CHAPTER 2

Ownership and Tamper Detection of Relational Data: Framework, Techniques and Security Analysis

Vidhi Khanduja, Shampa Chakraverty and O.P. Verma

Databases play a pivotal role in all domains of technology, encompassing data mining, medical records, stock market data, e-commerce etc. With this elevated need for databases and their wide distribution in the web sphere, their security has become a major concern today. It is in this context that watermarked protection of databases has started receiving increasing attention from researchers. We present two of the most important security concerns related to protection of relational database: (i) Ownership proof and (ii) Tamper detection. We explain the functions of various components of a Robust Watermarking Model (RWM) to resolve ownership issues and a Fragile Watermarking Model (FWM) to deal with Integrity issues. We discuss various attacks that are possible on databases and present the appropriate security solutions to overcome them.

Vidhi Khanduja, Shampa Chakraverty
Department of Computer Engineering, NSIT
Delhi,India
e-mail: vidhikhanduja9@gmail.com

O.P. Verma
C.S.E Department, DTU
India
e-mail: opverma.dce@gmail.com

Editors: C.W. Ahn, M. Ali, M. Pant, *Embodying Intelligence in Multimedia Data Hiding*

DOI: 10.15579/gcsr.vol5.ch2, GCSR Vol. 5, pp. 21-36, 2016

2.1 Introduction

In today's technology-driven world, databases are maintained and utilized by virtually each and every web application running on a host of electronic gadgets. Whether they are sold in pieces for data mining applications such as outsourced stock market data, consumer behaviour data, power consumption data and weather data or built and maintained in-house such as by e-commerce sites and hospitals for storing the medical history of patients, databases play a pivotal role in conducting businesses. However, businesses have become increasingly sensitive about the security of digital databases due to the barrage of internet-based attacks that are possible. Malicious or innocuous attacks on databases can obliterate sensitive information, incur legal procedures, cause financial losses and vilify a productive business environment.

It is in the context of the above scenario that watermarked protection of databases has started receiving increasing attention from researchers. Digital watermarking is a technologically protective self help measure that is widely used now. All watermarking techniques designed must satisfy the following properties [3, 2]:

1. **Imperceptibility:** The amount of perturbations that can be made to data, such that its usability is still maintained, defines imperceptibility. Encyclopedia Britannica has embedded watermarking by making small changes to population and surface area of countries. Since weather data can tolerate errors in daily temperatures of 1-2 degrees, Met departments embed watermarks in appropriate fields of their databases. The usability constraints of an application hint at where in a database, the watermark can be concealed. Even in the world of analog media, publishers of books of mathematical tables such as logarithm tables and astronomical ephemerides have been deliberately introducing small errors in their tables for centuries to identify pirated copies.

2. **Blind Watermarking:** A watermarking technique is blind if it does not require the original un-watermarked database for watermark extraction. Blind watermarking is also referred to as oblivious watermarking.

3. **Embedding effectiveness:** The watermarking system should successfully embed a watermark in a randomly selected database. The technique should not be specific to any particular database.

4. **Robustness:** This is the ability of the watermark to resist perturbations caused by benign or malicious attacks. For many copyright controlled applications, robustness is of major concern. The watermarks should be embedded in a manner such that the watermark can be recovered even after random alterations caused by different attacks.

5. **Security:** According to Kerckhoff's law [3], watermarking algorithms are publically known to everyone. Its security lies in the cryptographic key. A watermarking technique is secure if an attacker is not able to detect or remove the watermark even after knowing the algorithms for embedding and extraction.

6. **Low-complexity:** In any watermark scheme, the insertion, detection and extraction processes must employ efficient algorithms with low time complexity.

7. **Accuracy:** It is the probability of detecting our watermark in someone else's non watermarked relation. It is commonly referred as false hit. For a system to be accurate, this should be low.

8. **Incremental Updatability:** Watermarks should be incrementally up-datable; as the attribute values of the tuples in database are altered (added, modified or deleted) the watermark should be recomputed for only such modified tuples.

In this chapter we present the watermarking model for ownership proof and tamper detection of relational databases satisfying the above properties. The chapter discusses and analyses various existing techniques to make the model secure against attacks.

This chapter commences by presenting the state-of-the-art in the area of watermark digital databases. It then covers two most important security concerns related to relational database protection: (i) Ownership proof and (ii) Tamper detection. We present the Robust Watermarking Model (RWM) explaining its components for ownership proof in Section 2.3. In Section 2.4, we present a Fragile Watermarking Model (FWM) for tamper detection and discuss the functioning of various components involved in it. In Section 2.5, we discuss various attacks that are possible on databases and present the appropriate security solutions to overcome them. This chapter concludes by summarizing the state-of-the-art and discussing the future research scope in watermarking databases.

