Elliptic Curve Cryptography (ECC) based Relational Database Watermarking

Manoj Kumar
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India

O. P. Verma
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India

Archana Saxena
Department of IT
JSS Academy of Technical Education
NOIDA, India

ABSTRACT
Database Watermarking methods are used for copyright protection of relational database. Many techniques have been developed for watermarking multimedia digital assets like audio, video, text, images etc. But these methods are usually not applicable with numerical database, because to insert a watermark into a data, small error is created in data, called mark. An error in relational data is usually not acceptable, so a different approach need to be develop to create a mark into the numerical database. Many different approaches have been discussed in previous researches for relational database watermarking. In this paper a new approach has been introduced for relational database watermarking using Elliptic Curve Cryptography (ECC). It has been proofed that ECC provides more security with smaller key size in comparison with other encryption techniques. Proposed approach gives better results in subset deletion and selection attacks which have been compared with other methods.

Keywords
Relational Database Watermarking, Elliptic Curve Cryptosystem, Watermark Extraction, Watermark Insertion, Watermarking.

1. INTRODUCTION
During the last few years, internet has grown from 2G to 4G. Simultaneously, the challenges and issues faced by the data owners have grown multifold. The security of data is the primary challenge for the data owners as the threat of data piracy looms large over their digital assets. For the data such as image, video, audio, text or even software, a watermark can be inserted as a method of security of these digital assets. To create a watermark in to an object, watermarking software creates an intentional small error into the object 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. On internet, data provider create many applications and services where users can access and search the data remotely and pay for using database or buy a copy of database for use. So for this some watermarking techniques are used 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 demand for the database watermarking technologies to identify the pirated copies of their databases.

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 [1].

1. Watermark should not reduce the quality of the data.
2. Multiple watermarks inserted 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 possible prescribed attacks on watermarked data without degrading the quality of data.

Many watermarking techniques are based on different watermark information; most of these techniques are designed for numerical database and are distortion based. These techniques usually have almost similar steps to identify attribute, tuple, and marking position within attribute value for the watermark. Most of these techniques use a single attribute of a tuple to embed a watermark. This is done by embedding the watermarks in scattered fashion among different attributes at different places in attribute value. It makes it difficult for attacker to remove watermarks from different places from the database. Most of these techniques are also dependent on presence of primary key and use of cryptographic technique, thus making it difficult to observe presence of watermark.

To insert a watermark into the database, a basic technique is used. The watermark information is first converted into a binary digital watermark and then inserted into the original database through a user's secret key. It is done in such a way that attacker will not be able to identify the inserted watermark until he knows user's secret key as well as related parameters used during insertion of watermark. Watermark is correctly extracted using user's secret key and related parameters.

The database watermarking model mainly includes three algorithms: Digital Watermark Generation Algorithm, Digital Watermarking Embedding Algorithm and Digital Watermarking Extraction Algorithm:

1. Digital Watermark Generation Model: Digital watermark can be any text, image, secret code or pattern. To insert a watermark into a database, watermark should be processed and converted to some binary stream. It is shown in figure 1.

Fig 1: Digital Watermark Generation Model

2. Digital Watermarking Embedding Model: It is used to hide the processed binary stream derived from watermark into some database. Digital Watermarking Embedding Algorithm embeds watermarks in a way that it does not affect the use of database. A secret key is usually used to select rows and possibly attribute(s) of database where watermarking bits are to be embedded.

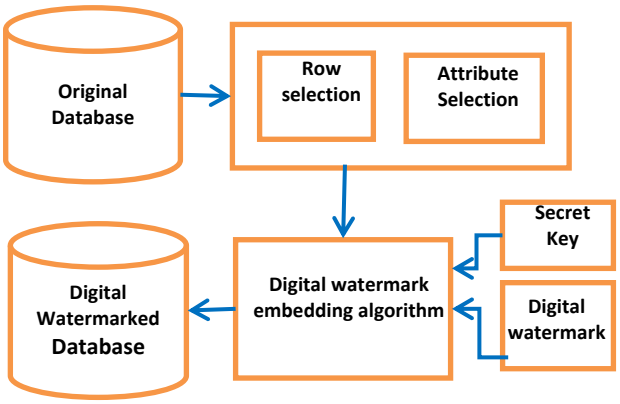

Fig 2: Digital Watermarking Embedding Model

3. Digital Watermarking Extraction: The secret key wasused to embed watermark in embedding process. Same secret key is used to extract the watermark from watermarked database by a process identical to embedding process. Instead of embedding, it identifies the watermarking bits from attribute values. Secret key is used to identify the rows and attribute(s) where watermarking bits are present. Watermarking Extraction Algorithm recovers the original digital watermarking signal after processing it. It is shown in figure 3.

