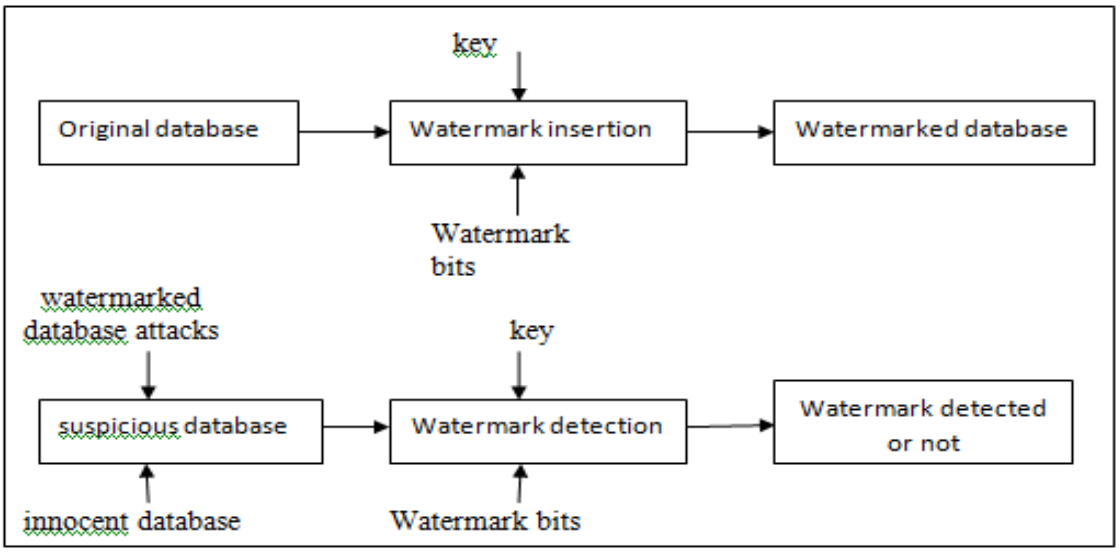

Figure 1.1 Basic Watermarking Technique[32]

1.1 Digital Watermarking

The area of digital watermarking became popular as a research in mid 1990s. Watermarking is the process of hiding a secret message into an image, text, audio, video or other type of objects. It is an effective tool to create trust management, integrity protection and copyright.

1.1.1 Characteristics of Digital Watermarking

Digital Watermarking is the method to hide any secret information to achieve trust management, integrity protection and copyright for digital asset like database, software, multimedia etc. The characteristics of a robust digital watermarking are[24].

1. Watermark should not reduce the quality of the data.
2. Multiple inserted watermarks into a data should not interfere with each other.
3. If a user generate different copies of the same object and distribute it to different users with different watermarks, then it should not be possible for any user to create a new copy from different watermarked copies that identifies none of them.
4. Watermark should survive all the prescribed attacks that is possible on watermarked data without degrading the quality of data.

The key idea behind any sort of digital watermarking is to introduce imperceptible (so that the attacker cannot detect them) and tolerable (to ensure that the value of data is not greatly depreciated) errors to the object.

1.1.2 Digital Database Watermarking

The generally speaking, database digital watermarking model mainly includes three algorithms: digital watermarking generation algorithm, digital watermarking embedding algorithm and digital watermarking extraction algorithm[33].

1.1.2.1 Digital Watermarking Generation Model

Digital watermarking can be any text or image. To insert a watermark into a database, watermarking should be processed and converted to some binary stream. Watermarking generation model is shown in figure 1.2.

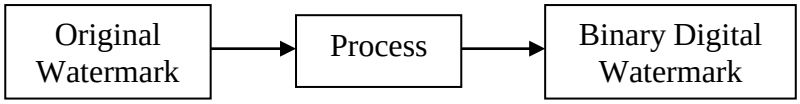

Figure 1.2 Digital watermarking generation model

1.1.2.2 Digital watermarking embedding Model

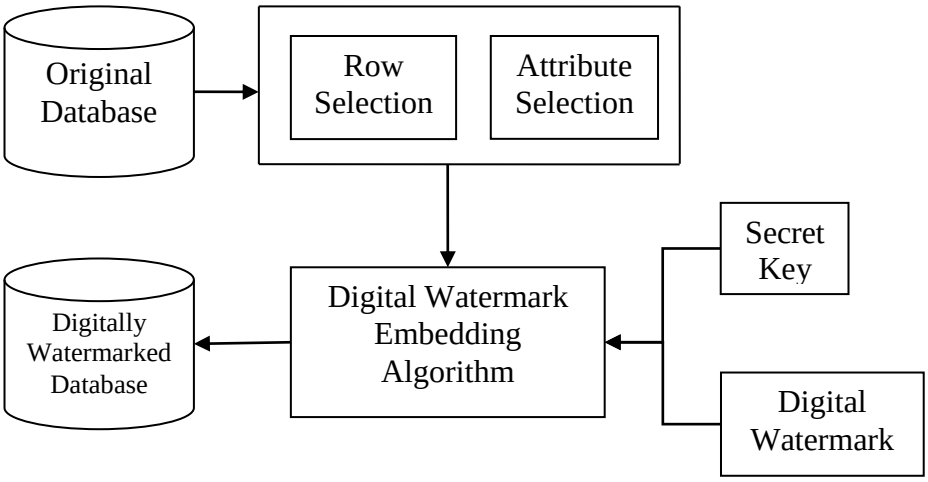

Figure 1.3 Digital watermarking embedding model

Digital watermarking embedding is usually hide the processed binary stream into some database data by digital watermarking embedding algorithm and doesn't affect the use of database. Watermarking embedding model is shown in figure 1.3.

1.1.2.3 Digital Watermarking Extraction Model

To extract the watermark Secret Key is utilized. This model extracts watermarking signal from database by watermarking extraction algorithm, and recovers to the original digital watermarking signal after process. This model is shown in figure 1.4.

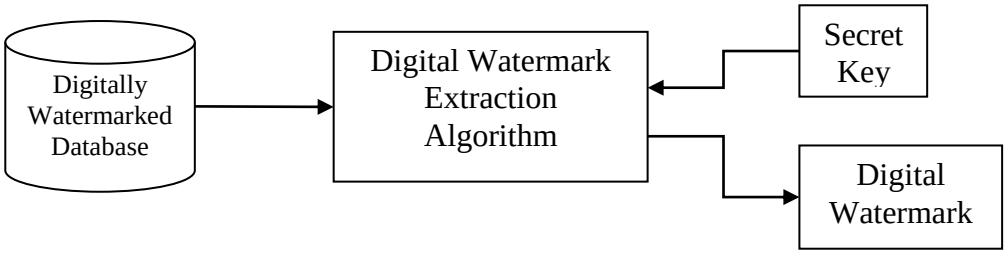

Figure 1.4. Digital watermarking extraction model

1.1.3 Challenges of Digital Database Watermarking

A watermark can be attacked through different ways. A watermarking technique should be resistant for intentional and unintentional attacks[9]. The type of attacks that is possible on user's watermarked data are defined as follow:

Benign updates: It can be unintentional update, where rows that are marked may be deleted or marked bits may be changed during any ordinary data processing.

Malicious attacks: An attacker intentionally steals the data and may attack to remove watermark in different forms as described follows:

- ***Bit Attacks:*** It is the simplest attack to remove a watermark by altering one or more bits, e.g., by deterministically flipping each bit or by setting each bit to 0 or 1 according to the independent toss of a fair coin. The effectiveness of such an attack is sensitive to the number of altered bits: if attacker alters every bit in the database, then he can easily destroy the watermark but in this way he can make his data useless.

- ***Rounding Attack:*** An attacker can round off the values of all numeric attributes to remove the marks hidden in decimal numbers. But he can degraded the quality of the data.
- ***Transformations:*** An attack related to the rounding attack is one in which the numeric values are linearly transformed. For example, attacker may convert the data to a different unit of measurement (e.g., Fahrenheit to Celsius).
- ***Subset Attack:*** Attacker may take a subset of the tuples or attributes of a watermarked relation and hope that the watermark is lost.
- ***Mix-and-Match Attack:*** Attacker may create his relation by taking disjoint tuples from multiple relations containing similar information.

False claims of ownership: An attacker can create a false claim of ownership, where he can insert his watermark into other user's watermarked data.

1.1.4 Properties of Digital Database Watermarking

A watermarking technique should satisfy the following properties as describe in following points [9]:

- ***Robustness:*** Watermarks should be robust against degradation caused by either benign updates or malicious attacks.
- ***Accuracy:*** User should, with high probability, not detect his watermark in someone else's non pirated database. We call such an erroneous detection a *false hit*.
- ***Incremental updatability:*** As User adds/deletes tuples or modifies the values of attributes; the watermark should be incrementally updatable. That is, the watermark values should only be recomputed for the added or modified tuples.
- ***Blind system:*** Watermarking detection technique should not require the original database or the watermark. This property is very complicated as it allows the watermark to be extracted in a copy of the database, irrespective of later updates to the original relation.
- ***Public system:*** The watermarking system should assume that the method used for inserting a watermark is public.

1.2 Related Previous Work

There is a tremendous work has been done in the area of Database watermarking as well as Elliptic Curve Cryptography. The following subsections describe the previous work in these areas.

1.2.1 Relational Data Base Watermarking

Agarwal el at.[9, 10] proposed the relational database watermarking in 2002. This method was only implemented for numeric data and marking done at a bit level.

Zhang et al[11] proposed the technique for image based relational database watermarking. The pixels of an image have the relative positions. This method is based on the tuples' order. The drawback of this method was that watermark extraction was not possible if attacker change the order of the tuples or attributes. This method is not efficient against some subset attacks.

Sun et al.[12] proposed another method to insert an image into the database as watermark information, where image was converted into flow of bits and inserted according to the hash value of the database tuple. They used hash value of database tuple to find the location of each pixel and marked bit. They considered mod of hash value and watermark's length. If someone takes large image as watermark information, then length of watermark increases. And this method cannot insert all the pixels into the database. Therefore this method is not efficient for small databases.

Wang et al.[13] proposed a different method to insert an image into the database as watermark information based on Arnold transform, where image is used as a binary string. This method is more effective because it used many factors. Suppose the length of this binary string is L. And whole database is divided into L groups. It computes 1hash value using private key, database's primary key and order of an image. According to this hash value, particular groups among all L groups are found. The ith bit of binary string is inserted into the specific bit position of the attribute value. This ith bit is chosen algorithmically. The efficiency of this technique is improved because it depends on many factors like private key, scrambling number and order of an image. This technique uses only one fixed attribute to insert watermarks.

Cao et al.[14] proposed an advanced technique to insert an image into relational database, where an image is converted into bit flow using EMC (Encrypted Mark Code). This technique does not consider the order of the image also. At last, the usefulness of the database is checked. The modification is committed if acceptable, otherwise rolled back.