2.2 Literature Survey

The idea of watermarking relational databases was first proposed by Agrawal et. al. in 2003 [3, 2]. He proposed a robust technique to embed watermark in selected numeric attribute of the certain selected tuples. Various attacks possible on databases are also discussed in these papers. Later, authors of [7, 8, 20] have proposed their own techniques to enhance this work. In [7], Cui, Qin and Sheng have proposed weighted algorithm. The technique assigns the weight to attributes according to their significance. The advantage is that the attribute with high rank has more chance to be marked than less important ones. In [8], Farfoura et.al., have proposed time-stamping based protocol to resolve additive attacks. The embedding process was reversible so that original values can be regained after extraction. In [20], authors have proposed a technique that marks multiple attributes with varying number of candidate bit positions within a single tuple. This results in more potential locations for embedding and hence, watermark is scattered throughout the database.

Certain statistical properties based on watermarking techniques are proposed in the literature [21, 26, 27]. Here, modifications are permitted according to the constraints of statistical properties of the selected attribute. The use of optimization techniques like Genetic Algorithm (GA) and bacterial foraging algorithms (BFA) further enhances the security of such watermarking schemes [14, 21, 26]. In [21], BFA is utilized which gives nearly global optimal values for target attributes, bounded by data usability constraints. This scheme enhances robustness, efficiency and imperceptibility. To reduce execution time, the parameters of BFA are tuned experimentally.

In later works, watermarking techniques started targeting attributes of varied data types apart from numeric. This includes float [8], text [13, 19] and categorical data types [16]. An information-centric data recovery watermarking technique is proposed in [18]. The central idea in this paper was that for many practical applications, it is not the raw data per se, but the information conveyed by it that is relevant and must be preserved at all cost [18]. Robust watermarks resist modifications, thereby serving as copyright indicators. In contrast, fragile watermarks become noticeable after the slightest modification. They serve to prove the fact that tampering has occurred in a suspect database. The works of [5, 10, 11, 15, 22] exist in the literature proposing tamper detection schemes. In cases where fragility is the primary objective, the watermark acts as a signature of the entire database and is secretly concealed in it.

Recently in [9], the authors proposed a robust scheme based on circular histogram modulation that can also be used to verify the integrity of the database. This is achieved by adding two different watermarks, one for ensuring robustness and the second one for incorporating fragility and reversibility. In [12], the authors have surveyed the state-of-the-art in watermarking techniques for relational databases. They categorize the techniques based on their intent, granularity level, watermark information and cover type. The cover type describes the data type of the target attribute where a watermark bit is concealed.

2.3 Robust Watermarking Model for Relational Databases (RWM)

Robust techniques are applied to resolve ownership issues. The watermark is embedded into the database redundantly so that watermark can be extracted successfully even after alterations introduced by an attacker. We now explain the robust watermarking model (RWM) for piracy detection.

In order to detect piracy, firstly a watermark that represents some information associated with the owner is prepared. This is embedded robustly into the database with minimum data modification. Sometimes, the watermark is registered with a Trusted Third Party (TTP) to provide transparent security [8]. The components of RWM are shown in Fig.(2.1).

2.3.1 Watermark Preparator

This component is responsible for selecting and securing the watermark with a secret key before embedding it into the database. The watermark selected must represent the owner's identity in order to prove ownership conclusively. Some of the early researches in this domain lack this aspect, simply treating the watermark as a string of bits to be embedded [3, 2, 26, 27, 13, 29].

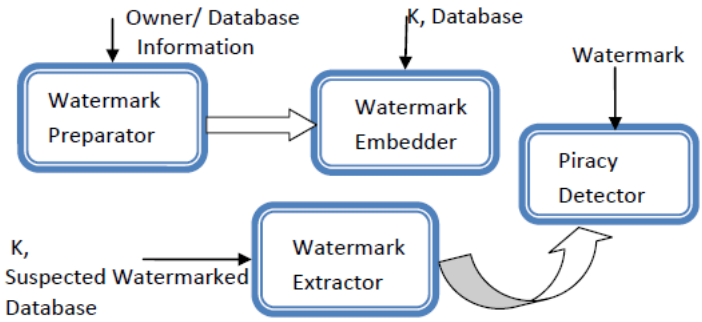

Figure 2.1: Components of RWM.

