Journal of Universal Computer Science, vol. 16, no. 21 (2010), 3164-3190
submitted: 18/12/09, accepted: 29/11/10, appeared: 1/12/10 © *J.UCS*

Watermarking Techniques for Relational Databases: Survey, Classification and Comparison

Raju Halder
(Università Ca' Foscari Venezia, Italy
halder@unive.it)

Shantanu Pal
(University of Calcutta, India
shantanu.smit@gmail.com)

Agostino Cortesi
(Università Ca' Foscari Venezia, Italy
cortesi@unive.it)

Abstract: Digital watermarking for relational databases emerged as a candidate solution to provide copyright protection, tamper detection, traitor tracing, maintaining integrity of relational data. Many watermarking techniques have been proposed in the literature to address these purposes. In this paper, we survey the current state-of-the-art and we classify them according to their intent, the way they express the watermark, the cover type, the granularity level, and their verifiability.

Key Words: Digital Watermarking, Fingerprinting, Relational Databases
Category: D.2.4, E.3, H.2.4

1 Introduction

The recent surge in the growth of the Internet results in offering of a wide range of web-based services, such as database as a service, digital repositories and libraries, e-commerce, online decision support system etc. These applications make the digital assets, such as digital images, video, audio, database content etc, easily accessible by ordinary people around the world for sharing, purchasing, distributing, or many other purposes. As a result of this, such digital products are facing serious challenges like piracy, illegal redistribution, ownership claiming, forgery, theft etc. Digital watermarking technology is an effective solution to meet such challenges. A watermark is considered to be some kind of information that is embedded into underlying data for tamper detection, localization, ownership proof, traitor tracing etc.

Initially, most of the work on watermarking was concentrated on watermarking of still images, video, audio, VLSI design etc [Lee and Jung, 2001], [Potdar et al., 2005], [Abdel-Hamid et al., 2004]. However, in the recent years watermarking of database systems started to receive attention because of the

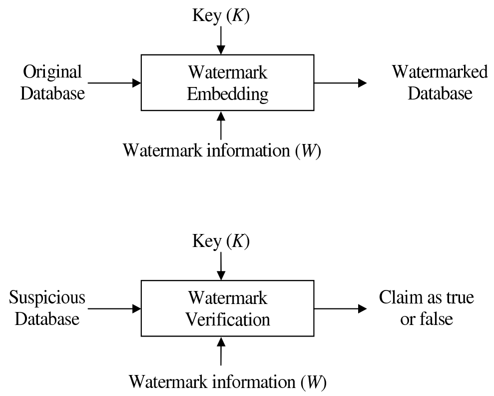

Figure 1: Basic Watermarking Technique

increasing use of it in many real-life applications. Examples where database watermarking might be of a crucial importance include protecting rights and ensuring the integrity of outsourced relational databases in service provider model [Hacigumus et al., 2002], in data mining technology where data are sold in pieces to parties specialized in mining it [Agrawal and Srikant, 2000], online B2B interactions [Hu and Grefen, 2002] etc. The idea to secure a database of map information (represented as a graph) by digital watermarking technique was first coined by Khanna and Zane in 2000 [Khanna and Zane, 2000]. In 2002, Agrawal et al. proposed the idea of digital watermarking for relational database [Agrawal and Kiernan, 2002].

In general, the database watermarking techniques consist of two phases: *Watermark Embedding* and *Watermark Verification*. During watermark embedding phase, a private key K (known only to the owner) is used to embed the watermark W into the original database. The watermarked database is then made publicly available. To verify the ownership of a suspicious database, the verification process is performed where the suspicious database is taken as input and by using the private key K (the same which is used during the embedding phase) the embedded watermark (if present) is extracted and compared with the original watermark information. Figure 1 depicts the basic database watermarking technique.

The relational data defers from multimedia data in many respects: (i) Few Redundant Data: Multimedia objects consists of large number of bits providing large cover to hide watermark, whereas the database object is a collection of independent objects, called tuples. The watermark has to be embedded into these tuples, (ii) Out-of-Order Relational Data: The relative spatial/temporal positions of different parts or components in multimedia objects do not change, whereas there is no ordering among the tuples in database relations as the collection of tuples is considered as set, (*iii*) Frequent Updating: Any portion of multimedia objects is not dropped or replaced normally, whereas tuples may be inserted, deleted, or updated during normal database operations, (*iv*) There are many psycho-physical phenomena based on human visual system and human auditory system which can be exploited for mark embedding. However, one can not exploit such phenomena in case of relational databases.

Due to these differences between relational and multimedia data, there exist no image or audio watermarking method which is suitable for watermarking of relational databases. These differences give rise to many technical challenges in database watermarking as well.

2 Applications of Digital Watermark for Relational Databases

Digital Watermarks for relational databases are potentially useful in many applications, including:

1. Ownership Assertion: Watermarks can be used for ownership assertion. To assert ownership of a relational database, Alice can embed a watermark into her database R using some private parameters (*e.g.* secret key) which is known only to her. Then she can make the watermarked database publicly available. Later, suppose Alice suspects that the relation S published by Mallory[1] has been pirated from her relation R. The set of tuples and attributes in S can be a subset of R. To defeat Mallory's ownership claiming, Alice can demonstrate the presence of her watermark in Mallory's relation. For such a scheme to work, the watermark has to survive intentional or unintentional data processing operations which may remove or distort the watermark.

2. Fingerprinting: Fingerprinting aims to identify a traitor. In the applications where database content is publicly available over a network, the content owner would like to discourage unauthorized duplication and distribution by embedding a distinct watermark (or fingerprint) in each copy of the database content. If, at a later point in time, unauthorized copies of the database are found, then the origin of the copy can be determined by retrieving the fingerprint.

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

[1] Conventionally, in cryptography literature, Mallory represents the malicious active attacker.

using a unique key associated with the source, and the integrity of the data is verified through the integrity of the extracted watermark.