Theodoros[15] proposed multipurpose scheme because it can be used for both watermarking (the same bit string is embedded and detected in every database copy) and fingerprinting(the different bit string is embedded and detected in each database copy). The watermark might be any digital object related to the underlying data, for example an image, a logo, a text message, a sound, a speech signal, etc. The encoding method can be independently applied to each tuple, therefore, the proposed method has the property of incremental updateability, i.e., the watermarked database can support normal user modifications (insertions, deletions and updates) by simply applying the encoding method to the tuples which are involved, without affecting any other.

Rao et al.[16] also proposed the method to insert an image into the database as watermark information, where image is processed as binary image. This method reduces the changes in the watermarked database.

Yu Fu et al.[17] proposed a different method of database watermarking where a secret key is divided into multiple parts and hide into the relational database. The (k, n) threshold scheme is used to decide the minimum number of parts required to recover the secret data completely.

The steps of the Deshpande et al.[18] proposed mechanism of watermarking relational database mainly involve decoding and encoding on numerical attribute of relational database. The first phase is to partition the original data and assign partition number to each and every tuple of the relation using Cryptographic Hashing Function (MD5). In the second phase , while changing the data , select the desired watermark and bit bi is selected from the partitioned data and then that bit bi is changed using watermark W. When the original value of data gets changed due to the watermark bit, it always checks the data usability constraints. In the third phase, after inserting the watermark in the partition, merge all partitions and get the complete watermarked data. While decoding, use majority voting algorithm to get the correct watermark. Detecting the watermark

neither requires access to neither the original data nor the watermark. The watermark can be detected even in a small subset of a watermarked relation as long as the sample contains some of the marks. This technique also handles the attack on database like tuple deletion, alteration and insertion.

Hanyurwimfura et al.[19] proposed an advanced technique for watermarking relational database for non-numeric data. The mark is inserted through the horizontally shifting the words within selected attribute of selected tuples; a word is displaced right or left unmoved depending on watermark bit . The location of insertion is calculated by Levenshtein Distance between two successive words within an attribute. This method is effective as it is robust against different forms of malicious attacks and it is blind as it does not require the original database in order to extract the embedded watermark.

Huang et al.[20] proposed a watermarking technique for relational database which is based on cluster theory, where a cluster is used to insert watermark into database. The cluster theory is used to cluster the source data and the clustering results determine the quantity of embedded watermark information and embedded position. This method makes the watermark information more disperse and hidden. They also introduced the method of odd-even modifying method for watermark information to decline the modification of original database.

Zhang et al.[21] approach is applied for protecting both textual and numerical data. This is done by embedding special mark and watermark bits into textual attributes and numerical attributes respectively. Carriage return character and linefeed character, representing 1 and 0 of watermarking bits respectively, are inserted into textual data, which does not change original appearance and meaning of textual data in relational databases. For numerical data, Watermarks are embedded into one of least significant bits (LSB) of the optional attributes. Watermark extraction does not require original data. Experiments show that even the database suffers from approximately 70% of addition, alteration and deletion attacks, our method can achieve above 95% extraction rate of watermark.

The modified method of previous has been proposed by Khanduja et al.[22]. In the proposed scheme the user-specified important attributes are partitioned into cohesive categories and their identifiers are used to prepare and insert the watermark in candidate

attributes. In essence the salient information that is encapsulated in the data can be regenerated afterwards. The contributions of this paper are summarized below:

1. The proposed scheme regenerates crucial information encoded in the data in the event of both illegal alterations in the data as well as deletion of data.

2. The granularity of the recoverable information is decided beforehand by the user. It illustrate the use of unsupervised Machine Learning in discovering salient information contained in the data by using K-means clustering where the number of clusters is user specified.

Kong et al.[23] proposed method to combine ECC with digital watermarking for OWL-based ontology encryption.

Javier et al.[28] proposed a robust lossless relational database watermarking scheme which makes use of circular histogram modulation. It is used to verify database authentication as well as for traceability when identifying database origin after it has been modified. This paper evaluates the performance of its scheme in terms of capacity, distortion, and robustness against two common database modifications: 1) addition and 2) removal of tuples. It models the impact of the embedding process and of database modifications on the probability distribution of the center of mass position.

1.2.2 Elliptic Curve Cryptography

Elliptic Curve Method (ECM) was applied on cryptography known as Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography (PKC). Its security comes from the Elliptic Curve Logarithm, which is the Discrete Logarithm Problem (DLP) in a group defined by points on an elliptic curve over a finite field. These results in a dramatic decrease in key size needed to achieve the same level of security offered in conventional public key cryptography schemes[3].

Elliptic Curve Cryptography (ECC) is a public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key

whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. "Domain parameters" in ECC is an example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography[4].

The mathematical operations of ECC is defined over the elliptic curve $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$. Each value of the "a" and "b" gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters "a" and "b", together with few more constants constitutes the domain parameter of ECC. One main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA[4].

1.3 Problem Statement

There are different watermarking techniques for watermarking different type of data. These techniques were first implemented watermarking into images and then used for audio and video data. There are some technical challenges to implement database watermarking technique because relational data is differ from other type of data (multimedia) as described follows[9]:

- A multimedia data has a large number of redundant bits so it is very easy to hide the watermarking bits into the multimedia data. Whereas in relational database multiple tuples represent a separate objects and it is not easy to insert watermark into these separate objects.
- The position of the different parts does not change in multimedia data whereas tuples have not any ordering relation as collection of tuples called a set of tuples.
- In multimedia data normally it is not easy to drop or replace the part of data without affecting data. But in case of relational database it is a normal procedure like tuple insertion, deletion etc.

So watermarking techniques implemented for other type of the data (multimedia data) cannot directly used for relational database.

1.4 Motivation

Use of database is increasing very rapidly in every application. The Internet is creating many applications where users can access and search the data remotely so data providers are creating the services for users. So it also required some watermarking techniques that can insert watermark into the database without changing the meaning and value of any field of the database. It is the biggest challenge of database watermarking techniques to insert a watermark into a database without changing the value of any attribute. Data providers are demanding for the database watermarking technologies to identify the pirated copies of their databases.

The motivation of this project is to combine the ECC with digital database watermarking to enhance security of relational database. Because of the much smaller key sizes involved, ECC algorithms can be implemented for Relational Database Watermarking

1.5 Project Objective and Scope

Due to increasing popularity of Internet, data piracy has become a big issue where data owners are more concern for their digital assets. To protect these digital assets, a digital watermark can be inserted in to the data, where data can be any software, image, video, audio or text.

This report proposed a method to implement a new technique for digital database watermarking for Relational Database using Elliptic Curve Cryptography (ECC). The objectives and scope of this project are as follows:

1 To implement an efficient technique for relational database watermarking, which justify that some bit position of some of the numbers of some of the rows contain some specific value. The bit location and watermarking information will be determined using ECC generated keys that will be accessible only to the owner. Watermark extraction method will not require original data as well as

watermark. Watermark will be efficiently maintained after insertion, deletion and updation also.

2 To proof that the propose technique is robust against different attacks as described in previous section.

3 To implements watermark insertion and extraction algorithm that perform with real-world applications.

4 To compare and contrast the propose technique with already implemented techniques for digital database watermarking and show the results.

1.6 Thesis Organization

This report consists of 5 chapters and these chapters are organized as follows:

- **Chapter 1 Introduction:** This chapter presents the problems of database watermarking and defines the digital watermarking. It justifies the aim and scope of this dissertation. It also highlights the significant contributions from the investigation. It introduces the topic to the reader, provide an overview of previous research on the topic, and identify the own hypothesis.
- **Chapter 2 Literature Review:** This Chapter presents a critical appraisal of the previous work published in the literature pertaining to the concept of Digital Database Watermarking and Elliptic Curve Cryptography (ECC). It also describes and analyzes the different type of watermarking algorithms.
- **Chapter 3 Proposed System Design:** This chapter gives the proposed architectures of Watermark Embedding module and Watermark Extraction module. This chapter also includes the details of the algorithms and approaches which are to be used in the development of the watermark insertion and watermark extraction.
- **Chapter 4 Results and Analysis:** This chapter gives the resulted system performance for different type of possible attacks in database through chart and table. It also justifies the performance of the proposed system.
- **Chapter 5 Conclusion and Future Work:** This chapter describes results and concludes the report by describing various observations and scope of future work. It gives an insight into the performance of the system. It also tells about possible future enhancements.

CHAPTER 2

LITERATURE REVIEW

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[25].

The importance of digital watermarking for digital assets such as relational databases to preserve their copyrights is becoming more and more important as time goes by. In the past few years, a large number of techniques have been proposed for hiding copyright marks specifically on relational databases. Some of these techniques are discussed in this chapter. There is a tremendous work has been done in the area of Database watermarking as well as Elliptic Curve Cryptography. Some of these techniques are discussed in this chapter.

2.1 Digital Database Watermarking

Generally database watermarking technique has two phases: Watermarking insertion and Watermark extraction. Now watermarked data can be publicly available for other users. Other users even do not know that database has any watermark. In case of any attack, like ownership proof watermark extraction phase is performed which take user's private key and apply it on suspicious database and extract the inserted watermark and compare it with user's original watermark information.

Relational data is differ from other type of data (multimedia) as described follows[9]:

(i) **Few Redundant Data:** Any type of data has a large number of bits so it is very easy to hide the watermarking bits into the multimedia data. Whereas in relational database multiple tuples represent a separate objects and it is not easy to insert watermark into these separate objects

(ii) **Out-of-Order Relational Data:** The position of the different parts does not change in multimedia data whereas tuples have not any ordering relation as collection of tuples called a set of tuples.

(iii) **Frequent Updating:** In multimedia data normally it is not easy to drop or replace the part of data without affecting data. But in case of relational database it is a normal procedure like tuple insertion, deletion etc.

(iv) **Psycho-physical phenomena:** There are many psycho-physical phenomena based on human visual system and human auditory system which can be exploited for mark embedding. However, one can not 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. So watermarking techniques implemented for other type of data cannot directly used for relational database.

2.1.1 Applications of Digital Watermark for Relational Databases

Relational database watermarking is useful in many applications as described follows:

(i) **Ownership Assertion:** To insert an ownership in to the database, database watermarking is used. A private key, known to the user is used to insert a watermark into the user's original database. Other users even do not know that database has any watermark. For ownership proof watermarking technique will take user's private key and apply it on suspicious database and extract the inserted watermark and compare it with user's original watermark information.

(ii) **Fingerprinting:** In some application database content is publicly available, but owner does not want to unauthorized distribution and duplication of the original content. Then watermarking can determine the unauthorized copies of database by retrieving the fingerprint.

(iii) **Fraud and Tamper Detection:** When database content is used for very critical applications such as commercial transactions or medical applications, it is important to ensure that the content was originated from a specific source and that it had not been changed, manipulated or falsified. This can be achieved by embedding a watermark in the underlying data of the database. Subsequently, when the database is checked, the watermark is extracted using a unique key associated with the source, and the integrity of the data is verified through the integrity of the extracted watermark.