2.3.1.1 Watermark selection

The watermark is generally selected by the owner of the database. This selected watermark is made cryptographically secure and then the resulting ciphertext is embedded into database. If we embed the watermark say the owner's identity directly then an attacker can easily guess the watermark knowing the owner of the database. On guessing the closest watermark and the algorithm, it is not impossible to crack the keys used. Hence, an attacker can claim that this is his/her database by successfully extracting the correct watermark in case of dispute with the owner.

2.3.1.2 Watermark security

To secure the watermark, encryption is performed on selected watermark. For this purpose cryptographic techniques such as playfair encryption [16] or one-way hash functions can be used [18, 21]. Use of hash function is highly beneficial. Apart from security, hashing fixes the length of variable-sized watermark. This gives the owner the liberty to select his own watermark; one can select large files such as an image, audio or even video also to be embedded as watermark [21]. Finally, taking a hash to represent the shortened reference to original selected entity serves the purpose of creating a watermark effectively.

2.3.1.3 Watermark for identification

In order to prove ownership along with identification, the owner's biometric feature such as voice or fingerprint [17, 28] can be used to create a watermark. The choice of voice as the distinguishing factor is influenced by its low deployment cost, universality and ease of use. Technically, Equal Error rate of voice is very less i.e. less than 1% as compared to approximately 4% for iris and face [1]. Equal error rate is the decision threshold of a system at which false rejections will be approximately equal to the proportion of false acceptances. It is one of the best methods of determining the efficacy of the biometric method. In [17], authors generated a watermark by creating a statistical model of the features extracted from the owner's voice. This precludes the

owner from pre-registering his/her watermark and the associated registration costs. The extracted watermark identifies the real owner and provides a biologically secured technological measure for database protection saving the owner from the costs of pre-registering her watermark. Other techniques of preparing watermarks include image based watermarking techniques [4, 6, 24] and cloud theory [23, 29].

2.3.2 Watermark Embedder

This component is responsible for embedding the watermark into the database. The prepared watermark is embedded in algorithmically determined secret positions of selected candidate attributes in the database. It is obviously desirable that the error which is introduced due to embedding is kept to a minimum possible level. At the same time, the procedure must impart a degree of randomness to ensure greater protection against attacks. To introduce randomness, firstly the database is subjected to a virtual partitioning process. The position of each tuple in the database is virtually shuffled. Further selection of candidate embed positions is done according to this reordered arrangement. Selection of candidate positions may involve the following steps:

i. *Selection of tuples:* Some of the watermarking techniques embed watermark bits into all the tuples in the database [4, 26, 29], while some select a subset of the tuples to embed watermark bits [3, 2, 7, 8, 20, 17, 23, 24, 28]. Embedding watermark into all tuples introduce more distortion in the database as errors will now be introduced in all the tuples. However, using more potential locations for embedding watermark bits adds a higher level of redundancy, thereby increasing robustness.
The choice of tuple selection depends on many factors like the number of tuples in the database, the total length of the watermark, the number and types of attributes, application domain etc. The application domain where a database is applicable decides its usability constraints. These are constraints that must be imposed on each attribute so that it can tolerate the perturbations introduced due to embedded watermark bits without affecting its intended applicability [21]. Usability constraints are pre-selected by the owner of a database and are different for different applications and the corresponding data. Some examples of constraints include (i) uniqueness - each value must remain unique (ii) scale - the ratio between any two numbers before and after the change must remain the same and (iii) classification - the objects must remain in the same class as determined by a range of values before and after watermarking [27]. These constraints define the feasible space for manipulation.

ii. *Selection of attributes:* After tuple selection, the next step is to select the attribute(s) where watermark bits can be embedded. The owner selects a set of candidate attributes out of which a single attribute or multiple attributes for each record may be selected, depending on their usability constraints. Figure 2.2 diagrammatically represents the selection methods and Table 2.1 describes the symbols used thereafter. More numbers of

Table 2.1: Symbol notations.

Symbol	Description
N_t	Number of tuples in database
N_a	Number of attributes in database
N_c	Number of candidate attributes selected by owner
N_p	Number of partitions in Database selected by owner
N_w	Number of bits in Watermark
$W[i]$	i-th bit of Watermark

attributes renders more potential locations thereby concealing more watermark bits. It must be noted that with the increase in potential locations, perturbations in the database also increases. Hence, a balance between the two is necessary.

iii. *Selection of bit positions in selected attribute:* Once the attributes are allocated, the exact bit positions where watermark bit is to be concealed must be decided. The process can use one bit per attribute or multiple bits per attribute depending on usability constraints of the attribute selected. The bit where watermark is embedded is secretly decided among some least significant bit (lsb) positions to ensure minimum distortions. Various techniques employ different methods to embed the full watermark; either the entire watermark is concealed in each partition or only one watermark bit is repeatedly hidden in a given partition.
In case of sensitive database applications which cannot tolerate distortions such as military and medical fields, one must follow reversible techniques [8, 9, 14, 17]. Here, the error introduced during embedding phase is reverted back once it reaches its intended user. Such watermarks are referred to as invertible or erasable watermark. Depending upon the sensitivity of database, the watermark embedded is registered with trusted third party and then released on the network. In case of dispute, the embedded watermark sequences are extracted in the next component. Following this, a decision is reached about the ownership in the final component Piracy Detector.