3 Different Types of Attacks

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

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

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

2. Value Modification Attack:

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

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

   – Rounding Attack: Mallory may try to lose the marks contained in a numeric attribute by rounding all the values of the attribute. Success of this attack depends on the estimation of how many bit positions are involved in the watermarking. Underestimation of it may cause the attack unsuccessful, whereas overestimation may cause the data useless.

   – Transformation: An attack related to the rounding attack is one in which the numeric values are linearly transformed. For example, Mallory may convert the data to a different unit of measurement (e.g., Fahrenheit to Celsius). The unnecessary conversion by Mallory would raise suspicion among users.

3. 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 has been lost.

4. Superset Attack: Some new tuples or attributes are added to a watermarked database which can affect the correct detection of the watermark.

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

6. False Claim of Ownership: This type of attack seeks to provide a traitor or pirate with evidence that raises doubts about merchant's claim.
   - Additive Attack: Mallory may simply add his watermark to Alice's watermarked relation and try to claim his ownership.
   - Invertibility Attack: Mallory may launch an invertibility attack to claim his ownership if he can successfully discover a fictitious watermark which is in fact a random occurrence from a watermarked database.

7. Subset Reverse Order Attack: Attacker enjoys this attack by exchanging the order or positions of the tuples or attributes in relation which may erase or disturb the watermark.

8. Brute Force Attack: In this case, Mallory tries to guess about the private parameters (*e.g.* secret key) by traversing the possible search spaces of the parameters. This attack can be thwarted by assuming that the private parameters are long enough in size.

4 Watermarking Issues

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

- Capacity: It determines the optimum amount of data that can be embedded in a cover and the optimum way to embed and extract this information.
- Usability: The changes in the data of the database during watermarking process should not degrade the usability of the data. The amount of allowable change differs from one database to another, depending on the nature of stored records.
- Robustness: Watermarks embedded in databases should be robust against malicious or accidental attempts at removal without destroying the usability of the database.
- Security: The security of the watermarking process relies on some private parameters (*e.g.* secret key) which should be kept completely secret. Owner of the database should be the only one who has knowledge about them.
- Blindness: Watermark extraction should require neither the knowledge of the original unwatermarked database nor the watermark information. This property is critical as it allows the watermark to be detected in a copy of the database relation, irrespective of later updates to the original relation.
- Incremental Watermarking: After a database has been watermarked, the watermarking algorithm should compute the watermark values only for the added or modified tuples, keeping the unaltered watermarked tuples untouched.
- Non-interference: If multiple marks are inserted into a single relational database, then they should not interfere with each other.
- Public System: Following Kerckhoffs [Kerckhoffs, 1983], the watermarking system should assume that the method used for inserting a watermark is public. Defense must lie only in the choice of the private parameters (*e.g.* secret key).
- False Positiveness and False Negativeness: The false hit is the probability of a valid watermark being detected from unwatermark data, whereas false miss is the probability of not detecting a valid watermark from watermarked data that has been modified in typical attacks. The false hit and false miss should be negligible.

5 Classification of Watermarking Techniques

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

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

6 Watermarking Techniques

In this Section, we try to cover the details of various watermarking techniques proposed so far. We categorize the proposed techniques based on (*i*) whether marking introduces any distortion, (*ii*) the type of the underlying data (cover) in which watermark information is embedded, and (*iii*) the type of the watermark information to be embedded.

Based on whether the marking introduces any changes in the underlying data of the database, the watermarking techniques can be categorized into two: *Distortion-based* and *Distortion-free*.

6.1 Distortion-based Watermarking

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

6.1.1 Watermarking Based on Numerical Data Type Attribute

(a) Arbitrary meaningless bit pattern as watermark information.

The watermarking schemes proposed by Agrawal et al. [Agrawal et al., 2003a], [Agrawal et al., 2003b], [Agrawal and Kiernan, 2002] (also known as AHK algorithm) is based on numeric data type attribute and marking is done at bit-level. The basic idea of these schemes is to ensure that some bit positions for some of the attributes of some of the tuples in the relation contain specific values. This bit pattern constitutes the *watermark*. The tuples, attributes within a tuple, bit positions in an attribute, and specific bit values at those positions are algorithmically determined under the control of the private parameters γ, ν, ξ and K known only to the owner of the relation. The parameters γ, ν, ξ and K represent number of tuples to mark, number of attributes available to mark, number of least significant bits available for marking in an attribute, and secret key respectively. In [Agrawal and Kiernan, 2002], the cryptographic MAC function $H(K||H(K||r.P))$ where $r.P$ is the primary key of the tuple r and $||$ represents concatenation operation, is used to determine candidate bit positions. The HASH function $H(K||r.P)$ is used to determine the bit values to be embedded at those positions. The choice of MAC and HASH is due to the one-way functional characteristics and less collision probability. In [Agrawal et al., 2003a], [Agrawal et al., 2003b], authors use pseudorandom sequence generator ($e.g.$, Linear Feedback Shift Register [Halder et al., 2009]) instead of HASH and MAC to identify the marking bits and mark positions. The security and robustness of this scheme relies on these parameters which are completely private to the owner. The watermark detection algorithm is blind and probabilistic in nature. The relation is considered as pirated if the matching pattern is present in at least τ tuples, where τ depends on the actual number of tuples marked and a preselected value α, called the significance level of the test. Observe that the success of watermark detection phase depends on the fixed order of attributes. Re-sort of attributes' order may yield to the detection phase almost infeasible. Although the main assumption of this scheme is that the relation has primary key whose value does not change, they also suggest an alternative to treat a relation without primary key. Li et al. [Li et al., 2003] also suggest three different schemes to obtain virtual primary key for a relation without primary key.