2.1.2 Different Types of Attacks

Generally, the digital watermarking for integrity verification is called fragile watermarking as compared to robust watermarking for copyright protection. In a robust watermarking scheme, the embedded watermark should be robust against various attacks which aim at removing or distorting the watermark. While in a fragile watermarking scheme, the embedded watermark should be fragile to modifications so as to detect and localize any modification in presence of different attacks.

The watermarked database may suffer from various types of intentional and unintentional attacks which may damage or erase the watermark, as described below:

1. **Benign Update:** In this case, the tuples or data of any watermarked relation are processed as usual. As a result, the marked tuples may be added, deleted or updated which may remove the embedded watermark or may cause the embedded watermark undetectable (for instance, during update operation some marked bits of marked data can be erroneously flipped). This type of processing are performed unintentionally.
2. **Value Modification Attack:**

    - **Bit Attack:** This attack attempts to destroy the watermark by altering one or more bits in the watermarked data. More information about the marked bit position makes attack more successful. However, in this case usefulness of data is crucial: more alternation may result the data completely useless.

Bit attack may be performed randomly which is known as *Randomization Attack* by assigning random values to certain bit positions; or by *Zero Out Attack* where the values in the bit positions are set to zero; or may be performed by inverting the values of the bit positions, known as *Bit Flipping Attack.*

- **Rounding Attack:** Attacker may try to lose the marks contained in a numeric attribute by rounding all the values of the attribute. Success of this attack depends on the estimation of how many bit positions are involved in the watermarking. Underestimation of it may cause the attack unsuccessful, whereas overestimation may cause the data useless.
- **Transformation:** An attack related to the rounding attack is one in which the numeric values are linearly transformed. For example, attacker may convert the data to a different unit of measurement (e.g., Fahrenheit to Celsius). The unnecessary conversion by attacker would raise suspicion among users.

3. **Subset Attack:** Attacker may consider a subset of the tuples or attributes of a watermarked relation and by attacking (deleting or updating) on them he may hope that the watermark has been lost.
4. **Superset Attack:** Some new tuples or attributes are added to a watermarked database which can affect the correct detection of the watermark.
5. **Collusion Attack:** This attack requires the attacker to have access to multiple fingerprinted copies of the same relation.

    - **Mix-and-Match Attack:** Attacker may create his relation by taking disjoint tuples from multiple relations containing similar information.
    - **Majority Attack:** This attack creates a new relation with the same schema as the copies but with each bit value computed as the majority function of the corresponding bit values in all copies so that the owner can not detect the watermark.

6. **False Claim of Ownership:** This type of attack seeks to provide a traitor or pirate with evidence that raises doubts about merchant's claim.

- **Additive Attack:** Attacker may simply add his watermark to user's watermarked relation and try to claim his ownership.
- **Invertibility Attack:** Attavker may launch an invertibility attack to claim his ownership if he can successfully discover a fictitious watermark which is in fact a random occurrence from a watermarked database.

7. **Subset Reverse Order Attack:** Attacker enjoys this attack by exchanging the order or positions of the tuples or attributes in relation which may erase or disturb the watermark.
8. **Brute Force Attack:** In this case, attacker tries to guess about the private parameters (*e.g.* secret key) by traversing the possible search spaces of the parameters. This attack can be thwarted by assuming that the private parameters are long enough in size.

2.1.3 Watermarking Issues

The important issues that arise in the study of digital watermarking techniques for relational databases are:

- **Capacity:** It determines the optimum amount of data that can be embedded in a cover and the optimum way to embed and extract this information.
- **Usability:** The changes in the data of the database during watermarking process should not degrade the usability of the data. The amount of allowable change differs from one database to another, depending on the nature of stored records.
- **Robustness:** Watermarks embedded in databases should be robust against malicious or accidental attempts at removal without destroying the usability of the database.
- **Security:** The security of the watermarking process relies on some private parameters (*e.g.* secret key) which should be kept completely secret. Owner of the database should be the only one who has knowledge about them.
- **Blindness:** Watermark extraction should require neither the knowledge of the original unwatermarked database nor the watermark information. This property

is critical as it allows the watermark to be detected in a copy of the database relation, irrespective of later updates to the original relation.

- **Incremental Watermarking:** After a database has been watermarked, the watermarking algorithm should compute the watermark values only for the added or modified tuples, keeping the unaltered watermarked tuples untouched.
- **Non-interference:** If multiple marks are inserted into a single relational database, then they should not interfere with each other.
- **Public System:** The watermarking system should assume that the method used for inserting a watermark is public. Defense must lie only in the choice of the private parameters (*e.g.* secret key).
- **False Positiveness and False Negativeness:** The false hit is the probability of a valid watermark being detected from unwatermark data, whereas false miss is the probability of not detecting a valid watermark from watermarked data that has been modified in typical attacks. The false hit and false miss should be negligible.

2.1.4 Classification of Watermarking Techniques

The watermarking techniques proposed so far can be classified along various dimensions as follows:

- **Watermark Information:** Different watermarking schemes embed different types of watermark information (e.g. image, text etc.) into the underlying data of the database.
- **Distortion:** Watermarking schemes may be distortion-based or distortion free depending on whether the marking introduces any distortion to the underlying data.
- **Cover Type:** Watermarking schemes can be classified based on the type of the cover (e.g. type of attributes) into which marks are embedded.
- **Granularity Level:** The watermarking can be performed by modifying or inserting information at bit level or higher level (e.g. character level or attribute level or tuple level).
- **Verifiability/Detectability:** The detection/verification process may be deterministic or probabilistic in nature, it can be performed blindly or non-

blindly, it can be performed publicly (by anyone) or privately (by the owner only).

- **Intent of Marking:** Different watermarking schemes are designed to serve different purposes, namely, integrity and tamper detection, localization, ownership proof, traitor detection etc.

2.2 Digital Database Watermarking Techniques

The database watermarking techniques can be categorized according to the following three categories[26]:

(i) Distortion Based: Whether marking introduces any distortion, the watermarking techniques in this category introduce small changes in the underlying data of the database during embedding phase but the degree of change should be tolerable and should not make the data useless.

(ii) Type of Attribute: The type of the underlying data (cover) in which watermark information is embedded. These techniques applies watermark on numerical and non-numeric data type.

(iii) Distortion Free: The watermark insertion phase does not depend on any specific type of attribute and does not introduce any distortion in the underlying data of the database

The different database watermarking techniques has been categorized as Distortion Based and Distortion Free. Further both types have been again categorized as type of attribute for watermark information.

2.2.1 Distortion-based Watermarking

The watermarking techniques in this category introduce small changes in the underlying data of the database during embedding phase. The degree of changes should be such that any changes made in the data are tolerable and should not make the data useless. The watermarking can be performed at bit level, or character level, or higher such as attribute or tuple level, over the attribute values of types numeric, string, categorical, or any.

2.2.1.1 Watermarking Based on Numerical Data Type Attribute

There are some techniques for watermarking that are based on numeric data type attribute and marking is done as the following methods.

a. Arbitrary meaningless bit pattern as watermark information:

The watermarking schemes proposed by Agrawal et al. [9, 10] (also known as AHK algorithm) is based on numeric data type attribute and marking is done at bit-level. The basic idea of these schemes is to ensure that some bit positions for some of the attributes of some of the tuples in the relation contain specific values. This bit pattern constitutes the watermark. The tuples, attributes within a tuple, bit positions in an attribute, and specific bit values at those positions are algorithmically determined under the control of the private parameters γ, ν, ξ and K known only to the owner of the relation. The parameters γ, ν, ξ and K represent number of tuples to mark, number of attributes available to mark, number of least significant bits available for marking in an attribute, and secret key respectively.

In [9], the cryptographic MAC function H(K||H(K||r.P)) where r.P is the primary key of the tuple r and || represents concatenation operation, is used to determine candidate bit positions. The HASH function H(K||r.P) is used to determine the bit values to be embedded at those positions. The choice of MAC and HASH is due to the one-way functional characteristics and less collision probability.

In [10], authors use pseudorandom sequence generator instead of HASH and MAC to identify the marking bits and mark positions. The security and robustness of this scheme relies on these parameters which are completely private to the owner.

The watermark detection algorithm is blind and probabilistic in nature. The relation is considered as pirated if the matching pattern is present in at least τ tuples, where τ depends on the actual number of tuples marked and a preselected value α, called the significance level of the test. Observe that the success of watermark detection phase depends on the fixed order of attributes. Re-sort of attributes' order may yield to the detection phase almost infeasible. Although the main assumption of this scheme is

that the relation has primary key whose value does not change, they also suggest an alternative to treat a relation without primary key.

b. Image as watermark information:

Rao et al.[7] proposed a technique which is based on inserting the bits of a binary image in relational database. Initially the watermark image is transformed into a binary matrix and then the secret key and the primary key of the tuple is considered together. At the next step its hash value is computed by the use of MD5 algorithm. The computed hash value and value of F determine whether the tuple will be marked or not. After that, the attribute index (i) and bit index (j) is determined where attribute index specifies that particular attribute is selected for watermarking amongst all available numerical attributes. Then the chosen bits of an image replace some bits of the selected attributes of particular tuple. The proposed technique is irrespective to the tuples' order and also minimizes the variation in watermarked database.

Wang et al.[13] described an image-based watermarking scheme where instead of embedding original image as watermark, an scrambled image based on Arnold transform with scrambling number d is used. Since Arnold transform of an image has the periodicity P, the result which is obtained in the extraction phase can be recovered from the scrambled form to the original after (P − d) iterations. In the embedding phase, the original image of size N × N is first converted into scrambled image which is then represented by a binary string b_s of length L = N × N. Secondly, all tuples in the relation are grouped into L groups. The hash value which is computed using tuple's primary key, secret key and order of the image, determines the group in which each tuple belongs. Finally, the i^{th} bit of b_s is embedded into the algorithmically chosen bit position of the attribute value for those tuples in i^{th} group that satisfy a particular criteria. The detection phase follows majority voting technique. However, the security of this scheme improves as it relies not only on the secret key but also the scrambling number d and the order of the image N.

Rather than embedding scrambled image, the watermarking technique in [Hu et al., 2009][14] embeds the original image by first converting it into a bit flow (EMC, Encrypted Mark Code) of certain length, and then by following similar algorithmic

steps as in [Wang et al., 2008][13]. The only two differences are that (i) the watermark insertion technique in [Wang et al., 2008] assumes single fixed attribute to mark for all tuples whereas [Hu et al., 2009] does not, and (ii) during selection of bit positions, the order of the image is not considered in [Hu et al., 2009]. Finally, after marking, [Hu et al., 2009] checks the usability of the data with respect to the intended use. If acceptable, the change is committed, otherwise rolled back.

c. Content characteristics as watermark information:

In [Zhang et al., 2006][11], the watermark insertion phase extracts some bits, called local characteristic, from the characteristic attribute A1 of tuple t and embeds those bits into the watermark attribute A2 of the same tuple. The selection of tuples depends on whether the generated random value (between 0 and 1) is less than the embedded proportion α of the relational databases and the nonNULL requirement of characteristic attribute value. In the watermark detection phase, by following similar procedure, the local characteristic of the characteristic attribute are extracted and compared against the last bits of watermark attribute.