2.3.3 Watermark Extractor

This component is responsible for extracting the watermark from the watermarked database at the receiver's end. Generally, extraction is the reverse of embedding. For robust watermarking, the bits are embedded multiple times. Hence, each bit is extracted multiple times. The extracted bits are input to a majority voter which counts the number of times $'1'$ and $'0'$ are extracted for the same bit watermark position. If the number of ones extracted is more than the number of zeroes, the majority voter assigns the final watermark bit as 1 for that position and vice versa. Majority voting

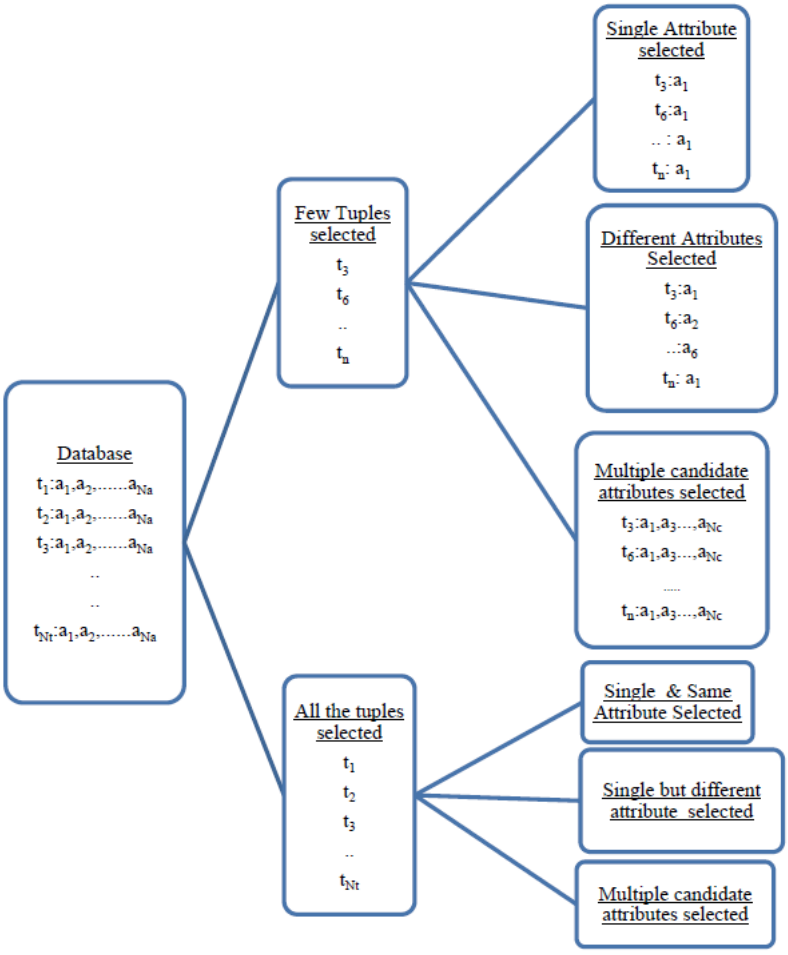

Figure 2.2: Methods to select attributes.

is used in nearly all techniques for robust watermarking to extract the bits correctly [7, 8, 9, 14, 17, 24, 26, 27, 28].

As a further step towards watermark recovery, error correcting codes can be used to accurately extract the watermark bits. Extra bits required for error correction are embedded along with the watermarking bits during embedding. At the time of extraction, using these error correcting bits one can rectify the errors introduced accidentally during transmission or deliberately by an attacker.

2.3.4 Piracy Detector

The watermark bits extracted from a suspected database are matched with the corresponding bits of the original watermark. If the number of matches m is very few or very large we suspect piracy. To assess very few and very large quantities, a threshold τ must be set up. A small value $a \in (0, 1)$ is used as a significance level of test and the pre-set threshold value τ is found by Eq.(2.1) [2]:

$$\tau = \max\{t \in [0, N_w/2]\} : \sum_{i=t}^{N_w - t} b(i; N_w, 1/2) \geq 1 - a \tag{2.1}$$

where

$$b(k; n, p) = \frac{n! p^k (1-p)^{n-k}}{k!\,(n-k)!}.$$

We suspect piracy if either $m < \tau$ or $m > \omega - \tau$, as the probability of so few or so many matches is less than or equal to a.