Fig 3: Digital Watermarking Extraction Model

A watermarking technique should satisfy the following properties as described in the following points [2]:

1. ***Robustness:*** Watermarks should be robust against degradation caused by either benign updates or malicious attacks.
2. ***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*.
3. ***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.
4. ***Blind system:*** Watermarking detection technique should not require the original database or the watermark.
5. ***Public system:*** The watermarking system should assume that the method used for inserting a watermark is public.

A watermarking technique should be resistant for intentional and unintentional attacks[2]. A watermark can be attacked through different ways. The type of attacks that are possible on user's watermarked data are described below:

1. **Benign updates:** These can be unintentional updates, where rows that are marked may be deleted or marked bits may be changed during any ordinary data processing.
2. **Malicious attacks:** An attacker intentionally steals the data and may attack to remove watermark in different forms as 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 this data may become useless.
   - **Rounding Attack:** An attacker can round off the values of all numeric attributes to remove the marks hidden in decimal numbers. But he may degrade 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.
   - **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.
3. **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.

The description given above comprises the introductory section. Section II gives an overview of the related previous work. Section III describes the detailed description of Watermark Embedding and Watermark Extraction Algorithm. Section IV shows the results and analysis of proposed method with comparison with some previous methods. Section V concludes the proposed method and gives the direction for future work.

2. RELATED WORK

Tremendous work has been done in the area of Database Watermarking as well as Elliptic Curve Cryptography (ECC). The area of digital watermarking been popular as a research

since the mid 1990s

There are different watermarking techniques for watermarking different type of data. These techniques implement watermarking into images and then are used for audio and video data. There are some technical challenges to implement database watermarking technique because relational data differs from other type of data (multimedia)[2]. The differences are highlighted as follows:

A multimedia data has a large number of redundant bits so it is very easy to hide the watermarking bits into the multimedia data. In relational database, multiple tuples represent a separate object 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 do not have 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 it. But for relational database, it is a normal procedure, like tuple insertion, deletion and updation.

Therefore, watermarking techniques implemented for other type of the data (multimedia data) cannot be directly used for relational database. Initially, Agarwal et al.[2, 3] have proposed the relational database watermarking in 2002. This method has been implemented only for numeric data. Then, Zhang et al.[4] have proposed the same technique for image based Relational Database Watermarking. The drawback of this method has been that watermark extraction was not possible if attacker change the order of the tuples or attributes. Sun et al.[5] have proposed another method to insert an image into the database as watermark information, where image is converted into flow of bits and inserted according to the hash value of the database tuple.

Wang et al.[6] proposed a different method to insert an image into the database as watermark information based on Arnold transform, where image has been used as a binary string. This method was more effective because it used many factors. Cao et al.[7] proposed an advanced technique to insert an image into relational database, where an image was converted into bit flow using EMC (Encrypted Mark Code).

Theodoros [8] proposed multipurpose scheme because it could be used for both watermarking (embedding and detecting the same bit string in every database copy) and fingerprinting (embedding and detecting the different bit string 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.[9] have 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.[10] have proposed a different method of database watermarking where a secret key is divided into multiple parts and embedded into the relational database.

Hanyurwimfura et al.[11] proposed an advanced technique for watermarking relational database for non-numeric data. This technique involves the insertion of mark through horizontally shifting the words and calculation of the location of insertion by Levenshtein Distance.

Huang et al.[12] have proposed a watermarking technique for relational database which is based on cluster theory, where a cluster is used to insert watermark into database. They have also introduced the method of odd-even modifying method for watermark information.

The modified method of [12] has been proposed by Khanduja et al.[13]. 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 illustrates 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.[14] have proposed a watermarking technique which uses ECC for watermark and is based on OWL-based ontology encryption.

Javier et al.[15] have proposed a robust lossless relational database watermarking scheme which makes use of circular histogram modulation. It is used for 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.

Most of the methods, mentioned above, have limited exposure because of watermark insertion technique procedures. Proposed method in this paper uses DES for watermark insertion which increases speed and decreases computational cost. Moreover, in the proposed method, insertion take place in more than one places for each key so it is not easy for attacker to attack the watermark.

3. PROPOSED WORK

The technique has been proposed in this section for Relational Database Watermarking satisfies all requirements discussed previously. 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 non-obvious. All of the numeric attributes of all the tuples are not marked. The data owner and primary key of the data decides the suitable attributes for marking.