Lafaye [Lafaye, 2007] describes the security analysis for the AHK algorithm where he analyzes the security and robustness in two situations: (i) Multiple Keys Single Database (MKSD): When a single database is watermarked several times using different secret keys and sold to different users, and (ii) Single Key Multiple Databases (SKMD): When several different databases are watermarked using a single secret key. An attempt of random attack on a watermarked content obtained by the AHK algorithm, may be successful when randomize the ξ^{th} least significant bits of all tuples of the relation. However, this attack is highly invasive since most values of the relation are impacted by the attack. The locations guessed based on MKSD and SKMD can be used to build a better focussed attack.

Qin et al. [Qin et al., 2006] suggest an improvement over the Agrawal and Kiernan's scheme [Agrawal and Kiernan, 2002]. Instead of using hash function, they use chaotic random series based on the Logistic chaos equation which has two properties: the non-repetitive iterative operation and the sensitiveness to initial value. It avoids the inherent weakness of collision of Hash function. The selection of bits of LSB for embedding watermark meets the requirements of both data range and data precision of each attribute, rather than simply to use a same ξ for all attributes. So the error caused by watermark is decreased significantly, hardly affects the availability of the database.

Among the most recent works, Gupta et al. [Gupta and Pieprzyk, 2009] propose a reversible watermarking scheme which is the modified version of Agrawal and Kiernan's one [Agrawal and Kiernan, 2002]. In this scheme, during the detection phase, the original unwatermarked version of the database can be recovered along with the ownership proof. The operation first extracts a bit $OldBit$ from the integer portion of the attribute value before replacing it by the watermark bit and inserts it in the fraction portion of the attribute value. Thus, the watermark bit can be recovered during detection and the attribute can be restored to its unmarked value by replacing the watermark bit with the original bit $OldBit$ extracted from the fraction part. They also propose another algorithm to defeat any attempt of additive or secondary attack which relies on the obvious fact that the database relation must be watermarked by the actual owner before Mallory.

The watermarking method in [Xiao et al., 2007] embeds random digits (between 0 to 9) at LSB positions of the candidate attributes for some algorithmically chosen tuples. During embedding phase, the tuples are securely partitioned into groups using the cryptographic hash function and only the first m (which is equal to the length of the owner's watermark) groups are considered. The decision whether to mark i^{th} $(1 \leq i \leq m)$ group depends on the i^{th} bit of the owner's watermark, whereas the selection of the tuples in a group is based on a secret key (which is different from that used during partitioning) as well as the information at second LSB positions of the numeric candidate attributes. Finally, for the selected tuples random numbers (between 0 and 9) are embedded at LSB positions in the attribute values of those tuples. Observe that although the owner has a watermark of length m, it is not actually embedded. Rather, it is used to identify some valid groups to embed the random values which acts as embedded watermark information. The detection phase determines the presence of mark in a group if the maximum occurrence frequency for a value between 0 and 9 for that group exceeds a threshold.

(b) Image as watermark information.

Wang et al. [Wang et al., 2008a] describe an image-based watermarking scheme where instead of embedding original image as watermark, an scrambled image based on Arnold transform with scrambling number d is used. Since Arnold transform of an image has the periodicity P, the result which is obtained in the extraction phase can be recovered from the scrambled form to the original after $(P - d)$ iterations. In the embedding phase, the original image of size $N \times N$ is first converted into scrambled image which is then represented by a binary string b_s of length $L = N \times N$. Secondly, all tuples in the relation are grouped into L groups. The hash value which is computed using tuple's primary key, secret key and order of the image, determines the group in which each tuple belongs. Finally, the i^{th} bit of b_s is embedded into the algorithmically chosen bit position of the attribute value for those tuples in i^{th} group that satisfy a particular criteria. The detection phase follows majority voting technique. However, the security of this scheme improves as it relies not only on the secret key but also the scrambling number d and the order of the image N.

Rather than embedding scrambled image, the watermarking technique in [Hu et al., 2009] embeds the original image by first converting it into a bit flow (EMC, Encrypted Mark Code) of certain length, and then by following similar algorithmic steps as in [Wang et al., 2008a]. The only two differences are that (i) the watermark insertion technique in [Wang et al., 2008a] assumes single fixed attribute to mark for all tuples whereas [Hu et al., 2009] does not, and (ii) during selection of bit positions, the order of the image is not considered in [Hu et al., 2009]. Finally, after marking, [Hu et al., 2009] checks the usability of the data with respect to the intended use. If acceptable, the change is committed, otherwise rolled back.

Another watermarking scheme to embed image in BMP format is presented in [Zhou et al., 2007]. In watermark insertion phase, the BMP image is divided into two parts: *header* and *image data*. An error correction approach of BCH (Bose-Chaudhuri-Hocquenhem) coding is used to encode the image data part into watermark. Based on the tuple's ID value which is computed by performing hashing function parameterized with tuple's primary key and BMP header, all the tuples are assigned to k distinct subsets, where k is the length of the watermark. Finally, each of the k bits of the watermark is used to mark each of the k subsets of tuples. During the marking, the selected least significant bit positions of selected attributes of some specific tuples satisfying a particular criteria are altered. The selection of bit positions depends on HASH or MAC function parameterized with BMP header, tuple's primary key and other parameters like number of least significant bits in the attribute value. Observe that the selected bit positions are not set to the watermark bits directly but rather, are set to mask bits which are computed from both the hash value and the watermark bit together.