2.4 Fragile Watermarking Model (FWM)

Fragile watermarking techniques aim at tamper detection. The watermark usually acts as a signature of the database. Even the slightest alterations made to a database, deliberately or otherwise, will immediately alter its signature and hence its watermark. The tampering effect can therefore be detected. We present a fragile watermarking model (FWM) and discuss the functioning of various components involved in it. The components of the FWM are shown in Fig.(2.3).

2.4.1 Watermark Preparator

A watermark is created by using all the values of each and every datum in the database. For creating a watermark, the signature of the database is generated by concatenating all attribute values and applying a secure hash function on it using a secret key. Next, the hashes of all tuples (tuple hash) are concatenated to get the final database hash. The secure hash function not only introduces randomization but also generates a fixed length output from a variable length input. Any suitable hash functions like MD5 or SHA-512 can be applied [25]. In [15], authors have proposed watermark generation based on content characteristics of numeric data values. Watermark consists of relative frequencies of digits in all the data values, relative frequency of each length of data

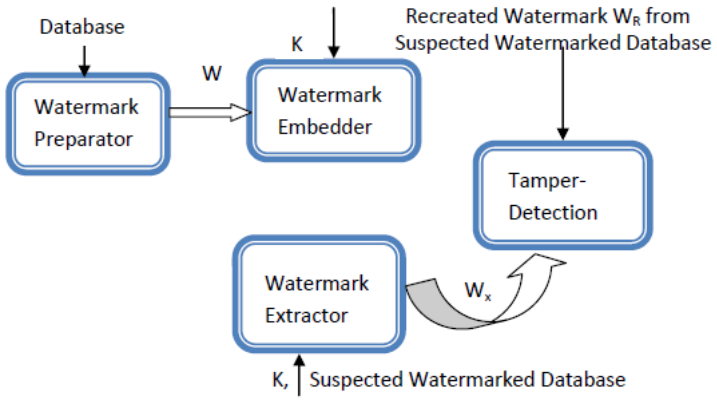

Figure 2.3: Components of FWM.

value and relative frequency of each range of data value. This helps in characterizing malicious modifications in the database.

2.4.2 Watermark Embedder

The prepared watermark is concealed within the database. Potential locations to hide the watermark are secretly decided by the database owner. These are generally *lsbs* of the important attributes. Such embedding *lsb* positions must obviously be excluded while preparing the watermark. The watermark is embedded once to attain tamper detection. In fragile watermarking, the embedder may or may not use database partitioning. If the database is partitioned, then the watermark of different partitions is created separately and embedded into the database. Such techniques distort the database by embedding the watermark [10].

A distortion-free embedding technique is proposed in the literature [5, 11, 15, 22]. In [22], the watermark is embedded by re-ordering the tuples. The database is partitioned and each partition is watermarked independently. The partition hash is computed for each partition separately that acts as a watermark for that particular partition. The length of watermark is made equal to the number of tuple pairs in a partition. To embed the watermark, in each tuple pair, their tuple hash values are compared. If the watermark bit is $'1'$, the tuples are arranged in decreasing order, according to their tuple hash, else in increasing order.

In [5, 15], authors proposed a zero-watermarking technique by not embedding the watermark into the database. The generated watermark is registered with the Certification Authority (CA) for certification purpose. This averts watermarking from being a complete self-help measure.

2.4.3 Watermark Extractor

Whenever the database is freely available on network, there exists the menace of integrity attacks. An affected database is a suspected database. The watermark extractor is responsible for reliably extracting the embedded watermark from a suspected database. This involves the reverse of embedding procedure. The extracted watermark is passed to the next component for tamper detection.

2.4.4 Tamper Detection

Tamper detection takes place at the receiver's side. Watermark W_X extracted from the suspected database is compared with the watermark W_R re-generated from the same suspected database to detect temperedness. Let the original embedded watermark be W. We now consider the following cases:

1. There were no integrity attacks. If neither the data nor the watermark were changed, then $W_R = W$ and $W_X = W$. Thus, the two watermarks will match $(W_R = W_X)$.

2. The content of the database was changed but not the embedded watermark. Then, re-generated watermark will not be the same as the original watermark, i.e. $W_R \neq W$. Thus, $W_X \neq W_R$, and the tampering event will surely be detected.