The partitioning of tuples in most of the techniques is based on hashing. Huang et al.[20], proposed the use of well-known techniques (e.g. k-means algorithm) to cluster the tuples into some equivalent classes. The embedding of the watermark bit is based on the comparison of the parity of watermark bit and the LSB of candidate attribute. The k-means method assures the location of the embedded watermark irregular.

The modified method of previous has been proposed by Khanduja et al.[22]. This paper proposed a new technique for multiple watermarking of relational databases that provides solutions to two major security concerns; ownership identification and information recovery. In order to resolve ownership conflicts a secure watermark is embedded using a secret key known only to the database owner. Another watermark encapsulates granular information on user-specified crucial attributes in a manner such that the perturbed or lost data can be regenerated conveniently later. The probability of successful regeneration of tampered/lost data improves dramatically as increase the number of candidate attributes for embedding the watermark.

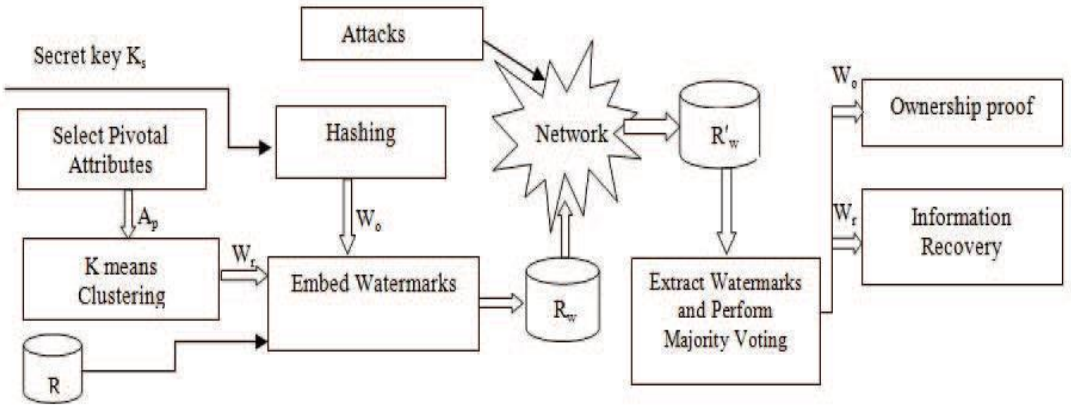

Figure 2.1: The Proposed Watermarking Model of Khanduja et al.[22]

In the proposed scheme, as shown in figure 2.1, the user-specified important attributes are partitioned into cohesive categories and their identifiers are used to prepare and insert the watermark in candidate attributes. In essence the salient information that is encapsulated in the data can be regenerated afterwards. The contributions of this paper are summarized below:

1. The proposed scheme regenerates crucial information encoded in the data in the event of both illegal alterations in the data as well as deletion of data.
2. The granularity of the recoverable information is decided beforehand by the user. It illustrate the use of unsupervised Machine Learning in discovering salient information contained in the data by using K-means clustering where the number of clusters is user specified.
3. This scheme combines proof of ownership along with data recovery within the same watermarking framework by embedding multiple watermarks.

Ying Wang et al.[29] proposed the watermarking technique which embedded by the Arnold transforming and scrambling technology and modifying the parity of the low decimal number of numeric attribute, connected with some attribute in the physical storage space in the relational databases. This algorithm may achieve the purpose of copyright protection of database effectively because of the strong robustness.

d. Genetic Algorithm based Watermarking

Saman Iftikhar et al.[30] introduced a Robust and Reversible Watermarking (RRW) technique for relational database watermarking. RRW mainly comprises a (1) data

preprocessing phase, (2) watermark encoding phase, (3) attacker channel, (4) watermark decoding phase and (5) data recovery phase. In data preprocessing phase, secret parameters are defined and strategies are used to analyze 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 a, the optimized value from the GA and r, a change matrix are used in the watermark encoding and decoding phases. 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 preprocessing step is performed again, and decoding strategies (feature selection on the basis of MI, the optimized value from the GA and r 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.

e. Watermarking based on Cryptographic Hashing Function

Deshpande et al.[18] proposed mechanism of watermarking relational database mainly involve decoding and encoding on numerical attribute of relational database. The first phase is to partition the original data and assign partition number to each and every tuple of the relation using Cryptographic Hashing Function (MD5). In the second phase , while changing the data , select the desired watermark and bit bi is selected from the partitioned data and then that bit bi is changed using watermark W. When the original value of data gets changed due to the watermark bit, it always checks the data usability constraints. In the third phase, after inserting the watermark in the partition, merge all partitions and get the complete watermarked data. While decoding, use majority voting algorithm to get the correct watermark. Detecting the watermark neither requires access to neither the original data nor the watermark.

The watermark can be detected even in a small subset of a watermarked relation as long as the sample contains some of the marks.

2.2.1.2 Watermarking Based on Non-Numerical Data Type Attribute

There are some techniques for watermarking that are based on non-numeric data type attribute and marking is done as the following methods.

a. Space based Watermarking on Non-Numeric Multi Word Attributes:

Al-Haj, A. and Odeh, A.[28] proposed a watermarking scheme which is based on hiding binary image in spaces of non-numeric multi-word attributes of subsets of tuples, instead of numeric attribute at bit-level. The watermark is divided into m string each containing n bits. On the other hand, the database is also divided into non-intersecting subsets each containing m tuples. The m short strings of the watermark image are embedded into each m-tuple subset. The embedding is done as follows: suppose the integer representation of the ith, i € [1 ...m], short string is di. A double space is created after di words of the pre-selected nonnumeric, multi-word attribute of i^{th} tuple in the subset. The extraction phase counts the number single spaces appearing before double space which indicates the decimal equivalent of the embedded short binary string. Since the proposed algorithm embeds the same watermark for all non-intersecting subsets of the database, it is robust against subset deletion, subset addition, subset alteration and subset selection attacks. Another advantage for space- based watermarking is that large bit-capacity available for hiding the watermark which may also facilitate embedding of multiple small watermarks. However, it may suffer from watermark removal attack if attacker replaces all double spaces between two words (if exist) by single space for all tuples in the relation.

b. Text format based relational database watermarking for non-numeric data

Hanyurwimfura, Damien Yuling Liu, Zhijie Liu [27] proposed a new relational database watermarking method for non-numeric multi words data. A mark is embedded by horizontally shifting the location of a word within selected attribute of selected tuples; a word is displaced right or left unmoved depending on watermark

bit, the Location where the mark to be inserted is determined by the Levenshtein Distance between two successive words within an attribute. This method is effective as it is robust against different forms of malicious attacks and it is blind as it does not require the original database in order to extract the embedded watermark.

2.2.1.3 Watermarking Based on Fake Tuple insertion

Pournaghshband[31] presented an effective watermarking technique for relational data that is robust against various attacks. Its approach inserts new tuples that are not real called "fake" tuples, to the relation as watermarks.

2.2.2 Distortion-Free Watermarking Techniques

Most of the distortion free watermarking techniques are fragile in the sense that in addition to the ownership claiming, they aim at maintaining the integrity of the information in the database. The watermark insertion phase does not depend on any specific type of attribute and does not introduce any distortion in the underlying data of the database.

2.2.2.1 Watermarking Based on Circular Histogram Modulation

Javier et al.[28] proposed a robust lossless relational database watermarking scheme which makes use of circular histogram modulation. The resulting scheme modulates the relative angular position of the circular histogram center of mass of one numerical attribute for message embedding. It can be used for verifying database authentication as well as for traceability when identifying database origin after it has been modified. Beyond the application framework, this paper theoretically evaluates the performance of its scheme in terms of capacity, distortion, and robustness against two common database modifications: 1) addition and 2) removal of tuples. It models the impact of the embedding process and of database modifications on the probability distribution of the center of mass position.

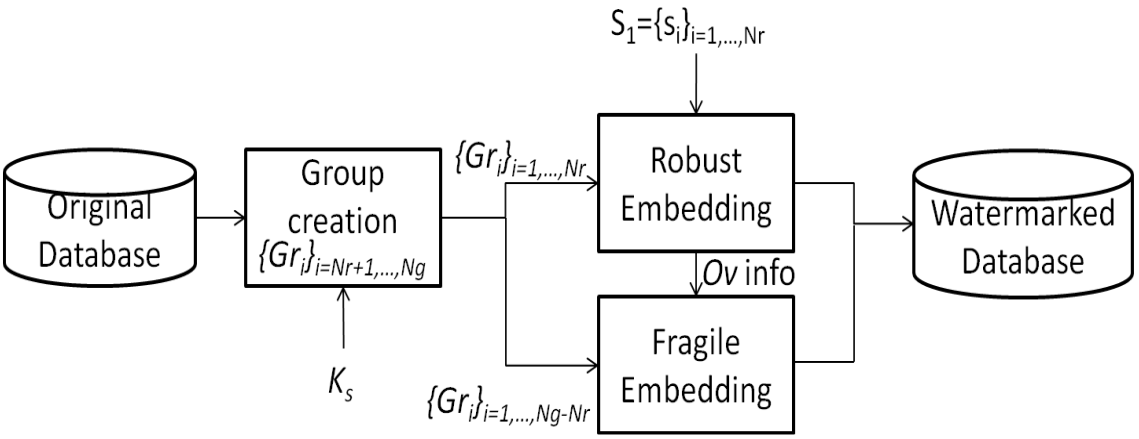

Figure 2.2: Proposed robust scheme of Javier et al.[28]

The proposed scheme is shown in figure 2.2. This scheme consists the embedding of a sequence of bits such as $S = \{s_i\}_{i=1,...,Nc}$, $s_i \in \{0, 1\}$, where N_c is the number of available carriers. This sequence includes the message m_s the user wants to insert along with the overhead O_v if necessary. At the detection, the sequence of bits S is extracted directly from the carrier groups. In an applicative context, m_s may correspond to the digital signature of the database. At the reception, the recipient just has to compare the extracted signature to the one recomputed from the restored database so as to decide about the database integrity.

S_1 corresponds to a fixed length pseudo-random sequence of Nr symbols such as: $S_1 = \{s_{1j}\}$ j=1,...,N_r, with $s_{1j} \in \{-1,+1\}$. S_1 is inserted into the N_r first groups of tuples. The second sequence S_2 is inserted into the other $N_g - N_r$ groups of tuples. It contains a sequence of bits which encodes the overhead O_v info required for reconstructing the whole database (O_v info indicates also overflow-groups into the first Nr groups).

2.2.2.2 Cluster-Based Watermarking Technique

Kaiyin Huang et al.[20] proposed a cluster-based database watermarking technique for relational database which first applies cluster theory to the database watermarking technology. The cluster theory is used to partitions subset through clustering the data in the source data and the clustering results determine the quantity of embedded watermark information and embedded position. This method makes the watermark information more disperse and hidden. This paper also introduces an odd-even modifying method which assures the minimum modification to original database.