The image-based fragile watermarking scheme in [Tsai et al., 2007] aims at maintaining integrity of the database and uses support vector regression (SVR) to train high correlation attributes to generate the SVR predicting function for embedding watermark into particular numeric attributes. This scheme consists of three phases: (i) Training Phase: Select training tuples and obtain trained SVR predicting function; (ii) Embedding Phase: All tuples in the relation are used to embed image watermark where the number of watermark bits is designed to be equal to the number of tuples. Each numeric attribute value C_i of i^{th} tuple t_i is predicted using the SVR prediction function f resulting $\bar{C}_i = f(t_i)$. Based on the i^{th} watermark bit b_i (obtained after converting the image into bit flow), the value of C_i is modified by $\bar{C}_i + 1$ or $\bar{C}_i - 1$; (iii) Tamper Detection: The trained SVR predicting function is used to generate the predicted value for each tuple and compared with the value contained in the database. The difference of these two values determines the watermark information and can ensure whether database is tampered or not. However, the limitation of this scheme is that it can identify the modification which takes place in the objective attribute set only. This scheme works good in the case where the tuples in the table are independent but highly correlated between the attributes.

(c) Speech as watermark information.

Wang et al. [Wang et al., 2008b] propose the use the owner's speech to generate unique watermark. The preparation of watermark from the speech consists of several stages: compression of speech signal to shorten the watermark, speech signal enhancement to remove noise in frequency domain, speech signal conversion into bit stream, and finally, watermark generation by using the message of the copyright of the holder and the result of the converted speech signal. The bit-level marking is performed during watermark embedding phase by following the same algorithmic steps as in image-based technique of [Hu et al., 2009].

(d) Genetic Algorithm based watermark signal.

The authors in [Meng et al., 2008] propose Genetic Algorithm-based technique to generate watermark signal, focusing on the optimization issue. They follow the same algorithmic framework as of [Hu et al., 2009].

(e) Content characteristics as watermark information.

The watermarking schemes in [Zhang et al., 2006], [Guo et al., 2006b] are performed based on the content of the database itself.

In [Zhang et al., 2006], the watermark insertion phase extracts some bits, called local characteristic, from the characteristic attribute A_1 of tuple t and embeds those bits into the watermark attribute A_2 of the same tuple. The selection of tuples depends on whether the generated random value (between 0 and 1) is less than the embedded proportion α of the relational databases and the non-

NULL requirement of characteristic attribute value. In the watermark detection phase, by following similar procedure, the local characteristic of the characteristic attribute are extracted and compared against the last bits of watermark attribute.

In [Guo et al., 2006b], the authors propose a fragile watermarking scheme that can verify the integrity of database relation. In the proposed scheme, all tuples in a database relation are first securely divided into groups and sorted. In each group, there are two kinds of watermarks to be embedded: attribute watermark W_1 which consists of γ watermarks of length ν and tuple watermark W_2 which consists of ν watermarks of length γ, where γ and ν are the number of attributes in a tuple and average number of tuples in each group respectively. W_1 and W_2 are created by extracting bit sequence from the hash value. For attribute watermark W_1, the hash value is generated according to the message authentication code and the same attribute of all tuples in the same group, while for tuple watermark W_2, it is formed from the same message authentication code and all attributes of the same tuple. Observe that, in both the embedding and detection phases, they ignore the least two significant bits of all attributes of numeric type except the primary key when computing hash values. The attribute watermark is embedded in LSB level, whereas tuple watermark is embedded at next to the LSB level. In this way, the embedded watermarks actually form a watermark grid, which helps to detect, localize and characterize modifications.

(f) Cloud model as watermark information.

The cloud watermarking scheme in [Zhang et al., 2005] is based on the Cloud Model with three digital characteristics: Expected value (Ex), Entropy (En) and Hyper Entropy (He). In watermark creation phase, it uses the forward cloud generation algorithm to generate cloud drops from cloud and embed those cloud drops into the relation as watermark, whereas the detection algorithm uses backward cloud generation algorithm to extract the cloud with parameters Ex, En and He from the embedded cloud drops, and finally, a similar cloud algorithm is used to verify whether both clouds (one used during watermark embedding and other extracted during watermark verification phase) are similar or not. This scheme is not blind as it requires original relational database during verification phase.

(g) Other meaningful watermark information.

The watermarking scheme in [Huang et al., 2004] embeds a meaningful watermark information by first converting it into a bit flow. The scheme computes unique ID for all tuples in the relation and sort them in ascending order according to their ID values. The tuples are then partitioned into p groups each containing m tuples. The i^{th} bit of the bit flow is embedded into the selected tuples in i^{th} group by following same selection criteria as in AHK algorithm with exception that it considers only single attribute to mark. Before committing the change, a constraint function is used to check whether the change exceeds the data usability bounds. The constraint function includes the basic data statistical measurement constraints, semantics constraints and structural constraints. Mean and standard deviation of the data set are very common aspects in basic data statistical measurement constraints. Semantics constraints and structural constraints are defined by user's input as SQL statements according to relational table. If during embedding phase, any tuple is selected but rolled back, it is recorded and avoided during extraction phase to reduce false positive. Watermark extraction phase is blind, probabilistic and follows the majority voting technique.

The partitioning of tuples in most of the techniques is based on hashing. Huang et al. [Huang et al., 2009], instead, propose the use of well-known techniques (*e.g.* *k-means* algorithm) to cluster the tuples into some equivalent classes. The embedding of the watermark bit is based on the comparison of the parity of watermark bit and the LSB of candidate attribute. The *k-means* method assures the location of the embedded watermark irregular.

The watermark insertion phase in [Hu et al., 2005] works in three phases: (*i*) select a group of candidates in all attributes of the relation, and record it as the watermark schema; (*ii*) append the error correction code (ECC Code) to the watermark; (*iii*) executes the watermark insertion algorithm. The insertion algorithm creates pseudo random sequence using primary key and secret key. This sequence is used to identify the attribute to mark based on the significance of the attributes and the watermark bit to be embedded. During marking, the local constraint and a bidirectional mapping (to reduce watermarking data of various types into numeric data) are used. The local constraints can be defined as the upper bounds of "the distance" of attributes after/before watermarking. Finally global constraints which is a series of SQL statements are evaluated to decide whether to commit the changes. Observe that watermark schema selection in embedding phase and watermark schema detection in verification phase exploit the non-blindness property.