3. The raw data in the database was not changed but the embedded watermark was changed. Hence $W_R = W$ but due to tampering, W_X changes. Hence, $W_X \neq W_R$, and the tampering is detected.

4. Both the content of the database as well as the embedded watermark were changed. This case has a very remote chance of the two new watermarks produced as a result of the changes to the database and the embedded watermark respectively, turns out to be exactly the same. Hence, $W_X \neq W_R$ and the tampering is detected.

2.5 Security Analysis

The security of embedded watermarks must be properly ensured since any damage to the watermark or the database can compromise with the authenticity of the watermarked document and result in loss of crucial information. Assume that Alice is the owner of a database R. The watermark W is embedded to generate a watermarked database R_w. We assume that attacker Mallory does not have access to R and has no knowledge of the secret information used in embedding the watermark. Mallory may try to execute malicious attacks or benign updates on a watermarked database with the intention of distorting the embedded watermark. Following are malicious attacks which Mallory can perform on R_w [12, 21, 26].

Subset Addition Attack: In this attack Mallory adds certain tuples to the existing database with an intention to destroy the watermark. These added tuples act as noise

in the database. Watermarking technique should ensure that even after adding tuples, the owner should be able to successfully extract the watermark from the database.

Newly added tuples may belong to any of the virtual partitions [17]. Now, if there is uniform partitioning, almost equal numbers of tuples will belong to all the partitions. With random additions, one can assume that noise is also equally distributed amongst the partitions. The technique where a single watermark bit is embedded repeatedly in a partition is highly resilient towards subset addition attacks. Owing to redundancy of the same watermark bit in a partition, a majority voting will output the more frequent value for each watermark bit position. The watermark will start getting spoiled only if (i) the noise reaches more than 50% of tuples in a partition and (ii) a majority of the bits extracted from noise turns out to be the inverse of the embedded bit. Note that a partially extracted watermark can still be utilized to match the original.

Subset Deletion Attack: In this attack, Mallory may delete certain tuples from the database to delete the watermark. To make the technique resilient to this attack, the watermark must be scattered throughout the entire database. The use of secret keys at various steps of embedding conceals the watermark as Mallory cannot guess the potential locations and can only randomly delete tuples from the database. Note that Mallory can delete tuples up to a certain limit after which the database itself becomes useless.

More redundant the watermark insertion, more resilient it is to subset deletion attacks. The watermark must be embedded into significant parts of the database which is inseparable from it.

The technique where a single watermark bit is embedded repeatedly in a partition is highly resilient towards this attack. After tuple deletion attack, if single tuple per partition is left, then watermark bit can be successfully extracted from that partition. Hence, complete watermark can be extracted from subsequent partitions.

Subset Alteration Attack: In this attack, Mallory may alter the values within database with an intention to distort the database. Since she does not know the potential locations, she may randomly alter the values within a database. However, Mallory can alter the values only within the usability constraints so that the data remains useful. Embedding the data in fixed *lsb* positions turns out to be a less resilient approach in this case. By altering all the *lsbs* Mallory can easily destroy the watermark. To counter this, one must select the actual embedding positions with a secure randomized function making it difficult for Mallory to anticipate the actual embedding locations.

Attribute Attack: In this attack, Mallory tries to distort an attribute with an intention to distort the watermark. She may add new attributes or delete existing ones or modify attributes randomly. Techniques which embed watermark into single selected attribute suffer from this attack. To counter this attack, the watermark must be distributed throughout the database profusely.

Tuple sorting Attack: In this attack, Mallory rearranges the tuples based on one or more attributes with an intention to distort the watermark. Distortion-free techniques based on tuple reordering are susceptible to this kind of attack. Partitioning based techniques, where tuples are virtually reordered before the watermark is embedded, are resistant to this attack.

Additive attack: In this attack, Mallory adds her own watermark to a watermarked

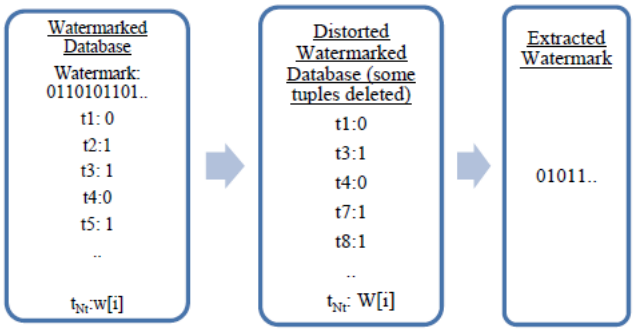

Figure 2.4: Synchronisation error.

relation. She then claims the ownership. To overcome such a situation, we propose the same solution as discussed by Agrawal et al. [2]. Both Alice and Mallory are asked to produce the original relation into which watermark is embedded. Mallory will present the database containing Alice's watermark. Now, Alice can easily extract her watermark from that relation and claim ownership while Mallory fails to do so.