2.2.2.3 Watermarking for joint ownership

Yu Fu et al.[17] proposed a technique to break the main secret into multiple parts and hide them individually in a relational database. The (k, n) threshold scheme is used to decide the minimum number of parts required to recover the secret data completely.

2.3 Elliptic Curve Cryptography

It is impossible to fix the key length in RSA cryptosystem. Due to the recent development in field of security, the requirement for key length for secure RSA has increased. The increment in the length of key increase the security of the RSA cryptosystem but it also increase the computational and communicational cost as well. So it is not easy to handle extra cost for those commercial application which uses large number of secure transactions. So in recent a new public key cryptosystem has challenged the RSA cryptosystem.

Elliptic Curve Method (ECM) was applied on cryptography known as Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography (PKC). Its security comes from the Elliptic Curve Logarithm, which is the Discrete Logarithm Problem (DLP) in a group defined by points on an elliptic curve over a finite field. These results in a dramatic decrease in key size needed to achieve the same level of security offered in conventional public key cryptography schemes[3].

Elliptic Curve Cryptosystem (ECC) is a type of public key cryptography. In public key cryptography each user has a pair of keys, from which one key is works as a private key and another key is works as a public key. Public key is distributed to all other users. Private Key is kept with the user only. If any user wants to communicate with that user in secure manner, a message is encrypted through the public key of that user and that user will decrypt it through the private key which is known only to that user. Public-private keys are related in such manner that if any key is used for encryption; another key of the pair can only use for decryption.

Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. “Domain parameters” in ECC is an

example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography[4].

The mathematical operations of ECC is defined over the elliptic curve $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$. Each value of the "a" and "b" gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters "a" and "b", together with few more constants constitutes the domain parameter of ECC. One main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA[4].

In constrained environments such as mobile phones, wireless pagers or personal digital assistant (PDA), the resources like bandwidth, memory and battery life are highly limited. Thus, a suitable public key scheme would be one that is efficient in terms of computing costs and key sizes. The ECC has the highest strength-per-bit compared to other public key cryptosystems. Small key sizes translate into savings in bandwidth, memory and processing power. This makes ECC the obvious choice in this situation[5].

2.3.1 Elliptic Curve over F_q

An elliptic curve over F_q is defined in terms of the solutions to an equation in F_q. The form of the equation defining an elliptic curve over F_q differs depending on whether the field is a prime finite field or a characteristic 2 finite field. This report describes only elliptic curve of prime finite field.

2.3.1.1 Elliptic Curves over F_p

Let F_p be a prime finite field so that p is an odd prime number, and let a; b ϵ F_p satisfy $4a^3 + 27b^2 \neq 0 \pmod p$. Then an elliptic curve $E(F_p)$ over F_p defined by the parameters a; b ϵ F_p consists of the set of solutions or points P = (x; y) for x; y ϵ F_p to the equation:

$$y^2 \equiv x^3 + ax + b \pmod p$$

together with an extra point O called the point at infinity. The equation $y^2 \equiv x^3 + ax + b \pmod p$ is called the defining equation of $E(F_p)$. For a given point $P = (x_P; y_P)$, x_P is called the x-coordinate of P, and y_P is called the y-coordinate of P[3].

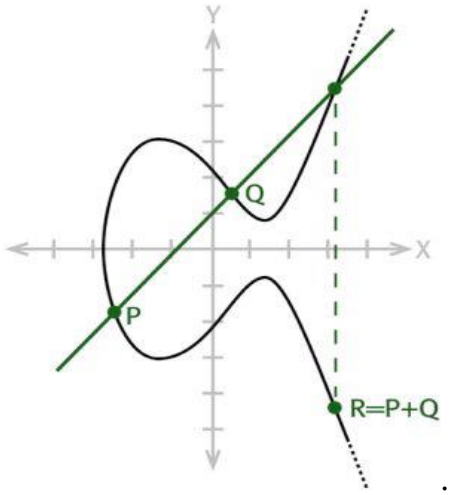

Fig 2.3: Point Addition[7]

To add two distinct points P and Q on an elliptic curve, draw a straight line between them. The line will intersect the elliptic curve at exactly one more point –R. The reflection of the point –R with respect to x-axis gives the point R, which is the result of addition of points R and Q as shown in figure 2.3[7].

It is possible to define an addition rule to add points on R. The addition rule is specified as follows[3]:

1. Rule to add the point at infinity to itself:
   $O + O = O$
2. Rule to add the point at infinity to any other point:
   $(x; y) + O = O + (x; y) = (x; y)$ for all $(x; y) \epsilon E(F_p)$
3. Rule to add two points with the same x-coordinates when the points are either distinct or have y-coordinate 0:
   $(x; y) + (x; -y) = O$ for all $(x; y) \epsilon E(F_p)$
4. Rule to add two points with different x-coordinates: Let $(x_1; y_1) \epsilon E(F_p)$ and $(x_2; y_2) \epsilon E(F_p)$ be two points such that $x_1 \neq x_2$. Then $(x_1; y_1) + (x_2; y_2) = (x_3; y_3)$, where:
   $x_3 \equiv \lambda^2 - x_1 - x_2 \pmod p$, $y_3 \equiv \lambda(x_1 – x_3) – y_1 \pmod p$ and $\lambda \equiv (y_2-y_1)/(x_2-x_1) \pmod p$

5. Rule to add a point to itself (double a point): Let $(x_1; y_1) \epsilon E(F_p)$ be a point with $y_1 \neq 0$. Then $(x_1; y_1)+(x_1; y_1)=(x_3; y_3)$, where:
   $x_3 \equiv \lambda^2 - 2*x_1 \pmod p$, $y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod p$ and $\lambda \equiv (3*x_1^2+a)/(2*y_1) \pmod p$

The set of points on $E(F_p)$ forms a group under this addition rule. Furthermore the group is abelian - meaning that $P_1+ P_2 = P_2+ P_1$ for all points $P_1; P_2 \epsilon E(F_p)$. Notice that the addition rule can always be computed efficiently using simple field arithmetic.

Cryptographic schemes based on ECC rely on scalar multiplication of elliptic curve points. Given an integer k and a point $P \epsilon E(F_p)$, scalar multiplication is the process of adding P to itself k times. The result of this scalar multiplication is denoted k*P or k_P. Scalar multiplication of elliptic curve points can be computed efficiently using the addition rule together with the double-and-add algorithm or one of its variants.

2.3.1.2 Double-and-Add algorithm[7]

Input: d=(dt-1, dt-2,…d0) , PεE.

1. Q←O
2. For i from 0 to t-1 do
   i. If d_i=1 then Q←Q+P
   ii. P←2P
3. Output: d_P=Q

2.3.1.3 Elliptic Curve Public-Key Pairs

Given a set of domain parameters that include a choice of base field prime p, an elliptic curve E/Fp, and a base point G of order n on E, an elliptic curve key pair (d, Q) consists of a private key d, which is a randomly selected non-zero integer modulo the group order n, and a public key Q = dG, the d-multiple of the base point G. Thus the point Q is a randomly selected point in the group generated by G[1].

To generate the public and private keys in the ECC, the user picks a large prime number p and elliptic curve parameters a and b for equation $y^2 \equiv x^3 + ax + b \pmod p$. Next user

choose a point G whose order is very large value n. This point G is called base point and the order of G means, nG=O such that n is the smallest positive integer. The parameters are common to all users.

Any user A picks $n_A < n$ and computes $P_A = n_A * G$. The number n_A is the private key and P_A is the public key of the user A.

2.3.1.4 Key Exchange in ECC

Consider two users A and B want to exchange their keys. The key exchange depends on the following steps:

1. A sends P_A to B
2. B calculates $K = n_B * P_A = n_B * (n_A * G)$
3. B sends P_B to A
4. A calculates $K = n_A * P_B = n_A * (n_B * G)$

Thus they can share the key K.

2.3.1.5 Encryption and Decryption in ECC

ECC can be used for encryption and decryption. Consider the user A wants to encrypt a message m for the user B, then the following steps are included:

1. A encodes the message m as $P_m = (x,y)$
2. A choose a random number k and produce the ciphertext $C_m = [k * G, P_m + kP_B]$ and sends this cipher text to B
3. B computes $n_B * k * G$
4. B again compute $P_m + k * P_B – n_B * k * G = P_m – k (n_B * G) + kP_B = P_m – kn_B + kn_B = P_m$

The security of ECC depends on the difficulty of finding k for the given P and kP. Pollard rho method is an available technique to get the elliptic curve logarithm very fast. A considerable small size can be used for ECC as compared to RSA.

2.3.2 Applications of ECC

There are different applications of ECC. Some of them are as follows[4][3].

2.3.2.1 ECDSA - Elliptic Curve Digital Signature Algorithm

For authenticating a device or a message sent by the device, Signature algorithm is used. For example consider two devices A and B. To authenticate a message sent by A, the device A signs the message using its private key. The device sends the message and the signature to the device B. This signature can be verified only by using the public key of device A. Since the device B knows A's public key, it can verify whether the message is indeed send by A or not. ECDSA is a variant of the Digital Signature Algorithm (DSA) that operates on elliptic curve groups. For sending a signed message from A to B, both have to agree up on Elliptic Curve domain parameters. The domain parameters are defined in section Elliptic Curve Domain parameters. Sender 'A' have a key pair consisting of a private key d_A (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key $Q_A = d_A*G$ (G is the generator point, an elliptic curve domain parameter). An overview of ECDSA process is defined below.

Signature Generation: For signing a message m by sender A, using A's private key d_A

1 Calculate e = HASH (m), where HASH is a cryptographic hash function, such as SHA-1
2 Select a random integer k from [1, n - 1]
3 Calculate $r = x_1 \pmod n$, where $(x_1, y_1) = k * G$. If r = 0, go to step 2
4 Calculate $s = k\text{-}1(e + d_{Ar}) \pmod n$. If s = 0, go to step 2
5 The signature is the pair (r, s)

Signature Verification: For B to authenticate A's signature, B must have A's public key Q_A

1 Verify that r and s are integers in [1, n - 1]. If not, the signature is invalid
2 Calculate e = HASH (m), where HASH is the same function used in the signature generation
3 Calculate $w = s\text{-}1 \pmod n$

4 Calculate $u_1 = e_w \pmod n$ and $u2 = r_w \pmod n$

5 Calculate $(x_1, y_1) = u_1G + u_2Q_A$

6 The signature is valid if $x_1 = r \pmod n$, invalid otherwise

2.3.2.2 ECDH - Elliptic Curve Diffie Hellman

ECDH is a key agreement protocol that allows two parties to establish a shared secret key that can be used for private key algorithms. Both parties exchange some public information to each other. Using this public data and their own private data these parties calculates the shared secret. Any third party, who doesn't have access to the private details of each device, will not be able to calculate the shared secret from the available public information. An overview of ECDH process is defined below. For generating a shared secret between A and B using ECDH, both have to agree up on Elliptic Curve domain parameters. The domain parameters are defined in section Elliptic Curve Domain parameters. Both end have a key pair consisting of a private key d (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key Q = d * G (G is the generator point, an elliptic curve domain parameter). Let (d_A, Q_A) be the private key - public key pair of A and (d_B, Q_B) be the private key - public key pair of B.