[Cui et al., 2006], [Guo et al., 2005], [Xinchun et al., 2007] adopts watermarking scheme which follows same algorithmic steps as of [Hu et al., 2009] but embeds other meaningful watermark information rather than image by first converting it into a bit flow of certain length. In [Xinchun et al., 2007], the selection of the candidate attribute is based on the weights of all numeric attributes with a different hashing function. Observe that in watermark insertion phase of [Guo et al., 2005] the mark position is determined using the mark bit itself.

6.1.2 Watermarking Based on Categorical Data Type Attribute

Unlike the aforementioned watermarking schemes where the marking is based on numeric attribute, the right protection scheme in [Sion et al., 2005],[Sion, 2004] proposed by Sion et al. is based on categorical type data. The watermark embedding process starts with a relation with at least a categorical type attribute A (to be watermarked), a watermark wm and a set of secret keys (k_1, k_2), and other parameters (e.g., e which determines the percentage of tuples to mark). Using the primary key K and secret key k_1 and parameter e, it discovers a set of "fit" tuples, used to encode the mark. The fit tuple selection process is same as AHK algorithm. Suppose the database relation has η tuples, then fit tuples set contains roughly η/e tuples. The shorter watermark wm is converted into wm_data of length equal to η/e by deploying Error Correcting Code (ECC). The marking algorithm generates a secret value of required number of bits to represents all possible categorical values for attribute A depending on the primary key and k_1, and then, forcing its least significant bit to a value according to a corresponding (random, depending on the primary key and k_2) position in wm_data data. The pseudorandom nature of hash function $H(T_i(K), k_2))$ guarantees, on average, that a large majority of the bits in wm_data data are going to be embedded at least once. The use of different key k_1 and k_2 ensures that there is no correlation between the selected tuples for embedding (selected by k_1) and the corresponding bit value positions in wm_data (selected by k_2). They also suggest to perform embedding based on multiple categorical attributes by considering not only the association between the primary key and single categorical attribute A but all association between primary key and categorical attributes to increase robustness of the scheme. Although this scheme is claimed to be robust against serious attacks (*e.g.* random attacks), however, the scheme is not suitable for database relations that need frequent updates, since it is very expensive to re-watermark the updated database relations. Though only a small part of selected tuples are affected by watermark embedding, the modifications of categorical attributes (*e.g.* change from "red" to "blue") in certain applications may be too significant to be acceptable. This watermarking technique is applied to binned medical data in a hierarchical manner [Bertino et al., 2005].

6.1.3 Watermarking Based on Non-Numeric Multi-Word Attributes

Ali and Ashraf [Al-Haj and Odeh, 2008] propose a watermarking scheme which is based on hiding binary image in spaces of non-numeric multi-word attributes of subsets of tuples, instead of numeric attribute at bit-level. The watermark is divided into m string each containing n bits. On the other hand, the database is also divided into non-intersecting subsets each containing m tuples. The m short strings of the watermark image are embedded into each m-tuple subset. The embedding is done as follows: suppose the integer representation of the i^{th}, $i \in [1 \dots m]$, short string is d_i. A double space is created after d_i words of the pre-selected nonnumeric, multi-word attribute of i^{th} tuple in the subset. The extraction phase counts the number single spaces appearing before double space which indicates the decimal equivalent of the embedded short binary string. Since the proposed algorithm embeds the same watermark for all non-intersecting subsets of the database, it is robust against subset deletion, subset addition, subset alteration and subset selection attacks. Another advantage for space-based watermarking is that large bit-capacity available for hiding the watermark which may also facilitate embedding of multiple small watermarks. However, it may suffer from watermark removal attack if Mallory replaces all double spaces between two words (if exist) by single space for all tuples in the relation.

6.1.4 Watermarking Based on Tuple or Attribute Insertion

(a) Fake tuples as watermark information.

The approach in [Pournaghshband, 2008] aims to generate fake tuples and insert them erroneously into the database. The fake tuple creation algorithm take care of candidate key attributes and sensitivity level of non candidate attributes. He uses Bernoulli sampling probability p_i for the i^{th} non-candidate attribute A_i to decide its fake value which may be chosen uniformly or as the value with higher occurrence frequency in the existing set of values of A_i in the relation. Unlike other algorithms, the detection algorithm is not an inverse algorithm to the watermark generating algorithm and insertion algorithm is probabilistic in nature. Detection algorithm checks to see whether the fake tuples inserted during watermark insertion phase, exist or has been changed. It checks it via primary key. As soon as it finds one match (i.e. identical or similar tuples), detection is done. The detection will fail for the watermarked database when all of the fake tuples are deleted by benign deletions. The number of fake tuples to be inserted is decided by the database owner. However, the watermark insertion phase must take into account the fact that the values of the fake tuples marks should not by any means degrade the quality of the data in the database and should not impact the query results. One advantage of this scheme is that the ownership can be publicly verified more than once until all the fake tuples are revealed and the scheme does not suffer from incremental updatability.

(b) Virtual attribute as watermark information.

Rather than inserting fake tuples, the author in [Prasannakumari, 2009] proposes another watermarking technique by inserting a virtual attribute in the relation which will serve as watermark containing parity checksum of all other attributes and an aggregate value obtained from any one of the numeric attribute of all tuples. The process of virtual attribute insertion is performed independently for each non-overlapping partitions obtained from the original relation. This scheme is designed to authenticate the tamper-proof receipt of the database over an insecure communication channel. Although this approach is fragile and can easily detect any of the deletion or insertion or alter attacks, it suffers from the watermark removal attack.