Invertibility Attack: In the above case, if both Alice and Mallory are able to extract their respective watermarks from the original database, it leads to an invertibility attack. Mallory claims ownership by finding a key which extracts the embedded watermark from the pirated database fortuitously. The probability of this can be made close to zero by carefully selecting the values of secret parameters and length of watermark N_w.

Linear transformation Attack: An attacker can easily change a database without losing its usability by changing the units of attribute values. In such a case, the database is not considered distorted as the information conveyed by that attribute remains the same. Mallory may alter the attribute values by changing their measuring units. By doing so, she is at will to completely change the values however without losing any information. In doing so, the embedded watermark can get easily erased. An approach to avoid such attacks is to convert the attribute values to the smallest basic unit and then embed the watermark. Similarly, while extracting the watermark, the attribute is converted to the smallest basic unit and then watermark bit is extracted [21].

Synchronization Error: Whenever there is an attack (perturbation) on a watermarked database, some bits might be lost. While extracting, if some bits in between are missed, the extracted watermark bits arrive out of sequence. Watermarking schemes that use marker tuples are prone to synchronization errors [27]. Figure 2.4 shows this kind of error occurring from a tuple deletion attack. Similarly, in the case of tuple addition attack, newly added tuples in between the original tuples can change the order of the extracted watermark bits. To counter this, a single bit is embedded multiple times in one partition barring marker tuples.

2.6 Conclusions

In this chapter we have discussed the robust and tamper detection watermarking technique for Relational databases. We have presented the state-of-the-art in the domain of these two important security issues: (i) Ownership proof and (ii) Tamper detection. We explain the Robust Watermarking Model (RWM) to resolve ownership issues. We have analysed various existing ways to prepare the watermark and to embed the watermark robustly. To resolve integrity issues Fragile Watermarking Model (FWM) is proposed. Theoretical analysis of the tamper detection is done, analysing various cases w.r.t. watermark extracted and re-generated. We discuss various attacks that are possible on databases and present the appropriate security solutions to overcome them.

The area of digital watermarking is rife with challenges and ample research is still ongoing. While several robust watermarking techniques targeting numeric attributes have been reported in the literature, the potential of other non-numeric attributes types must be examined. The bulk of prior research focused on robust watermarking techniques. We now need to develop more fragile watermarking schemes for traditional databases as well as several non structured databases that are flourishing in the web.

Apart from relational databases, suitable techniques must be developed for watermarking Object Relational Databases (ORDBs) and No-SQL databases. ORDBs have recently gained attention due to their various advantages over conventional relational databases. An ORDB merges object oriented modelling techniques such as objects, classes, behavioural aspects of objects and inheritance into a relational database schema and makes use of a query language to extract information from the database. Applications that process multimedia data and require powerful querying capability profitably utilize an ORDB.

Relational Databases does not cope with the scale and agility challenges of modern applications. NoSQL provides this solution by encompassing different database technologies developed with a rise in the volume of data stored about users, objects and products, the frequency in which this data is accessed, and performance and processing needs.

Another area of work is biometrics-based watermarking. This will provide an additional means for safeguarding the interests of owners against illegal claims of ownership. Biometrics-based watermarking is well suited for the distributed applications that require authentication of participants, such as in case of collaborative software development. Given these promising research directions, we hope to see a boost in investigative efforts on digital watermarking with a thrust on database security.

References

[1] `http://www.authenticationworld.com/Authentication-Biometrics/index.html`.

[2] R. Agrawal, P.J. Haas, and J. Kiernan. Watermarking relational data: framework, algorithms and analysis. *The VLDB Journal*, 12(2):157–169, 2003.

[3] R. Agrawal and J. Kiernan. Watermarking relational databases. In *28th International Conference on Very Large Data Bases (VLDB)*, pages 155–166, 2002.

[4] A. Al-Haj and A. Odeh. Robust and blind watermarking of relational database systems. *Journal of Computer Science*, 4(12):1024–1029, 2008.

[5] L. Camara, J. Li, R. Li, and W. Xie. Distortion-free watermarking approach for relational database integrity checking. *Mathematical Problems in Engineering*, 2014(Article ID 697165):10 pages, 2014.