1 The end A computes $K = (x_K, y_K) = d_A * Q_B$

2 The end B computes $L = (x_L, y_L) = d_B * Q_A$

3 Since $d_AQ_B = d_Ad_BG = d_Bd_AG = d_BQ_A$. Therefore K = L and hence $x_K = x_L$

4 Hence the shared secret is x_K

Since it is practically impossible to find the private key d_A or d_B from the public key K or L, it is not possible to obtain the shared secret for a third party.

2.3.2.3 Elliptic Curve Integrated Encryption System (ECIES)

Integrated Encryption Scheme (IES) is a hybrid encryption scheme which provides semantic security against an adversary who is allowed to use chosen-plaintext and chosen-ciphertext attacks. The security of the scheme is based on the Diffie–Hellman problem. The elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based on encryption algorithm. It is called integrated, since it is a hybrid

scheme that uses a public key system to transport a session key for use by a symmetric cipher. ECIES is a public-key encryption algorithm.

To send an encrypted message to receiver using ECIES sender needs the following information:

- Cryptographic suite to be used:
- EC domain parameters (p, a, b, G, n, h) for a curve over prime field or (m, f(x), a, b, G, n, h) for a curve over binary field;
- Receiver's public key: K_B(Receiver generates it as follows: $K_B = K_{BG}$, where K_B is the private key he chooses at random: K_B € [1, n-1]
- Optional shared information: S_1 and S_2.

To encrypt a message m Alice does the following:

1. Generates a random number r € [1,n-1] and calculates R = rG;
2. Derives a shared secret: $S = P_x$, where $P = (P_x, P_y) = rK_B$ (and $P \neq 0$)
3. Uses KDF to derive a symmetric encryption and a MAC keys: $K_E \,||\, K_M = KDF(S||S_1)$;
4. Encrypts the message: $c = E(K_E; m)$;
5. Computes the tag of encrypted message and S_2: $d = MAC (K_M; c||S_2)$; outputs R||c||d.

To decrypt the ciphertext R||c||d Bob does the following:

1 Derives the shared secret: $S = P_x$, where $P = (P_x, P_y) = KBR$ (it is the same as the one Alice derived because $P = K_BR = K_BrG = rK_BG = rK_B$), or outputs failed if P = 0;

2 Derives keys the same way as sender did: $K_E||K_M = KDF(S||S_1)$;

3 Uses MAC to check the tag and outputs failed if $d \neq MAC(K_M ; c||S_2)$;

4 Uses symmetric encryption scheme to decrypt the message $m = E^{-1}(K_E; c)$

Elliptic curve cryptography (ECC) serves as an excellent candidate because of its small key size and high security protection. Digital watermark is the practice of hiding a message within that work itself. In this report, a novel method is proposed to combine elliptic curve cryptography with digital watermark for database.

This chapter reviewed different techniques that were based on relational database watermarking. Every author worked for the robustness of the technique. Many watermarking techniques are based on different watermark information; most of these techniques are designed for numerical database and are distortion based. There are almost similar steps to identify attribute then tuple and then marking position for the watermark. Most of these techniques used a single attribute of a tuple to embed a watermark. So, this work will be extended towards embedding the watermarks at different attributes at different places. Therefore, it will be difficult for attacker to remove watermarks from different places from the database. Most of these techniques are also depend on presence of primary key.

CHAPTER 3

PROPOSED SYSTEM DESIGN

In this chapter a technique has been proposed for relational database watermarking that satisfies all the requirements discussed in previous chapter. Proposed technique marks only numeric attributes and assumes that the marked attributes are such that small changes in some of their values are acceptable and nonobvious. All of the numeric attributes of all the tuples are not marked. The data owner and primary key of the data is able to decide that which attributes are suitable for marking.

Elliptic curve cryptography (ECC) encryption is used to create watermark where user create its own watermarking bits using its public key. A secret key is used for selection of the tuples for watermarking, which is known only to the user. Elliptic curve cryptography (ECC) decryption is used for watermark detection where user uses its private key and secret key.

As Elliptic curve cryptography (ECC) serves as an excellent candidate because of its small key size and high security protection, the proposed method is more secure and fast in comparison with other methods.

3.1 Proposed System Architecture

The system Architecture is defined in two phases: Watermark Embedding and Watermark Extraction. These modules are described in the following sections.

3.1.1 Watermark Embedding Model

Figure 3.1 shows the architecture for watermark embedding phase, where a watermarked data is created after inserting watermark in original data. Watermark is embedded in selected tuples only. For selecting a tuple, primary key (P_k) of that tuple and a secret key (K_s) is processed and tested for some selection criteria. If tuple is selected for watermarking then an Elliptic Curve point (P) is generated for primary key (P_k) using ECC definition. A public-private key pair is generated for the same Elliptic Curve through the user and user's public key is used to apply encryption on the

generated elliptic curve point (P). After encryption, this module returns two elliptic curve points, C_1 and C_2. These points are inserted into the selected attributes in selected positions (after decimal point).

If any tuple is not selected for watermarking, it is copied into watermarked database. So after processing every tuple watermarked data is created.

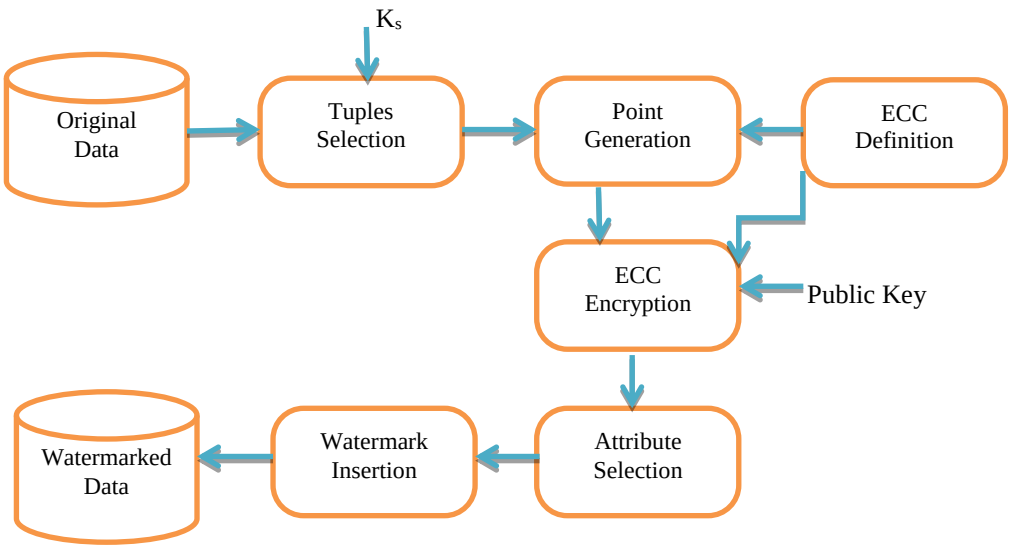

Figure 3.1 Watermarking Embedding Module

3.1.2 Watermark Extraction Model

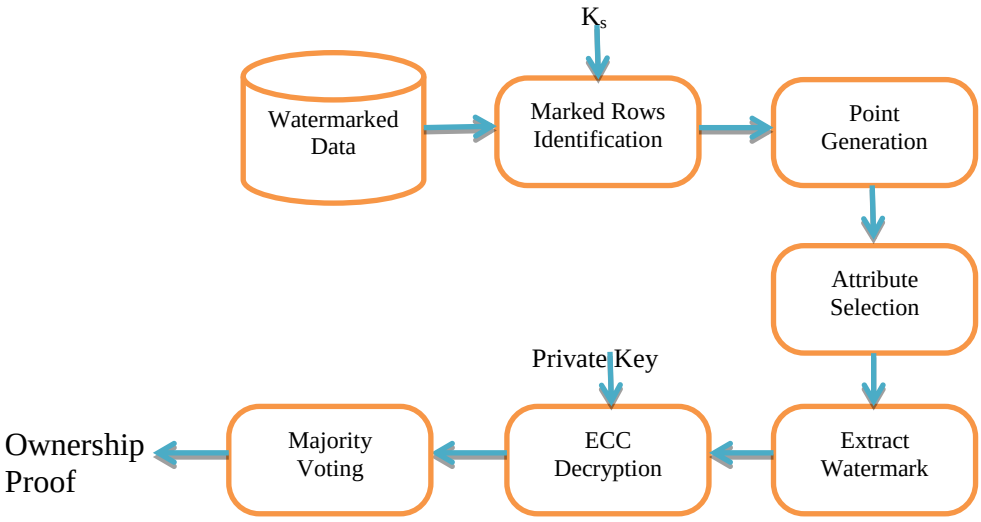

Figure 3.2 Watermarking Extraction Module

Figure 3.2 shows the architecture for watermark extraction module, where watermark is extracted from the marked rows, after identification of marked rows. Apply ECC decryption through user's private key and generate an elliptic curve point (P'). The previous method is used to generate elliptic curve point (P) of primary key (P_k) for same elliptic curve. If P' and P are same points, watermark is detected for this tuple.

Apply the same method for every tuple and count the match found. Test these count through the threshold value, if count is larger than threshold, watermark is detected.

3.2 Proposed Algorithms

Suppose that watermarking a database relation R whose scheme is R (P_k, A_0, . . . , A_{v-1}), where P_k is the primary key attribute. For simplicity, assume that all ν attributes A_0, . . . , A_{v-1} are candidates for marking. Thus each attribute is numeric and decimal number with values such that changes in ξ least significant bits (LSBs) after decimal point are imperceptible.

There are basically two algorithms used, one for watermark embedding and another for watermark extraction. Explanation of each algorithm is as follows:

3.2.1 Watermark Embedding Algorithm

Watermark embedding algorithm is defined as follows:

1. Given a set of domain parameters that include a choice of base field prime p, an elliptic curve E, a base point E_1 on E and a random number d of order p is selected.
2. ECC Public key (E_1, E_2, E_p) and private key (d) has been generated.
3. for each tuple r ∈ R do

    a. Q = **MAC(P_k, K_s, p)**
    b. if ((Q % t) == 0) then // mark this tuple

        i. P = **point_generation (Q, p)** //Elliptic Curve point is created for P_k

ii. Generate watermarking text C: (C_1, C_2) through ECC Encryption method

iii. For i=0 to 3 repeat

a. a = attribute_selection( ) //Select any 4 subsequent attributes for insertion in rotation with v attributes ((a+3)%v)

b. $r.A_a$ = **mark(C, r.A_a)**

In the above algorithm t is the percentage of tuples that user want to mark. In general case it can be taken as 10. **MAC( )** is a subroutine to apply a MAC function between Primary key P_k and User's Secret key K_s that return a digest of order p. **point_generation( )** is a subroutine call which converts any numeric value in to an elliptic curve point P(x,y). **attribute_selection** is a subroutine call which is used to select any numeric attribute in the tuple. And **mark( )** is a subroutine call that will insert ciphertext values, elliptic curve points, C_1(x,y) and C_2(x,y) into the selected attributes.

