ISBN 978-81-936279-8-3
10th International Conferences on Emerging Engineering Technologies and Healthcare (EETH-18)
London (UK) June 25-26, 2018

Distortion-Free Watermarking Techniques for Relational Databases: Classification and Analysis

Seema Siledar[1] and Dr. Sharvari C. Tamane[2]
[1]Research Scholar, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad
[2]Professor, Jawaharlal Engineering College, Aurangabad

Abstract**:** *Along with the growing use of digital data, there come various challenges that raise question over security of such data. Now-a-days, some common issues faced by relational database owners are tamper detection, ownership protection, integrity verification, theft and so on. To solve these problems, digital watermarking emerges as a powerful tool. Most of the watermarking techniques for relational database introduce small distortion or error into underlying data. This may not always be tolerable by the user. In this paper, we present systematic review of various distortion-free watermarking techniques. According to their applications, these techniques are being classified into two types: fragile and robust. We analyze fragile techniques based on the way they create watermark, their compatibility with various attribute types, watermarking key used and probability for tamper detection. For robust techniques, the analyzing parameters are detectability in public or private, false hit, false miss and incremental updatability.*

Keywords: *watermarking, database, distortion-free, fragile, robust, ownership; integrity*

1. Introduction

Internet technology has seen a rapid growth in past few years. Due to this, the use of digital media has been increased tremendously over the Internet. Digital media is published and distributed on electronic devices. It has become a preferred way for faster and easier communication. As a result, several organizations are offering web-based services such as online decision support system, e-commerce, database as a service and many more. However, this has posed new challenges for copyright protection, tamper detection, integrity verification, theft, *etc*. To overcome these issues, there are various techniques like encrypting data, digitally signing data, digital watermarking and many others. Although encryption and digital signature ensure secure distribution of digital data, they cannot guarantee protection once the data is decrypted by the third- party. Digital watermarking emerges as an effective solution to protect data against tampering or theft. It is possible to embed an imperceptible watermark in digital images and videos due to considerable redundancy of pixels in them. However, relational database consists of distinct tuples which represents distinct records. Apart from this, relational databases possess following properties described in [1]:

- Tuples of a relation constitute a set and there is no implied ordering in them
- Pirate of relation can simply drop some tuples or substitute them with tuples from other relations

All these properties impose the need to explore new watermarking techniques for relational databases.

2. Basic Watermarking Technique

In 2002, Agrawal and Kiernan proposed watermarking technique specifically geared for relational data [1]. The basic watermarking technique consists of two phases, namely, watermark insertion and watermark extraction. These phases are illustrated in Fig. 1 and Fig. 2 respectively.

During watermark insertion, some watermark information *W* is inserted in original database *R* by using owner's private key *K*. The watermarked database is distributed for public use. In case, the owner suspects some publically available database *R'*, then, watermark can be extracted in next phase. During watermark extraction, owner uses her private key *K* to extract embedded watermark information *W'* from suspicious database *R'*. If the inserted watermark information *W* and extracted watermark information *W'* are exactly same, then, the owner can claim her authority over the database. Otherwise, *R'* can be treated as a genuine database.

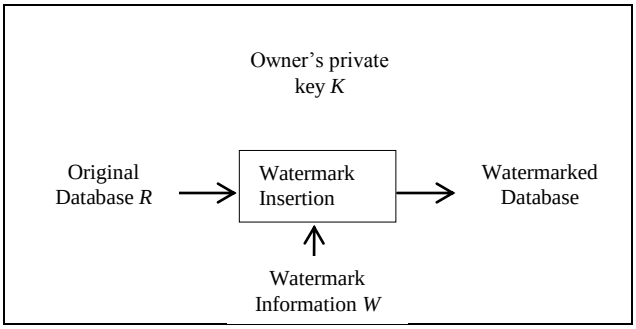

Fig. 1: Process for Watermark Insertion

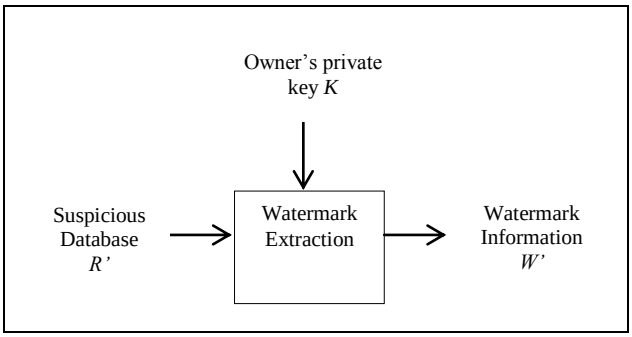

Fig. 2: Process for Watermark Extraction

3. Terminology

3.1. Types of Watermark

Watermark can be categorized into two main types: robust and fragile. A watermark is robust if it remains unaltered even after the modifications are made to the database. A watermark is fragile if it gets destroyed due to the slightest modification made to the database [2].