Elliptic Curve Cryptography (ECC) encryption is used to create watermark where user creates 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.

Elliptic Curve Cryptography (ECC) also serves as an excellent

candidate for watermark 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 as follows:

1. Watermark Embedding Model

Figure 4 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 floating point attributes in selected position of mantissa part.

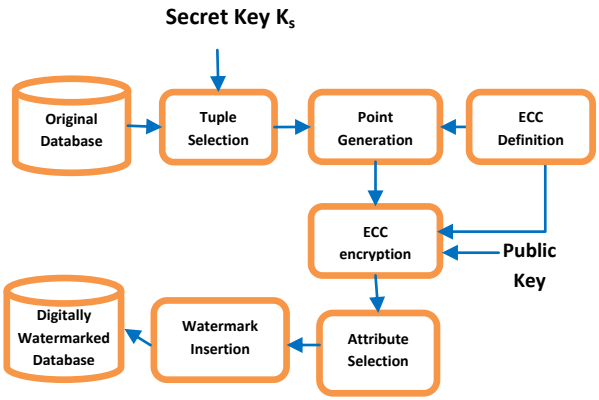

Fig 4: Architecture for Watermark Embedding Model

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

2. Watermark Extraction Model:

Figure 5 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.

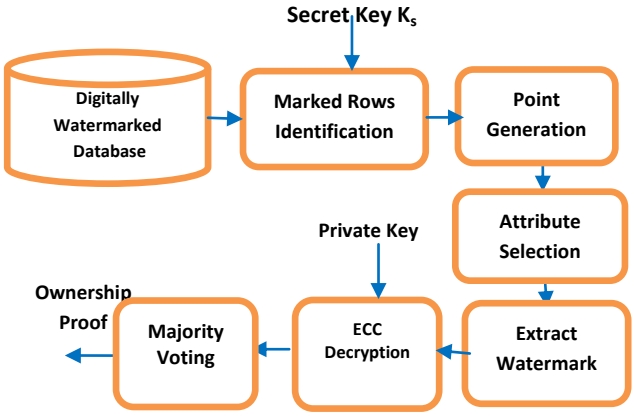

Fig 5: Architecture of Watermark Extraction Model

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 (PK, A_0, . . . , A_{v-1}), where PK is the primary key attribute. For simplicity, assume that all v 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 Proposed Watermark Insertion Algorithm

Watermark insertion 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. $E_2 = E_1 * d$ //Where public key: (E_1, E_2, E_p)
   and private key: d
3. for each tuple r ∈ R do
   a. Q = **MAC(PK, K_s, p)** of order 2p
   b. if((Q % t) == 0) then // mark this tuple
      i. P = **point_generation(Q, p)** //Elliptic Curve point is created for PK
      ii. Select a random number r
      iii. $C_1 = r * E_1$; $C_2 = P + r * E_2$
      // Ciphertext C: (C_1, C_2)
      iv. 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)**
         // for i=0: C = C_1.x

// for i=1: C = C_1.y

// for i=2: C = C_2.x

// for i=3: C = C_2.y

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 PK 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 into 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 as follows.

MAC Algorithm:
The MAC subroutine algorithm is as follows:

MAC(PK, K_s, p)

1. L_1=length of p in bits
2. L_2=length of PK 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 $2*L_1$
   b. If last block (MSBs) is having lesser bits then $2*L_1$, Append number of 0's in MSB's as last block size become $2*L_1$.

   Else if ($2*L_1 > L_2$)
   a. Append number of 0's in MSB's as block size of PK become $2*L_1$.
5. If ($2*L_1 < L_3$)
   a. Make blocks of K_s from LSB of size $2*L_1$
   b. If last block (MSBs) is having lesser bits then $2*L_1$, Append number of 0's in MSB's as last block size become $2*L_1$.

   Else if ($2*L_1 > L_3$)
   a. Append number of 0's in MSB's as block size of K_s become $2*L_1$.
6. Q = $K_s \oplus$ p //Apply Longitudinal XOR between blocks of q and K_s of size $2*L_1$.
7. Convert Q into integer numbers of order 2*p and return integer value of Q.