6.2 Distortion-free Watermarking

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

6.2.1 Extracting Hash Value as Watermark Information

In order to achieve the purpose of fragile watermark, authors in [Li et al., 2004], [Bhattacharya and Cortesi, 2009a] proposed watermarking schemes which are able to detect any modifications made to a database relation. These schemes are designed for categorical data that cannot tolerate distortion, hence, the watermark embedding is distortion free. In [Li et al., 2004], partitioning of tuples is based on the hash value parameterized with primary key and secret key, whereas in [Bhattacharya and Cortesi, 2009a], partitioning is based on categorical attribute values. After partitioning, the tuple level and group level hash values for each group are computed. In [Li et al., 2004], a watermark of length equal to the number of tuple pairs in the group, is extracted from the group level hash value and for each tuple pair, the order of the two tuples are changed or unchanged according to their tuple hash values and the corresponding watermark bit. Moreover, Li [Li, 2007] suggests to perform the exchange of tuples' positions based on Myrvold and Ruskeys linear permutation unranking algorithm to increase the embedding capacity. In these schemes, any modification of an attribute value will affect the watermarks in two groups as the modified tuple may be removed from one group and be added to the other group.

6.2.2 Combining Owner's Mark and Database Features as Watermark Information

The scheme proposed by Tsai et al. [Tsai et al., 2006] aims at maintaining the integrity of the information in the database and is based on public authentication mechanism. The idea behind of this scheme is that, first an watermark W is created which is a $\sqrt{n} \times \sqrt{n}$ white image, where n is the no. of tuples in the relation, besides four corners having mark of the owner. It creates a value C_i ($0 \leq C_i \leq 255$) for each tuple t_i in the database using hash function MD5 and XOR operation. If there are n tuples in the database, it produces a feature C of length n by combining all C_i in order. Finally, a certification code R is produced by XOR-ing C and W. The encrypted form of R using private key is made available publicly. During verification the integrity of the relation T', in similar way, it generates feature C' from T'. After decryption using public key the certification code R is XOR-ed with C' that yield the watermark W'. The integrity of this extracted watermark proves the integrity of the database.

6.2.3 Converting Database Relation into Binary Form used as Watermark Information

The public watermarking scheme by Li and Deng [Li and Deng, 2006] is applicable for marking any type of data including integer numeric, real numeric, character, and Boolean, without fear of any error constraints. The interesting features of this scheme is that it does not use any secret key and can be verified publicly as many times as necessary. The unique watermark key, used in both creation and verification phase, is public and obtained by one-way hashing from various information like the Identity of the owner(s) and characteristics of the database (*e.g.* DB Name, Version etc.). Observe that the public watermark key is different from the public-private key pair of asymmetric cryptography. This watermark key is used to generate a watermark W from the relation R. The watermark W is a database relation whose schema is $W(P, W_0, ..., W_{\gamma-1})$, where $W_0, \ldots, W_{\gamma-1} \in \{0, 1\}$. Compared to database relation R, the watermark W has the same number η of tuples and the same primary key attribute P . The number γ of binary attributes in W is a control parameter that determines the number ω of bits in W , where $\omega = \eta \times \gamma$ and $\gamma < number\ of\ attributes$ in R. In the algorithm, a cryptographic pseudorandom sequence generator (*e.g.*, Linear Feedback Shift Register [Halder et al., 2009]) to randomize the order of the attributes and the MSBs of the attribute values are used for generating the watermark W. The use of MSBs is for thwarting potential attacks that modify the data. Since the watermark key K, the watermark W, and the algorithm are publicly known, anyone can locate those MSBs. Any modification to these MSBs introduces intolerable errors to the underlying data and can easily be captured during verification phase. However, alteration of other bits in the data can not be detected by this scheme.

The watermarking techniques in [Bhattacharya and Cortesi, 2009b] follows the same algorithmic steps as of [Li and Deng, 2006]. The basic difference is that the former considers a private key instead of public, and thus, can not be publicly verifiable. In addition, the former is partition based and considers the extracted binary watermark as an image which is used to prove the ownership. This image is treated as the abstract counterpart of the concrete relation R, and the abstraction is sound in the sense that concretization of the abstract image must cover R. However, the disadvantage of this scheme is that the extracted image may not have any meaningful pattern.

In [Halder and Cortesi, 2010b], [Halder and Cortesi, 2010a], the authors address the issue of persistency of watermarks, that serves as a way to recognize the integrity and ownership proof of the database while allowing the evaluation of the database by queries in a set of queries Q. The persistency of the watermark is preserved by exploiting the invariants (*Static Part* and *Semantics-based Properties* of the underlying data in the database *w.r.t.* Q) of the database states. The watermarking algorithms are designed as an improvement of the proposal by Li and Deng [Li and Deng, 2006] in terms of fragileness and persistency.

6.2.4 R-tree Based Permutation as Watermark

In contrast to the traditional watermarking schemes, an R-tree data structure-based watermarking technique has been proposed in [Kamel, 2009]. The proposed technique takes advantage of the fact that R-trees do not put conditions on the order of entries inside the node. In the proposed scheme, entries inside R-tree nodes are rearranged, relative to a secret initial order (a secret key), in a way that corresponds to the value of the watermark. To achieve that, they propose a one-to-one mapping between all possible permutations of entries in the R-tree node and all possible values of the watermark. Without loss of generality, watermarks are assumed to be numeric values. The proposed mapping employs a numbering system that uses variable base with factorial value. The detection rate of the malicious attacks depends on the nature of the attack, distribution of the data, and the size of the R-tree node. The proposed watermarking technique has the following desirable features: (*i*) It does not change the values of the data in the R-tree node but rather hides the watermark in the relative order of entries inside the R-tree node; (*ii*) It does not increase the size of the R-tree; (*iii*) The proposed technique does not interfere with R-tree operations; (*iv*) The performance overhead is minimal; (*v*) The integrity check does not require the knowledge of unwatermarked data (blind watermark).