Generally, the digital watermarking for integrity verification is called fragile watermarking and the one for ownership protection is called robust watermarking [3].

3.2. Watermarking Techniques

Watermarking techniques can be categorized into two types depending on whether the inserted watermark introduces any distortion to the database or not. They include:

- Distortion-based watermarking: If the inserted watermark introduces small errors into the database being watermarked, then, the technique is distortion-based. These intentional errors are called marks and all the marks together form watermark [4].
- Distortion-free watermarking: If the inserted watermark does not introduce any error or change into the database being watermarked, then, the technique is distortion-free.

4. Classification of Distortion-Free Watermarking Techniques

Most of the researchers in past have focused on distortion-based watermarking techniques for relational databases since it was first introduced in [1]. Some techniques are based on numeric whereas others on categorical data attributes. However, the distortion introduced by these techniques may not be tolerable in certain applications. To overcome this problem, distortion-free techniques can be used to protect database without introducing any distortion in it. Such techniques can be further classified into robust watermarking and fragile watermarking based on the kind of application they are used for. According to the application, it is essential to choose appropriate type of watermark. Robust watermark is inserted for the purpose of ownership protection in which the watermark is supposed to resist all types of attacks against watermark destruction. On the other hand, fragile watermark is suitable for tamper detection where the aim is to distort watermark even if a minor change is made to any of the value.

4.1. Fragile Watermarking for Tamper Detection

Sometimes, data integrity is more important as compared to ownership proof. In such cases, watermark can be used to check whether data has been tampered or not. Suppose Alice has distributed her database relation R publically after inserting watermark information W using private key K. If she suspects that her database R has been tampered, then, she extracts watermark information W' from published database. If W and W' does not match, Alice can understand that her relation R has been tampered.

4.1.1. Extracting group based hash value to generate secret order among tuples

Authors in [5] proposed a watermarking technique based on virtual grouping and sorting operations. Tuples are grouped by using secure key hash value which is computed for each tuple r_i according to embedding key K and primary key of the tuple $r_i.P$. Tuples in group G_k are sorted considering their primary key hash values h_i^P. Like primary key hash, a secure tuple hash h_i is computed for each tuple using embedding key K and all attributes $A_1, A_2, .. A_\gamma$. Watermark W is formed by extracting some selected bits from group hash value H which is obtained from tuple hash h_i of sorted tuples in group and embedding key K. For each tuple pair, the order is changed or unchanged based on tuple hash h_i and corresponding watermark bit w. During watermark verification, watermark W' is extracted from the tuples. For every tuple pair, their respective hash values h_i and h_{i+1} are compared. If tuple hash h_i is greater than the hash of second tuple hash h_{i+1}, then, the related bit is 1, else, it is 0. In case, W and W' do not match, the database has been tampered. However, in some cases, the user may change the order of tuples unintentionally or without modifying any of tuple values, then, the embedded watermark may not be correctly verified.

In [6], during watermark embedding, tuples are partitioned based on some categorical attribute. It is similar to virtual grouping performed in [5]. All tuples are then sorted group-wise according to their primary key. *HMAC* function is used to compute keyed group hash value H_{qk} based on tuple hash value r_i . H_{qk} is used to embed watermark W by permutating the order of tuples. Myrvold and Ruskey's linear permutation unranking algorithm [7] is applied for calculating new order π from W. During watermark detection, ranking algorithm is used to get the new order from W' performing the prior steps of unranking in the same fashion. If W and W' are same, the database can be proved to be authentic, otherwise, it is supposed to be tampered.