Point Generation Algorithm:

The point_generation subroutine algorithm is as follows:

point_generation(q, p)

1. L1=length of p in bits
2. Convert q into binary number block of size $2*L_1$
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)

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
      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

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 Proposed Watermark DetectionAlgorithm

Watermark Detection 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. total_count = match_count = 0

3. for each tuple r ∈ R do
   a. Q = **MAC(P_k, K_s, p)** of order 2p
   b. If ((Q % t) == 0) then// marked tuple
      i. total_count = total_count + 1
      ii. P = **point_generation(Q, p)** //Elliptic Curve point is created for PK
      iii. For i=0 to 3 repeat
         a. a = attribute_selection() //Select four subsequent attributes where watermark was inserted.
         b. C = **extract(r.A_a)**

// for i=0: C = C_1.x

// for i=1: C = C_1.y

// for i=2: C = C_2.x

// for i=3: C = C_2.y

iv. $P_1 = (C_2 - (d * C_1)) \bmod p$ // $d*C_1$ is additive inverse of C_2

v. If (P_1 = = P) then

match_count=match_count+1

c. τ = threshold

d. if ((match_count/total_count) > τ) then

suspect piracy

Watermark Extraction Algorithm first finds the watermarked rows through the Primary Key PK and Secret key K_s in the same way as in watermark insertion algorithm and convert its Primary key PK 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 have been inserted. For extraction a subroutine extract() is used. Through C_1, C_2 and d (Private key of user) value of P1 is calculated through the ECC Decryption Function.

Whenever a marked row is found, total_count is incremented to define the total number of rows that are marked and identified. Whenever a match is found, match_count 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 total_count and extracted or matched watermarked rows are defined by match_count so for watermark extraction match_count/total_count should be greater than τ, or in other words, percentage of matched rows should be larger than τ. Thus the value of τ is depends on the sensitivity of the security of database to be watermarked.

4. EXPERIMENTAL RESULTS

The proposed method has been evaluated and tested with the help of experimental database. The database contains 3216 tuples and 90 attributes, from which 17 numeric (floating point) attributes have been 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 real-life "indiacompfirm" dataset. Without loss of generality, 3216 tuples, 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 was applied to a database having 17 numeric attributes selected from this dataset.

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

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.

Invisibility

The Elliptic Curve Group, public-private key pairs and secret key are known only to the database owner. Therefore, the 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.

Robustness

This method assures that 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 robustness of this technique to a great degree.

Effect to the data

Table 1 shows the difference between original database and database embedded. It selects five attributes which are embedded watermark. The table below shows that after embedding, the database changes to a small extent and ensures the availability of database.

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

Table 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

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. An attacker can corrupt or delete the watermark through various type of attacks while maintaining the quality of data so that it remains useful for attacker. Attacker has no access the original data and the private parameters one 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.

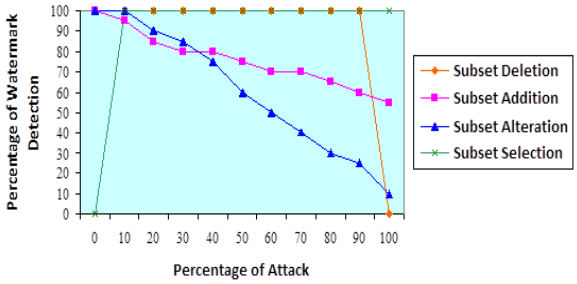

Fig 6: Performance of proposed algorithm against different Attacks

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 6 in form of chart where rows represent the percentage of attack of any type and column represent the percentage of watermark detection.

The result of the proposed algorithm for the different attacks is shown in table 2 in tabular form also.

Table 2Performance of proposed algorithm against different Attacks

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

4.3 Comparisons

The result of the proposed work has been compared with the previously published works which are also based on the Bit Insertion Method for watermark insertion where single or multiple bits are inserted into numeric data after decimal places. Results of subset addition attack, subset deletion attack, subset alteration attack and subset selection attack of proposed work DBWECC has been compared with RDWEBNF[16], RDWTN[17] and RDWOP[9] and are shown in figures 9, 10, 11 and 12 respectively.

4.3.1 Subset Addition Attack:

In subset addition attack, the attacker inserts some more random or duplicate set to the original data. Figure 6 shows experiment result of this attack, where pink colored line shows the percentage of watermark detection for subset addition attack. This figure shows that 70% watermark is detected even if the attacker adds 70% original tuples.

Comparisons of proposed work with the previous work [16,17,9] is shown in figure 7 where orange colored line shows the proposed work and red blue and green colored lines show the previous work [17], [16] and [9] respectively.