The Detailed descriptions of these subroutines are described in the following subsection.

3.2.1.1 MAC Algorithm

The MAC subroutine algorithm is as follows:

MAC(int P_k, int K_s, int p)

1. L_1=length of p in bits
2. L_2=length of P_k in bits
3. L_3=length of K_s in bits
4. If ($2*L_1 < L_2$)
    a. Make blocks of P_k from LSB of size $2L_1$
    b. If last block (MSBs) is having lesser bits then $2L_1$, Append number of 0's in MSB's as last block size become $2L_1$.

Else if ($2*L_1 > L_2$)

a. Append number of 0's in MSB's as block size of P_k become $2L_1$.

5. If ($2*L_1 < L_3$)
   a. Make blocks of K_s from LSB of size $2L_1$
   b. If last block (MSBs) is having lesser bits then $2L_1$, Append number of 0's in MSB's as last block size become $2L_1$.

   Else if ($2*L_1 > L_3$)

   a. Append number of 0's in MSB's as block size of K_s become $2L_1$.

6. $Q = K_s \oplus p$ //Apply Longitudinal XOR between blocks of q and K_s of size $2L_1$.
7. Convert Q into integer numbers of order 2p and return integer value of Q.

3.2.1.2 Point Generation Algorithm

The point_generation subroutine algorithm is as follows:

point_generation(int q, int p)

1. L_1=length of p in bits
2. Convert q into binary number block of size $2L_1$
3. Break q in 2 equal blocks of size L_1
4. Convert each block into decimal number of order p that become an elliptic curve point P(x,y)
5. Return point P(x,y)

3.2.1.3 Attribute Selection Algorithm

The attribute_selection subroutine algorithm is as follows:

attribute_selection( )

1. Take an array A[ ]
2. j = 0
3. For each attribute A_i // where i=0 to v-1

a. If A_i is a decimal number attribute

i. A[j++] = i

4. Use a random number generator which generate any number r between 0 and j-1 or r=r%j// Random Number Generator should generate same series for similar seed values, where seed can be Primary Key P_k.
5. Return value of A[r]

For next 3 attributes

r = (r+1) % j, and return A[r] // r should be static

3.2.1.4 Mark Algorithm

The mark subroutine algorithm is as follows:

mark(C, r.A_a)

1. A[ ] = Binary representation of C
2. B[ ] = Binary representation of fraction part of r.A_a
3. L = length of A
4. For (I = 0 to L-1)

a. B[i+3] = A[i]

5. C = decimal conversion of B[ ]
6. Replace fraction part of r.A_a with C
7. Return r.A_a

3.2.2 Watermark Extraction Algorithm

Watermark extraction algorithm is defined as follows:

1. Given a set of domain parameters that include a base field prime p and an elliptic curve E.
2. totalcount = matchcount = 0
3. for each tuple r ∈ R do

c. Q = **MAC(P_k, K_s, p)** of order 2p

d. if ((Q % t) == 0) then // marked tuple

   i. totalcount = totalcount + 1

   ii. P = **point_generation (Q, p)** //Elliptic Curve point is created for P_k

   iii. For i=0 to 3 repeat

      a. a = attribute_selection( ) //Select 4 subsequent attributes where watermark was inserted.

      b. C = **extract($r.A_a$)**

   iv. P_1 = Decrypt C through private key of user using ECC

   v. If (P_1 = = P) then

      matchcount=matchcount+1

e. τ = threshold

f. if ((matchcount/totalcount) > τ) then

   suspect piracy

Watermark extraction algorithm first finds the watermarked rows through the Primary key P_k and Secret key K_s in the same way as in watermark insertion algorithm and convert its Primary key P_k into an elliptic curve point P. Then it extracts C_1 and C_2 through the selection of attributes where C_1 and C_2 were inserted. For extraction a subroutine extract( ) is used. Through C_1, C_2 and d (Private key of user) value of P_1 is calculated through the ECC decryption function.

Whenever a marked row is found, totalcount is incremented to define the total number of rows that are marked and identified. Whenever a match is found, matchcount is incremented to define the number of matching rows identified.

A threshold value, τ should be defined by the user in the range from 0.5 to 0.9. Total watermarked rows are defined by totalcount and extracted or matched watermarked rows are defined by matchcount so for watermark extraction matchcount/totalcount

should be greater than τ, or in other words, percentage of matched rows should be larger than τ. So value of τ is depends on the sensitivity of the security of database to be watermarked.

This chapter proposed a technique of watermarking relational databases that embeds the watermark bits in the database using Elliptic Curve Cryptography (ECC). This technique is designed for numerical database and distortion based. It first identifies tuple then attribute and then marking position for the watermark. It embeds the watermarks at different attributes at different places. Therefore, it will be difficult for attacker to remove watermarks from different places from the database. Watermark bits are depending on the primary key, elliptic curve group and secret key. So it will be difficult for the attacker to identify the watermarked data.

CHAPTER 4

RESULTS AND ANALYSIS

This chapter presents experimental results of watermarking data. The experimental results are shown by graph that handles deletion attack for 90% deletion of rows from watermarked data. The proposed technique is also able to handle 70% insertion of new rows in watermarked data.

The proposed technique has been tested and evaluated with the help of experimental database. The database consist 3216 tuples and 90 attributes, from which 17 numeric (floating point) attributes selected for experiments. Experiments performed on Windows 7 Home Premium operating system with 2.53 GHz Intel ® Core(TM) i3 CPU and 4GB RAM.

4.1 Performance Analysis

The proposed algorithm has been evaluated and tested on an experimental database "indiacompfirm" that is downloaded from some open source. The watermark insertion and extraction were implemented using Turbo C++. The experiments were performed on a computer running Microsoft Windows 7, with 2.53 GHz Intel Core i3 processor and 4 GB RAM. Algorithm is applied to the real-life "indiacompfirm" dataset. Without loss of generality, 3216 tuples, having 17 numeric attributes have been selected from this dataset.

The performance evaluation of the robustness of the proposed algorithm must be developed in such a way to make it difficult for an adversary to remove or without destroying the value of the object.

4.1.1 Blind detection

Watermark extraction method should not require the original database and the watermark information. In watermark extraction algorithm only the private key of the original user and watermarked data is required and does not need original database.

4.1.2 Invisibility

The elliptic curve group, public-private key pairs and secret key are only known to the database owner, so attacker will not be able to detect the position of watermark in the database. Watermarking information is inserted at four attributes which make the watermark more hidden.

4.1.3 Robustness

This method assures the location of the embedded watermark is irregular. This technique embeds encrypted primary key values using ECC as watermark. These properties ensure this technique efficiently defenses the attack of subset selection, subset adding and subset updating. It improves the robust of this technique in a great degree.

4.1.4 Effect to the data

Table 4.1 shows the difference between original database and database embedded. It selects five attributes which are embedded watermark. This Table shows that after embedding the watermark change very small and ensures the availability of database.

Table 4.1: Difference between Original and Watermarked Data Attributes

Primary Key	Cost of equity in US$		Total Default Spread for cost of debt (Company + Country)		Pre-tax cost of debt in US $		After-tax cost of debt in US $	
	Original Attribute	Embedded Attribute	Original Attribute	Embedded Attribute	Original Attribute	Embedded Attribute	Original Attribute	Embedded Attribute
BSE:531500	0.0987671	0.094372	0.047	0.036746	0.0774	0.063483	0.0511	0.031569
BSE:532733	0.1207206	0.094353	0.042	0.036629	0.0724	0.06361	0.0478	0.031686
BSE:500850	0.1042949	0.094285	0.037	0.036756	0.0674	0.064714	0.0445	0.041326
BSE:500251	0.09109	0.063013	0.037	0.036756	0.0674	0.07155	0.0445	0.046208
BSE:532839	0.09006	0.063205	0.037	0.036756	0.0674	0.076677	0.0445	0.043035

In this table, some attributes with lesser digits after decimal point, having similar values in original database are having similar watermark values. So it is required that attributes should have large digits after decimal point to store watermark effectively.

4.2 Attack Analysis

Suppose a user has generated a watermarked database by inserting its watermarking information in to the database through the proposed watermark insertion algorithm. If an attacker wants to corrupt or delete the watermark through various type of attacks. Attacker also wants to maintain the quality of data so that it remains useful for attacker. Attacker has no access the original data and the private parameters used to insert watermark into the database. Moreover, it is also not possible for him to guarantee that his attack will not violate the usability constraints because he does not have access to the original data set. Robustness of proposed watermarking scheme against tuple deletion, insertion, and alteration attacks is tested.

The database watermarking algorithm should make the watermarked database robust against the following types of attacks: subset deletion attack, subset addition attack, subset alteration attack, and subset selection attack. The result of the proposed algorithm for these attacks is shown in figure 4.1 in form of chart; where rows represent the percentage of attack of any type and column represent the percentage of watermark detection.

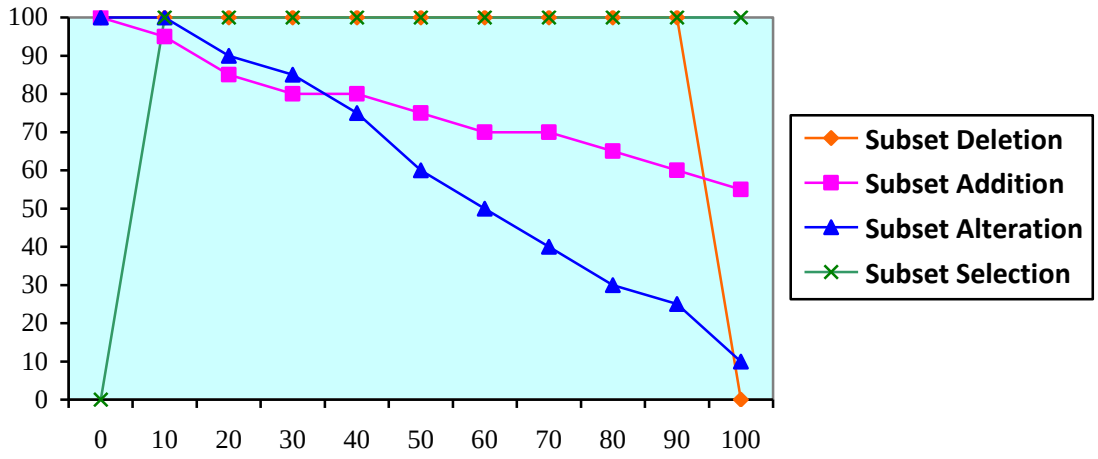

Figure 4.1 Performance of proposed algorithm against different Attacks

The result of the proposed algorithm for the different attacks is shown in table 4.2 in form tabular form also.