In 2011, Guo proposed watermarking scheme [8] which is influenced by the ideas suggested in [5] and [6]. In [5], only tuples are grouped together using primary key hash value. However, in this scheme, attributes are also grouped based on attribute name hash h_j obtained from attribute name $c_j.A$ and grouping key K_g. For each group, hash information is calculated from secret embedding key K_e and canonical form of data in this group. The canonical form represents group of data sorted in ascending order according to primary key for rows and attribute name for columns. During watermark embedding phase, Myrvold and Ruskey's linear permutation unranking algorithm [7] is applied to return a unique permutation sequence π for rearranging entries in the group G based on verification information h. This is quite similar to the work done in [6].

The major difference between [6] and [8] is that integrity verification in [6] is based on only tuple group whereas it is based on both tuple and attribute group in [8]. If neither entities nor the order in the group have been changed, the ranking algorithm returns *h* which is exactly similar to verification information *h*. This indicates that database integrity is preserved.

Authors in [9] have proposed the use of Linear Feedback Shift Register (LFSR) to generate watermark for Internet based IP protection scheme. This idea has been further extended for relational databases in [10]. According to this scheme, watermark *W* is generated from the bits obtained by LFSR with initial fill as key *K*. The order of tuples in pair is changed or unchanged according to the watermark bit (0 or 1). During watermark verification, hash value *h[i]* and *h[i+1]* is calculated for every tuple pair. Like [5], if tuple hash *h[i]* is greater than that of *h[i+1]* , then, the related watermark bit *W'[j]* is set to 1. Else, it is 0. Similarly, all watermark bits are calculated to generate *W'*. In case, *W* and *W'* match, the integrity is verified, otherwise, it is clear that the data has been tampered. Since order of tuples is treated as watermark, owner may be confused about integrity, in case, user accidentally changes order of the tuples. Moreover, attribute modification has probability of half in terms of detection [10].

To overcome this problem, authors have improvised this technique by integrating it with patch-work method [10]. According to this, groups of all the sorted tuples *w* are created based on LFSR states. Only those tuples which are between [1, w] should be chosen to form two groups, one group *A* contains odd positioned tuples, and other group *B* contains even positioned tuples. For each element in each group, new hash values *h'(A)* and *h'(B)* are calculated based on some random number *r* and old hash values *h(A)* and *h(B)*. Finally, the value of *d* is computed from difference between *h'(A)* and *h'(B)*for all elements. The database is ordered again based on generated watermark *W* by switching the position of every tuple pair *i* and *i-1 mod w* for *W[i]*=1.During watermark verification, value *d'* is computed in the similar fashion as *d* is obtained in watermark embedding phase. If *d* and *d'* match, then, database integrity is verified, else, tuples of the database have been tampered. However, in case, the tuples which are not in the group are modified; then, such modification may not be detected.

4.1.2. Using factorial format watermark to generate secret ordering among entries

Kamel proposed R-tree based watermarking in [11]. Its aim is to verify integrity of the database relation R. The basic idea is to establish one-to-one mapping between entries in R-tree nodes and all the values of watermark W. This can be achieved by factorial based number system. In contrast to B-tree, R-tree has no condition on arrangement of entries inside the node. The order of entries in R-tree is determined by left circular shift operation performed with respect to MSB of watermark W and the first reference order ER. Corresponding watermark W' is obtained by using embedded watermark EW and hypothetical reference order ER'. The value of W' is to be stored for each node. If the secret order among the values is disturbed, then, it can be considered as an attack on database integrity. However, this algorithm is based on the assumption that the data is uniformly distributed.

In [12], the authors have extended the same idea of shuffling the values to hide factorial format watermark in relative order of data as seen in R-tree based watermarking[11]. The only difference between [11] and [12] is that the watermark is used to secretly order entries in R-tree in the former whereas it is used for secret grouping of tuples of the database in the later. The scheme proposed in [12] depends on choosing a sensitive attribute which needs to be protected and sorting the tuples in ascending order based on this attribute. The subset $G[x,y]$ is shuffled using left circular shift operation by the value $W_F[i]$ to get watermarked group G_W. If change in value is too small to disturb the secret order of tuples, then, it may neither be detected in [11] nor in [12]. Moreover, in case, the attack percentage is small, then, its detection rate is also low.