[6] X. Chen, P. Chen, Y. He, and L. Li. A self-resilience digital image watermark based on relational database. In *International Symposium on Knowledge Acquisition and Modeling (KAM)*, pages 698–702, 2008.

[7] X. Cui, X. Qin, and G. Sheng. A weighted algorithm for watermarking relational databases. *Wuhan University Journal of Natural Sciences*, 12(1):79–82, 2007.

[8] M.E. Farfoura, S.J. Horng, J.L. Lai, R.S. Run, R.J. Chen, and M.K. Khan. A blind reversible method for watermarking relational databases based on a time-stamping protocol. *Expert Systems with Applications*, 39(3):3185–3196, 2012.

[9] J. Franco-Contreras, G. Coatrieux, F. Cuppens, N. Cuppens-Boulahia, and C. Roux. Robust lossless watermarking of relational databases based on circular histogram modulation. *IEEE Transactions on Information Forensics and Security*, 9(3):397–410, 2014.

[10] H. Guo, Y. Li, A. Liu, and S. Jajodia. Fragile watermarking scheme for detecting malicious modifications of database. *Information Sciences*, 176(10):1350–1378, 2006.

[11] J. Guo. Fragile watermarking scheme for tamper detection of relational database. In *International Conference on Computer and Management (CAMAN)*, pages 1–4, 2011.

[12] R. Halder, S. Pal, and A. Cortesi. Watermarking techniques for relational databases: survey, classification and comparison. *Journal of Universal Computer Science*, 16(21):3164–3190, 2010.

[13] D. Hanyurwimfura, Y. Liu, and Z. Liu. Text format based relational database watermarking for non-numeric data. In *International Conference on Computer Design and Applications (ICCDA)*, volume 4, pages V4–312–V4–316, 2010.

[14] S. Iftikhar, M. Kamran, and Z. Anwar. RRW – a robust and reversible watermarking technique for relational data. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1132–1145, 2015.

[15] A. Khan and S.A. Husain. A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations. *The Scientific World Journal*, 2013(Article ID 796726):16 pages, 2013.

[16] V. Khanduja, S. Chakraverty, and O.P. Verma. Robust watermarking for categorical data. In *IEEE International Conference on Control, Computing, Communication and Materials (ICCCCM)*, pages 174–176, 2013.

[17] V. Khanduja, S. Chakraverty, O.P. Verma, and N. Singh. A scheme for robust biometric watermarking in web databases for ownership proof with identification. In *International conference on Active Media Technology (AMT)*, volume 8610 of *LNCS*, pages 212–225, 2014.

[18] V. Khanduja, S. Chakraverty, O.P. Verma, R. Tandon, and S. Goel. A robust multiple watermarking technique for information recovery. In *IEEE International*

Advance Computing Conference (IACC), pages 250–255, 2014.

[19] V. Khanduja, A. Khandelwal, A. Madharaia, D. Saraf, and T. Kumar. A robust watermarking approach for non numeric relational database. In *International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–5, 2012.

[20] V. Khanduja and O.P. Verma. Identification and proof of ownership by watermarking relational databases. *International Journal of Information and Electronics Engineering*, 2(2):274–277, 2012.

[21] V. Khanduja, O.P. Verma, and S. Chakraverty. Watermarking relational databases using bacterial foraging algorithm. *Multimedia Tools and Applications*, 74(3):813–839, 2015.

[22] Y. Li, H. Guo, and S. Jajodia. Tamper detection and localization for categorical data using fragile watermarks. In *4th ACM Workshop on Digital Rights Management (DRM)*, pages 73–82, 2004.

[23] H. Min, C. Jia-heng, P. Zhi-yong, and Z. Cheng. A new mechanism based on similar clouds watermark for database¡¯s information security. *Wuhan University Journal of Natural Sciences*, 9(4):415–419, 2004.

[24] H.M. Sardroudi and S. Ibrahim. A new approach for relational database watermarking using image. In *5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 606–610, 2010.

[25] B Schneier. *Applied Cryptography*. John Wiley, New York, USA, 1996.

[26] M. Shehab, E. Bertino, and A. Ghafoor. Watermarking relational databases using optimization-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):116–129, 2008.

[27] R. Sion, M.J. Atallah, and S. Prabhakar. Rights protection for relational data. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1509–1525, 2004.

[28] H. Wang, X. Cui, and Z. Cao. A speech based algorithm for watermarking relational databases. In *International Symposiums on Information Processing (ISIP)*, pages 603–606, 2008.

[29] Y. Zhang, X. Niu, and D. Zhao. A method of protecting relational databases copyright with cloud watermark. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 1(3):750–754, 2007.