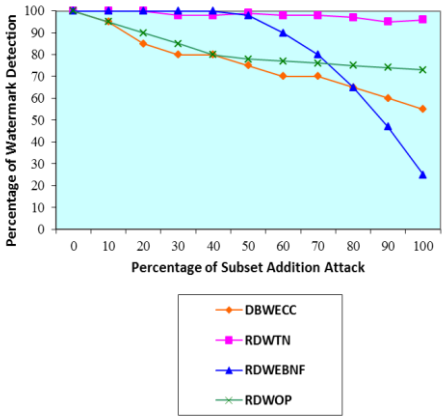

Fig 7: Subset Addition Attack

4.3.2 Subset Deletion Attack:

In subset deletion attack, an attacker deletes some rows or data set or some attributes of the watermarked database randomly. To proof this attack, some tuples or rows or attributes of the database are randomly deleted. The result is shown in figure 6, where black colored 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.

Comparisons of proposed work with the previous work [16,17,9] is shown in figure 8 where orange color line shows the proposed work and red, blue and green colored lines show the previous work [17], [16] and [9] respectively.

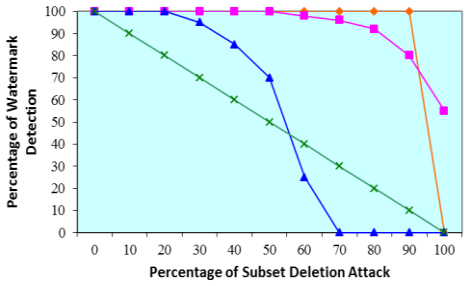

Fig 8: Subset deletion Attack

4.3.3 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 6, 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.

Comparisons of proposed work with the previous work [16,17,9] is shown in figure 9 where orange colored line shows the proposed work and red blue and green colored lines show the previous work [17], [16] and [9] respectively.

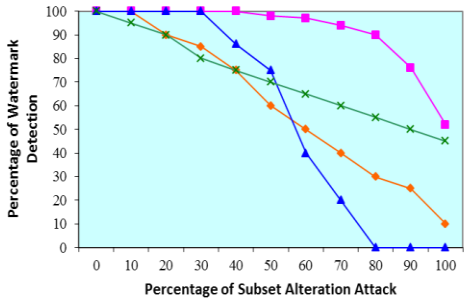

Fig 9: Subset Alteration Attack

4.3.4 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 6, where blue colored 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.

Comparisons of proposed work with the previous work [16,17,9] is shown in figure 10 where orange colored line shows the proposed work and red blue and green colored lines show the previous work [17], [16] and [9] respectively.

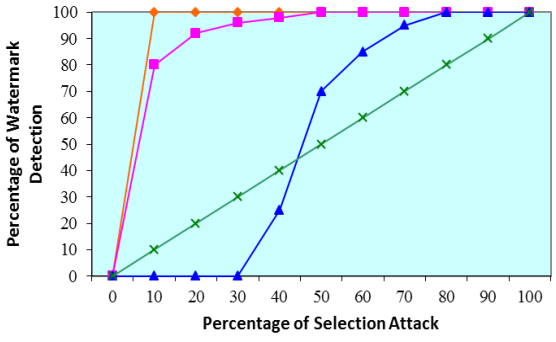

Fig 10: Subset Selection Attack

5. CONCLUSION AND FUTURE WORK

In this paper, 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 include the development of other effective methods for MAC and Random Attribute Selection and the development of other effective Database Watermarking Algorithms using ECC and other public key cryptographic algorithms, that can work on alphanumeric data and can resist all types of attacks and can evaluate it on large database.

6. REFERENCES

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

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

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

[4] Zhi-hao Zhang, Xiao-mingJin, 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.

[5] 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.

[6] 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 2008International Conference on Multimedia and Ubiquitous Engineering*, 2008, pp. 263-270.

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

[8] TheodorosTzouramanis, "A Robust Watermarking Scheme for Relational Databases", in *IEEE 6thInternational Conference on Internet Technology and Secured Transactions*, December 2011.

[9] UdaiPratap 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.

[10] Yu Fu, Tianyu Ye, ZhiguoQu, XinxinNiu 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.

[11] 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.

[12] 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.

[13] VidhiKhanduja, ShampaChakraverty, Om Prakash Verma, RakshitaTandon and SahilGoel, "A Robust Multiple Watermarking Technique for Information Recovery", *IEEE International Advance Computing Conference* (IACC), 2014.

[14] 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.

[15] Javier Franco-Contreras, GouenouCoatrieux, FrédericCuppens, 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.

[16] Wang Yanmin and GaoYuxi, "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.

[17] 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.