4.1.3. Registering encrypted watermark with certification authority

In [13], authors have proposed a technique that generates watermark WM' as a white image besides four corners using owner's mark. This image is made up of N tuples with dimension $\lceil\sqrt{N}\rceil$ X $\lceil\sqrt{N}\rceil$. MD5 hash

algorithm is applied on all tuples to get fixed 128 bits length Mi which is bisected into parts with first 64 bits as bi and next 64 bits as fi. XOR operation is performed on bi and fi to get Xi. To control the feature Ci in the range of grey scale [0-255], (Xi mod 256) operation is performed. All features Ci are fetched and combined in order to produce C. A certification code R is generated by taking XOR operation of C and WM'. R is then changed to a $\lceil\sqrt{N}\rceil$ X $\lceil\sqrt{N}\rceil$ grey scale image. Finally, the certification image SD is encrypted with the help of secret key Skey and made public in the network. In order to verify database integrity, the certification image SD can be decrypted using owner's public key Pkey to determine certification code R. The parameters Mi', bi', fi', Xi' and Ci' can be obtained as mentioned in watermark generation phase. Like C, C' is generated by combining all fetched Ci' in order. The watermark WM" is fetched by performing XOR operation on C' and R. If embedded watermark WM' and extracted watermark WM" match, then, database integrity is verified, else, database has been tampered. However, if any tuple value apart from extracted 128 bits used for watermark generation is altered, then, such change would not be detected.

In 2013, a sub-watermark based scheme was proposed in [14]. According to this scheme, sub-watermarks are generated for digit count ωd, length ωl and range ωr of data values assuming that the database contains numeric attributes. During digit sub-watermark generation, length of each value $r_i A_j$ in database is calculated. For each numeric value, digit frequency of all digits (0-9) is determined using *Mid$()* function. Digit sub-watermark ω_d is created by finding relative frequency for each digit rfd_i and concatenating it with total digit count. Likewise, length sub-watermark ω_l can be obtained from relative frequency for length rfl_j of data values. For range sub-watermark ω_r, different ranges of data values are defined and frequency for each data value is calculated accordingly. Range sub-watermark ω_r is generated by computing relative frequency for each range of data value rfr_k and total range count. The final watermark ω_R obtained by concatenating ω_d, ω_l and ω_r for relation R is encrypted using secret key to get ω_c which is then registered with certification authority *CA*. In case the database *R'* is suspected, then, ω_c is retrieved from *CA* and is decrypted to get ω_R. If this ω_R and $\omega_R^{'}$, generated from *R'* as described earlier, are different, then the database is tampered. The type of modification can be categorized into digit, length and range by calculating fractional change in respective frequencies. However, swapping digits in some data values will not modify digit count, length and range. For example, data value 123 can be changed to 321 without affecting any of the above frequencies. In such case, tampering cannot be detected.

Data partitioning in [15] is performed in similar way as described in [5] except that the concept of grouping in [15] is square matrix based. According to this technique, database is partitioned into ν groups based on primary key hash value and then sorted accordingly. During watermark generation, authors in [15] have proposed an approach which obtains group watermark W_j by computing the determinant D_j of group G_j and the minor M_i^j of the j^{th} diagonal. All watermark groups are concatenated to get watermark W_R for database relation *R*. Like [14], W_R is encrypted and registered with *CA*. During watermark verification from suspicious database *R'*, *Wc* is obtained from *CA* and decrypted to get W_D. The group watermark $W_j^{'}$ is computed in same way as W_j was obtained. If W_j and $W_j^{'}$ are different, then, database has been tampered. Although the properties of square matrices can be exploited to localize modifications, some changes which do not affect determinant and minor values, will not be detected.

4.2. Robust Watermarking for Ownership Protection

Suppose Alice is the owner of database relation R. She embeds some watermark information W into R using her own private key K. She then distributes database relation R for public use. Later, suppose she suspects some relation S published by Mallory has been pirated from her relation R. To claim her ownership, Alice extracts watermark information W' from relation S and matches with earlier inserted watermark information W. As W and W' are same, Alice can claim that the database S, published by Mallory, is pirated version of original database R [1].