7 Public Vs. Private Watermarking Techniques

Most of the existing watermarking techniques [Guo et al., 2006b], [Li et al., 2004], [Agrawal et al., 2003b], [Zhou et al., 2007], [Sion et al., 2005] etc in the literature are private, meaning that they are based on some private parameters (e.g. a secret key). Only the authorized people (e.g. database owners) who know these private parameters are able to verify the watermark and prove their ownership of the database in case of any illegal redistribution, false ownership claim, theft etc. However, private watermarking techniques suffer from disclosure of the private parameters to dishonest people once the watermark is verified in presence of the public. With access to the private parameters, attackers can easily invalidate watermark detection either by removing watermarks from the protected data or by adding a false watermark to the non-watermarked data. In contrast, in case of public watermarking techniques [Li and Deng, 2006], [Halder and Cortesi, 2010a], [Tsai et al., 2006], any end-user can verify the embedded watermark as many times as necessary without having any prior knowledge about any of the private parameters to ensure that they are using correct (not tampered) data coming from the original source. For instance, when a customer uses sensitive information such as currency exchange rates or stock prices, it is very important for him to ensure that the data are correct and coming from the original source.

8 Fingerprinting Techniques

The fingerprinting techniques proposed in the literature are based on numerical data type attributes.

Li et al. [Li et al., 2003], [Li et al., 2005] propose an extension of agrawal and Kiernan's scheme [Agrawal and Kiernan, 2002] to embed the fingerprint into the database. The basic difference is that instead of embedding meaningless bit pattern, they embed meaningful fingerprint where the fingerprint of length L (where $L > logN$, N=number of buyers) is computed from cryptographic hash function whose input is the concatenation of a secret key K (known by the merchant only) and user identifier n. The index of the fingerprint bit to be embedded is computed using hash function which is different from the hash used to select the bit positions, to ensures that the fingerprint bits are not correlated with the locations in which they are embedded. However, since the method is applicable for the relations with primary key only, [Li et al., 2003] mentions three different approaches to perform fingerprinting for a relation which has no primary key: (i) S-Scheme: The bits other than the least significant bits available for marking in the single numeric attribute of each selected tuples are considered as virtual primary key. But it suffers from both duplicate and deletion problem; (ii) E-Scheme: All numeric attributes of each selected tuples are examined independently by computing a virtual primary key from the attribute values. However, E-Scheme suffers from duplicate problem; (iii) M-Scheme: Dynamically selects the bit positions used to construct a virtual primary key.

Liu et al. [Liu et al., 2004] propose a block oriented finger printing scheme. For each buyer, the fingerprint is obtained using hash function based on the private key and buyer's ID. By combining the least significant bits of the table attributes, a two dimensional image is obtained and is divided into sub-images. Using Pseudo-random generator a bit is chosen in each sub-image and is XOR-ed with a fingerprint bit.

In [Guo et al., 2006a], the authors propose two level fingerprinting scheme to identify both the owner and the traitor. In the first embedding process, they embed a unique fingerprint to identify each recipient to whom the relational data is distributed. In this embedding technique, firstly tuples are partitioned into m group where m is the number of bits in the binary representation of the fingerprint. Next i^{th} fingerprint bit is embedded to the candidate bit position for selected tuples in the i^{th} group. The second embedding process is designed for verifying the extracted fingerprint and giving a numerical confidence level. They use the fingerprint itself as a secret key. The selected positions will be set to "1" or "0" depending on whether the hash value (seeded with secret key concatenating with Primary Key) is odd or even. To avoid conflict between the two embedding, only tuples not selected in the first embedding process are allowed to be marked in the second. The fingerprint extracting algorithm is the converse process to this first embedding process. In fingerprint extracting algorithm, they extract a bit of the fingerprint from each group and a numerical confidence level of each bit could be calculated. The bits those do not meet the pre-set confidence level are unreliable but could be localized. These bits could either be "1" or "0". Thus, a candidate set of suspect fingerprints can be obtained. In the fingerprint verification algorithm, they use each suspect fingerprint as the secret key to detect the pattern embedded in the second embedding process. Once the pattern is detected, the fingerprint is proved to be the exact originally embedded fingerprint at a high numerical confidence level.

The classification and comparison of different schemes are depicted in Table 1 and 2. Table 1 depicts distortion-based watermarking and fingerprinting schemes, whereas distortion-free watermarking schemes are listed in Table 2. As the watermark detection phase for most of the schemes is probabilistic in nature, we explicitly mention when the detection phase is deterministic.

9 Probabilistic Issues

Consider n Bernoulli trials of an event, with probability p of success and $q = 1-p$ of failure in any trial. Let $b(k; n, p)$ be the probability of obtaining exactly k successes out of n Bernoulli trials. Then,

$$b(k; n, p) = \binom{n}{k} p^k q^{n-k}$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad 0 \leq k \leq n$$

Let $B(k; n, p) = \sum_{i=k+1}^{n} b(i; n, p)$ which is the probability that more than k successes take place in n Bernoulli trials. Consider the robust watermarking scheme AHK algorithm [Agrawal and Kiernan, 2002],[Agrawal et al., 2003b]. In AHK algorithm, a watermark is successfully detected if number of match is more