Table 4.2 Performance of proposed algorithm against different Attacks

Watermark detection (%) after Attack / Effect of Attack on data (%)	Subset Deletion Attack	Subset Addition Attack	Subset Alteration Attack	Subset Selection Attack
0	100	100	100	0
10	100	95	100	100
20	100	85	90	100
30	100	80	85	100
40	100	80	75	100
50	100	75	60	100
60	100	70	50	100
70	100	70	40	100
80	100	65	30	100
90	100	60	25	100
100	0	55	10	100

4.2.1 Subset deletion attack

In subset deletion attack, an attacker deletes some rows or data set or some attributes of the watermarked database randomly. The main target of the attacker is to disturb or remove the watermark. To proof this attack, some tuples or rows or attributes of the database are randomly deleted. The result is shown in the form of chart in figure 4.1, where black color line shows the percentage of watermark detection for subset deletion attack. According to the chart 100% watermark will be extracted even if an attacker deletes 90% of the database.

4.2.2 Subset alteration attack

In subset alteration attack, the attacker deletes some original data set and also inserts some more new data sets. It may create double data damage. The experiment result of this type of attack is shown in figure 4.1, where yellow color line shows the percentage of watermark detection for subset alteration attack. This figure shows that if 50% of the data is changed then also most of watermark (50%) is detected. In this testing most of the attributes altered which are marked.

4.2.3 Subset addition attack

In subset addition attack, the attacker inserts some more random or duplicate set to the original data. Figure 4.1 shows experiment result of this attack, where pink color line shows the percentage of watermark detection for subset addition attack. This figure shows that 70% watermark will be detected even if the attacker added 70% original tuples.

4.2.4 Subset selection attack

In subset selection attack, an attacker can randomly select and use only some set of the original database. The result of experiment is shown in figure 4.1, where blue color line shows the percentage of watermark detection for subset selection attack. This figure shows that the watermark will be available in the selected database even if the attacker selects a subset of size 10% of the original database, because proposed algorithm embed a watermark in the whole database in random manner.

In this chapter, the results of the simulation experiment have been represented to evaluate the performance of the proposed watermark embedding and extraction algorithm. The proposed technique is highly resilient against insertion, deletion, alteration, and selection attack yet it results in minimum distortions in the original data set. Regardless of the severity of malicious attack on the watermarked data, the watermark bits are successfully decoded with 100 percent accuracy because the decoding accuracy of the proposed approach is independent of the usability constraints. Moreover, this security mechanism also helps to resolve ownership conflicts over watermarked data set in case of additive attacks. Furthermore, this technique provides "maximum possible robustness" and delivers data with "minimum data distortions." Recall that the proposed technique is restricted to numeric data only. A logical extension of this research is to make it scale to non-numeric relational data sets as well.

CHAPTER 5
CONCLUSION AND FUTURE WORK

Watermarking algorithms are often used in larger system designed to achieve certain goals e.g., prevention of illegal copying. Watermarking relational databases is a merging research area that deals with the legal issue of copyright protection of relational databases. Watermarking database can be used to prevent database piracy, where somebody takes somebody else's database, slaps their name on it and then goes into competition with the original database producer. Protection from the piracy of digital assets is usually based upon the embedding of digital watermarks into the data. Watermarking approaches do not prevent copy rather it deter illegal copying by providing a means of establishing the original owners a redistributed copy.

In this report, a database watermarking algorithm is proposed which is based on the elliptic curve cryptography (ECC) to hide the watermark bits. A major advantage of using the ECC in database watermarking is the small key size security used to hide watermarks in the database. This is opposite to the more common bit level database watermarking algorithms where watermark bits have limited potential bit-locations that can be used to hide them without being subjected to removal or destruction. 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. The watermark resilience was improved by the repeated embedding of the watermark and using majority voting technique in the watermark decoding phase. Moreover, the watermark algorithm can be applied for more than one attribute of the same table.

Ongoing and future research includes the development of other effective database watermarking algorithms that can work on alphanumeric data and can resist all types of attacks and can evaluate it on large database.

References

[1] Joppe W. Bos, J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig and Eric Wustrow, "Elliptic Curve Cryptography in Practice", in *Financial Cryptography and Data Security* – FC 2014, Lecture Notes in Computer Science, Springer-Verlag (2014), to appear. Cryptology ePrint Archive, Report 2013/734.

[2] Elaine Brow, "Elliptic Curve Cryptography", in *Math 189A: Algebraic Geometry*, p.g. 1-5 December 2010.

[3] Samta Gajbhiye, Dr. Sanjeev Karmakar, Dr. Monisha Sharma, Dr. Sanjay Sharma and Dr. M K Kowar, "Application of Elliptic Curve Method in Cryptography: A Literature Review", *International Journal of Computer Science and Information Technologies*, Vol. 3 (3) , 2012.

[4] Ruchika Markan and Gurvinder Kaur, "Literature Survey on Elliptic Curve Encryption Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering* Volume 3, Issue 9, September 2013.

[5] William Stallings, "Asymmetric Ciphers", in *Cryptography and network security, Principles and Practice*, 5th Edition, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2011, pp. 267-350.

[6] Behrouz A. Forouzan, "Asymmetric-Key Encipherment, Digital Signature", in *Cryptography and Network Security*, Special Indian Edition, Tata McGraw Hill, 2007, pp. 249-411.

[7] Fuwen Liu, " *A Tutorial on Elliptic Curve Cryptography (ECC)*", Brandenburg Technical University of Cottbus Computer Networking Group, 2014.

[8] Manoj Kumar, *Cryptography and Network Security*, Krishna Prakashan Media Ltd. India, 2009.

[9] Agrawal, R. and Kiernan, J., "Watermarking relational databases", in *Proceeding of the 28th International conference on Very Large Databases*, p. 155-166, 2002.

[10] Agrawal, R., Haas, P.J. and Kiernan, J., "Watermarking relational data: framework, algorithms and analysis", *VLDB Journal*, vol.3, 2003.

[11] Zhi-hao Zhang, Xiao-ming Jin, Jian-min wang and De-yi Li, "Watermarking relational database using image", in *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 3, 2006, pp. 1739-1744.

[12] Jianhua Sun, Zaihui Cao and Zhongyan Hu, "Multiple Watermarking Relational Databases Using Image", in *IEEE International Conference on MultiMedia and Information Technology*, 2008, pp. 373-376.

[13] Chaokun Wang, Jianmin Wang, Ming Zhou, Guisheng Chen and Deyi Li, "Atbam: An Arnold transform based method on watermarking relational data", in *Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering*, 2008, pp. 263-270.

[14] Zhongyan Hu, Zaihui Cao and Jianhua Sun, "An Image Based Algorithm for Watermarking Relational Databases*", in Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation*, 2009, pp. 425-428.

[15] Theodoros Tzouramanis, "A Robust Watermarking Scheme for Relational Databases", in *IEEE 6th International Conference on Internet Technology and Secured Transactions*, December 2011.

[16] Udai Pratap Rao, Dhiren R. Patel and Punitkumar M. Vikani, "Relational Database Watermarking for Ownership Protection", in *Elsevier 2nd International Conference on Communication, Computing & Security* [ICCCS-2012], 2012.

[17] Yu Fu, Tianyu Ye, Zhiguo Qu, Xinxin Niu and Yixian Yang, "A Novel Relational Database Watermarking Algorithm for Joint Ownership", in *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008.

[18] Ms. Arti Deshpande and Mr. Jayant Gadge, "New Watermarking technique for Relational Databases", in *IEEE Second International Conference on Emerging Trends in Engineering and Technology*, ICETET-09, 2009.

[19] Damien Hanyurwimfura, Yuling Liu and Zhijie Liu, "Text Format Based Relational Database Watermarking for Non-numeric Data", in *IEEE International Conference On Computer Design And Appliations* (ICCDA 2010), 2010.

[20] Kaiyin Huang, Min Yue , Pengfei Chen, Yanshan He and Xiaoyun Chen, "A Cluster-Based Watermarking Technique for Relational Database", in *IEEE First International Workshop on Database Technology and Applications,* 2009.

[21] Lizhong Zhang, Wei Gao, Nan Jiang, Liqiu Zhang and Yan Zhang, "Relational Databases Watermarking for textual and numerical data", in *IEEE International Conference on Mechatronic Science, Electric Engineering and Computer*, Jilin, China, August 2011.

[22] Vidhi Khanduja, Shampa Chakraverty, Om Prakash Verma, Rakshita Tandon and Sahil Goel, "A Robust Multiple Watermarking Technique for Information Recovery", *IEEE International Advance Computing Conference* (IACC), 2014.

[23] Hongbin Kong, Zhengquan Zeng, Lijun Yan, Jicheng Yang, Shaowen Yao and Nuoya Sheng, "Combine Elliptic Curve Cryptography with Digital Watermark for OWL Based Ontology Encryption", in *International Conference on Computational Intelligence and Security*, 2009.

[24] Vahab Pournaghshband, "A New Watermarking Approach for Relational Data", *ACM-SE '08, March 28–29*, 2008, Auburn, AL, USA. Copyright 2008 ACM ISBN.

[25] Raju Halder, Shantanu Pal and Agostino Cortesi, "Watermarking Techniques for Relational Databases: Survey, Classification and Comparison", in *Journal of Universal Computer Science*, vol. 16, no. 21 (2010), 3164-3190, submitted: 18/12/09, accepted: 29/11/10, appeared: 1/12/10.

[26] Al-Haj, A. and Odeh, A., "Robust and blind watermarking of relational database systems" in *Journal of Computer Science*, 2008 pp. 4:1024–1029.

[27] Hanyurwimfura Damien and Yuling Liu "Text Format Based Relational Database Watermarking for Non-numeric Data", in *IEEE ICCDA*, Vol. 4 2010, pp.312-316.

[28] Javier Franco-Contreras, Gouenou Coatrieux, Fréderic Cuppens, Nora Cuppens-Boulahia, and Christian Roux, "Robust Lossless Watermarking of Relational Databases Based on Circular Histogram Modulation", in *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 3, MARCH 2014.

[29] Ying Wang, Geng-Ming Zhu and Shao-Bo Zhang, "Research On The Watermarking Algorithm Based On Numerical Attribute In the Relational Database", in *IEEE International Conference on Computer Science and Electronics Engineering*, 2012.

[30] Saman Iftikhar, M. Kamran and Zahid Anwar, "RRW - A Robust and Reversible Watermarking Technique for Relational Data", in *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[31] Vahab Pournaghshband, "A New Watermarking Approach for Relational Data", *ACM-SE'08*, 2008.

[32] Ms. Snehal, S. Kshatriya and Prof. Dr. S. S. Sane, "A Study of Watermarking Relational Databases", in *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Volume 3, Issue 10, October 2014.

[33] Wang Yanmin and Gao Yuxi, "The Digital Watermarking Algorithm of the Relational Database Based on the Effective Bits of Numerical Field", in *IEEE Explore, World Automation Congress (WAC)*, June 2012.