4.2.1. Verifiable through Public Key

Li and Deng in [16] suggests that the watermark key K should be chosen based on cryptographic hash function h() over owner's identity ID, database name DB_name, database version version and other required parameters as defined in (1).

$$K = h(ID \mid DB\_name \mid version \mid ....) \quad (1)$$

In identity based cryptography [17], above generated key is called public key. Using this key, the public watermark *W* is created with same number of tuples η and same primary key *P* as that of relation *R*. Watermark generation parameter γ is equivalent to number of binary attributes in watermark $W(P, W_0, W_1,..W_{\gamma-1})$. It is used to determine number of bits ω in *W*, where, $\omega = \eta \cdot \gamma$ *and* $\gamma \leq$ *number of attributes in R.* During watermark generation, a cryptographic pseudorandom number generator *G* takes as input the watermark key *K* and primary key of relation *r.P* and gives sequence of numbers as output. The MSBs of these selected attribute values are used for generating watermark *W*. If the owner of the database suspects some relation *R'* that has been pirated from her relation *R*, then, the watermark *W* can be identified by counting the number of matches between detected MSBs from *R'* and corresponding bits in *W*. The MSBs can be extracted from the attributes of *R'* selected in the similar manner as in watermark generation. However, there is a trade-off between watermark generation parameter γ and time and space overhead. Likewise, increasing watermark detection parameter τ would make false miss and false error rate worse. The advantage of this scheme is that it can be verified in public as many times as required. The verification process is based on watermark certificate which enlists watermark key and hash of watermark used in watermark detection.

4.2.2. Verifiable through Private Key

Authors in [18] have proposed a scheme that partitions the relation *D* into *m* non- overlapping partitions by key-hashed *MAC* function. For each partition S_k using private key *R*, watermark W_k is generated in same manner as described in [16]. The difference between techniques in [16] and [18] is that the former uses public key *K* and the later uses private key *R*. Moreover, the watermark W_k in [18] is interpreted as binary image with same number of rows but one less number of columns of actual partition. It is treated as abstract counterpart of the concrete relation *R*. All the detected MSBs should be completely matched with the binary values in watermark W_k. Even a single mismatch cannot prove ownership whereas percentage of matches in [16] depends on adjustable watermark detection parameter τ for ownership proof. The percentage of matches should be more than τ which is in range 0.5 to 1.

The technique proposed in [19] is a combination of [6] and [18] for integrity verification and ownership proof respectively. In [19], m MSBs and n LSBs are extracted from the attributes which are identified by applying cryptographic pseudorandom generator *G* on private key *R* and primary key *r.P*. The watermark W_k is generated by concatenation of *m MSBs and* n LSBs such that *m+n=8*. As the maximum value that can be obtained from 8 bits is 256 which belong to range 0 to 255, the watermark W_k for partition S_k can be interpreted as grey scale image. In [18], instead of grey scale image, binary image is formed by using only one MSB bit 0 or 1from the attributes. However, the images generated both in [18] and [19] do not have any meaningful information which can be used as ownership proof. Moreover, a single mismatch cannot prove ownership, there has to be 100% match between watermark *W'* obtained from *R'* and original watermark *W*.

In [19], the watermark *W* comprised of all image fragments $W_0, W_1, \ldots W_{m-1}$ is interpreted as abstract counterpart of concrete table *D*. Myrvold and Ruskey's linear permutation ranking algorithm [7] is used to derive an equivalent image *W'* which is compared with *W* obtained from unranking algorithm. This is same as that of [6]. Finally, if *W* and *W'* match, then, the integrity of the database can be verified, otherwise, not.

5. Analysis of Distortion-Free Watermarking Techniques

5.1. Parameters for Fragile Watermarking Techniques

Following parameters can be used to analyze the fragile watermarking techniques used for tamper detection:

- 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.
- 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.
- 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 must be detected by fragile watermarks. This type of processing are performed unintentionally.
- Subset Attack: Mallory 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 is unable to identify such attack.
- Superset Attack: Some new tuples or attributes are added to a watermarked database which can affect the correct detection [4].

Table I summarizes classification and comparison between various fragile watermarking techniques [5], [6], [8], [10], [11], [12], [13], [14] with respect to above parameters.

TABLE I: Comparison of Fragile Watermarking Techniques for Tamper Detection

Proposed Technique	Suitable Attribute Type	Granularity Level	Watermarking Key	Probability of Tampering Detection			Massive Tampering Detection
				Single tuple insertion	***Single tuple deletion***	***Single attribute value modification***	
Hash-based							
Group hash-based secret ordering among tuples [5]	Categorical	Group	Private	$Prob = 1 - \frac{1}{2^{\frac{\nu+1}{2}}}$	$Prob = 1 - \frac{1}{2^{\frac{\nu-1}{2}}}$	$Prob = 1 - \frac{1}{2^{\frac{\nu}{2}}}$	Yes
Hash-based grouping and permutating for secret ordering among tuples [6]	Categorical	Group	-	$Prob = 1 - \frac{1}{2^{\ln(q_{k+1})}}$	$Prob = 1 - \frac{1}{2^{\ln(q_{k-1})}}$	$Prob = 1 - \frac{1}{2^{\ln(q_k)}}$	No
Hash-based grouping and permutating for secret ordering among tuples as well as attributes [8]	Any	Group	Private	Not specified	Not specified	Not specified	No
Grouping based on hash values and LFSR states for secretly ordering tuples[10]	Any	Group	Private	Not specified	Not specified	Not specified	No
Factorial-based							
Shuffling values using left-circular shift for secret ordering among entries in R-tree nodes [11]	Any	Group	-	$P = 1 - \frac{kh-h}{kh}$	$P = 1 - \frac{kh-h}{kh}$	$P = 1 - \frac{kh-h}{kh}$	Yes
Organizing and shuffling tuples among group for generating secret order among tuples [12]	Any	Group	-	Not specified	Not specified	Not specified	No
Certification-based							
Extracting database features for generating grey scale image [13]	Any	Data value	Public	Not specified	Not specified	Not specified	Yes
Calculating frequency of digits, length and range of data values for generating sub-watermarks [14]	Numeric	Digit, length or range	Private	Not specified	Not specified	Not specified	Yes
Computing watermark group based on determinant and minor of square matrices [15]	Numeric	Square matrix group	Private	$P(S) = 1 - \left(\frac{4*\gamma}{(\gamma^2)}\right)^{(\alpha/\gamma)^k}$	$P(S) = 1 - \left(\frac{4*\gamma}{(\gamma^2)}\right)^{(\alpha/\gamma)^k}$	$P(S) = 1 - \left(\frac{4*\gamma}{(\gamma^2)}\right)^{(\alpha/\gamma)^k}$	Yes

5.2. Parameters for Robust Watermarking Techniques

Robust watermarking techniques can be analyzed on the basis of following enlisted parameters:

- Public System: The watermarking system should assume that the method used for inserting a watermark is public. Defence must lie only in the choice of the private parameters.
- False Hit: It is the probability of a valid watermark being detected from un-watermarked data. False hit should be negligible.
- False Miss: It is the probability of not detecting a valid watermark from watermarked data that has been modified in typical attacks. It should be negligible.
- 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 [4].

Table II illustrates comparison between robust watermarking techniques [16], [18], [19] with respect to above parameters.

TABLE II: Comparison of Robust Watermarking Techniques for Ownership Protection

Proposed Technique	Suitable Attribute Type	Granularity Level	Detectability	Robustness Probability		Incremental Updatability
				False hit	*False miss*	
Publically verifiable						
Use MSB from selected attribute values to generate watermark [16]	Any	Tuple	Public	$H = C_{1/2}(\tau\omega, \omega)$	$M_\varsigma = \begin{cases} 1 & if \quad \varsigma \geq \gamma\eta - \tau\gamma\eta \\ 0 & otherwise \end{cases}$	Yes
Privately verifiable						
Use MSB from selected attribute values to generate binary image as watermark [18]	Any	Group	Private	$F_h = B\left(\omega, \frac{\omega}{\tau}, \frac{1}{2}\right)$	$F_m = B\left(\upsilon\varsigma, \frac{\upsilon\varsigma}{\tau}, \frac{1}{2}\right)$	No
Use MSB from selected attribute values to generate grey scale image as watermark [19]	Any	Group	Private	$F_h = B\left(\omega, \frac{\omega}{\tau}, \frac{1}{2^8}\right)$	$F_m = B\left(\upsilon\varsigma, \frac{\upsilon\varsigma}{\tau}, \frac{1}{2^8}\right)$	No

6. Conclusion

In this paper, we have investigated the current distortion-free watermarking techniques for relational databases. These techniques are classified into fragile and robust according to the purpose they are used for. Most of the fragile techniques are based on creating groups of tuples or entries and ordering them secretly. The watermark information is used to decide the secret order. Moreover, they can detect modifications upto few tuples whereas massive tampering may be undetected. Robust techniques have gained comparitively less attention. Almost all of them extract MSB from the attribute values to generate the watermark. Finally, we observe that stronger watermarking algorithms need to be developed to achieve broader coverage of parameters. Hence, making them more reliable.

7. References

[1] Rakesh Agrawal and Jerry Kiernan, "Watermarking Relational Databases," *28th VLDB Conference*, pp. 155-166, 2002. https://doi.org/10.1016/B978-155860869-6/50022-6

[2] Michael Arnold, Martin Schmucker, Stephen D. Wolthusen, "Techniques and Applications of Digital Watermarking and Content Protection," Artech *House: Bouston*, pp. 21, 2003.

[3] E. T. Lin and E. J. Delp, "A review of fragile image watermarks," *Multimedia and Security Workshop (ACM Multimedia' 99)*, Orlando: FL, pp. 25–29, 1999.

[4] Raju Halder, Shantanu Pal and Agostino Cortesi, "Watermarking techniques for relational databases: survey, classification and comparison," *Journal of Universal Computer Science*, vol.16, no. 21, pp. 3164-3190, 2010.

[5] Y. Li, H. Guo, S. Jajodia, "Tamper detection and localization for categorical data using fragile watermark," *4th ACM workshop on Digital Rights Management (DRM' 03)*, Washington, pp. 73-82, 2003.

[6] Sukriti Bhattacharya and Agostino Cortesi, "A distortion free watermark framework for relational databases," *4th International Conference on Software and Data Technologies (ICSOFT' 09)*, Bulgaria, pp. 229-234, 2009.

[7] W. Myrvold and F. Ruskey, "Ranking and unranking permutations in linear time," *Information Processing Letters*, vol. 79, no. 6, pp. 281-284, 2001.

https://doi.org/10.1016/S0020-0190(01)00141-7

[8] Jie Guo, "Fragile watermarking scheme for tamper detection of relational database," *IEEE 2011 International Conference on Computer and Management (CAMAN)*, China, pp. 244-247, 2011.

https://doi.org/10.1109/CAMAN.2011.5778907

[9] R. Halder, P. Dasgupta, S. Naskar and S. Sarma, "An Internet-based IP Protection Scheme for Circxuit Designs using Linear Feedback Shift Register(LFSR)-based Locking," *22nd Annual Symposium on Integrated Circuits and System Design*, Brazil, pp. 84-94, 2009.

[10] R. Arun, K.Praveen, D. Bose and H. Nath, "A distortion free relational watermarking using patch work method," *InConINDIA, Springer-Verlag,* Berlin, pp. 531-538, 2012.

https://doi.org/10.1007/978-3-642-27443-5_61

[11] Ibrahim Kamel, "A schema for protecting the integrity of databases," *Computers and Security*, pp. 698-709, 2009.

https://doi.org/10.1016/j.cose.2009.04.001

[12] I. Kamel, W. Yaqub and K. Kareem, "An empirical study on the robustness of a fragile watermark for relational databases," *IEEE 2013 9th International Conference on Innovations in Information Technology*, UAE, pp. 227-232, 2013.

https://doi.org/10.1109/Innovations.2013.6544423

[13] M. Tsai, H. Tseng and C. Lai, "A database watermarking technique for temper detection," *2006 Joint Conference on Information Sciences*, Taiwan, 2006.

https://doi.org/10.2991/jcis.2006.206

[14] A. Khan and S. Husain, "A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations," *The Scientific World Journal*, 2013.

https://doi.org/10.1155/2013/796726

[15] L. Camara, J. Li, R. Li and W. Xie, "Distortion-free watermarking approach for relational database integrity checking," *Mathematical Problems in Engineering*, vol. 2014, 10 pages, 2014.

[16] Y. Li and R. Deng, "Publically verifiable ownership protection for relational databases," *2006 ACM Symposium on Information, Computer and Communication Security (ASIACCS ' 06)*, Taiwan, pp. 78-89, 2006.

[17] Adi Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Advances in Cryptology: Proceedings of CRYPTO' 84, Lecture Notes in Computer Science*, pp. 47-53, 1984.

[18] Sukriti Bhattacharya and Agostino Cortesi, "A generic distortion free watermarking technique for relational databases," *5th International Conference on Information Systems Security (ICISS' 09)*, India, pp. 252-264, 2009.

[19] Sukriti Bhattacharya and Agostino Cortesi, "Database authentication by distortion free watermarking," *5th International Conference on Software and Data Technologies (ICSOFT' 10),* Greece, pp. 219-226, 2010.