Proposed Schemes	Watermark Information	Cover Type	Granularity Level	Verifiability	Intent
AHK algorithms [Agrawal and Kiernan, 2002]; [Agrawal et al., 2003a]; [Agrawal et al., 2003b]	Meaningless Bit Pattern	Numeric	Bit-level	Blind, Private	Ownership Proof
Gupta et al. [Gupta and Pieprzyk, 2009]	Meaningless Bit Pattern	Numeric	Multi-Bit level	Blind, Reversible, Private	Ownership Proof
Image-based [Wang et al., 2008a]; [Zhou et al., 2007]; [Al-Haj and Odeh, 2008]; [Tsai et al., 2007]	Image	Numeric or Non-Numeric Multi-Word	Bit-level or Whole Attribute Value or Character-level	Blind, Private	Ownership Proof and/or Tamper Detection
Speech-based [Wang et al., 2008b]	Owner's Speech	Numeric	Bit-level	Blind, Private	Ownership Proof
Content-based [Zhang et al., 2006]; [Guo et al., 2006b]	Database Content	Numeric	Multi-Bit level	Blind, Private	Ownership Proof and/or Tamper Detection and Localization
Cloud Model-based [Zhang et al., 2005]	Cloud model with three characteristics: Expected value, Entropy and Hyper Entropy	Numeric	Whole Attribute Value	Non-Blind, Private	Ownership Proof
Categorical Attribute-based [Sion et al., 2005]; [Sion, 2004]	Meaningful Binary String	Categorical	Bit-level	Blind, Private	Ownership Proof
Fake Tuple-based [Pournaghshband, 2008]	Fake Information obtained from database content	Database Table	Tuple-level	Blind, Private	Ownership Proof
Virtual Attribute-based [Prasannakumari, 2009]	Database content	Database Table	Attribute-level	Blind, Deterministic, Private	Tamper Detection
Others [Guo et al., 2005]; [Huang et al., 2004]; [Hu et al., 2005]	Meaningful Information of any type	Numeric	Bit-level	Blind or Non-Blind, Private	Ownership Proof
Fingerprinting Techniques [Li et al., 2005]; [Li et al., 2003]; [Liu et al., 2004]; [Guo et al., 2006a]	Meaningful Fingerprint Identifying Buyers uniquely	Numeric	Bit-level	Blind, Private	Traitor Detection

Table 1: Comparison of Distortion-based Watermarking and Fingerprinting Schemes

Proposed Schemes	Watermark Information	Cover Type	Granularity Level	Verifiability	Intent
Permutation-based [Li et al., 2004]; [Bhattacharya and Cortesi, 2009a]	A Part of Group-level Hash Value	Tuples' Positions	Tuple-level	Blind, Private	Tamper Detection
Characteristic-based [Tsai et al., 2006]	White Image with Owner's Mark at Four Corners	Nil	Nil	Blind, Public	Tamper Detection
Binary Form Relation [Li and Deng, 2006]; [Halder and Cortesi, 2010a]; [Halder and Cortesi, 2010b]; [Bhattacharya and Cortesi, 2010]; [Bhattacharya and Cortesi, 2009b]	Relation in Binary form	Nil	Nil	Blind, Public or Private	Ownership Proof and/or Tamper Detection
R tree-based Scheme [Kamel, 2009]	Numeric Value Identifying Owner	Order of Entries in R-tree Nodes	R-tree Nodes	Blind, Private	Tamper Detection

Table 2: Comparison of Distortion-free Watermarking Schemes

than the threshold τ which is just a percentage of the total number of embedded bits. Suppose ω is the total number of embedded bits. If the detection algorithm scans ω number of bits and observes the number of bits whose values match those assigned by the marking algorithm, the probability that at least τ out of ω random bits matches the assigned value is $B(\tau;\omega,0.5)$. Thus, Alice should choose τ such that $B(\tau;\omega,0.5) < \alpha$ where α is the false hit *i.e.* probability that Alice will discover her watermark in a database relation not marked by her. By choosing lower values of α, Alice can increase her confidence that if the detection algorithm finds her watermark in a suspected relation, it probably is a pirated copy. Suppose, Mallory knows the private parameters ν and ξ used by AHK algorithm. Since Mallory does not know the exact marked positions, he randomly chooses ζ tuples out of η tuples and flips all of the bits in all of ξ bit positions in all of ν attributes. The attack would be successful, if he flips at least $\bar{\tau} = \omega - \tau + 1$ marks. The probability that this attack will succeed is represented as $\sum_{\bar{\tau}}^{\omega} \frac{\binom{\omega}{i}\binom{\eta-\omega}{\zeta-i}}{\binom{\eta}{\zeta}}$.

Consider now the categorical attribute based schemes in [Sion et al., 2005]. Suppose an attacker randomly alters q number of tuples and succeeds in each case to flip the embedded watermark bit with a success rate p, then the probability of success of altering at least r $(r < q)$ watermark bits in the result is:

$$P(r,q) = \sum_{i=r}^{q} \binom{q}{i} \times p^q \times (1-p)^{(q-i)}$$

This metric illustrates the relationship between attack vulnerability and embedding bandwidth. Since only e tuple (on average) is watermarked, thus, Mallory effectively attacks only an average of a q/e tuples actually watermarked. If $r > q/e$, then $P(r,q) = 0$. For $r < q/e$, we have

$$P(r,q) = \sum_{i=r}^{q/e} \binom{q/e}{i} \times p^{q/e} \times (1-p)^{(q/e-i)}$$

Now consider a fragile watermarking scheme [Li et al., 2004]. Assume that each group consists of exactly ν tuples; thus, the length of each embedded watermark W is $\nu/2$. Let the j^{th} attribute of the i^{th} tuple is modified. This modification will affect the tuple hash value, group hash value and thus yield to watermark W'. The probability that this modification can be detected *i.e.* $W \neq W'$ is, thus, $P = 1 - \frac{1}{2^{\nu/2}}$. If the value of the primary key has been modified. The probability that the tuple will be in the same partition is $1/g$ and probability of shifting to another group is $1 - 1/g$. Shifting of a tuple between groups affect the hash value of both groups. The probability that the modification can be correctly detected is $P = \frac{1}{g}(1 - \frac{1}{2^{\nu/2}}) + \frac{g-1}{g}(1 - \frac{1}{2^{(\nu-1)/2}})(1 - \frac{1}{2^{(\nu+1)/2}})$. Clearly, the probability in this case is less than that in the case of modifying non-primary key value.

10 Conclusions

In this paper we survey the current state-of-the-art of different watermarking and fingerprinting techniques for relational databases. We classify all the techniques based on (i) whether the technique introduces the distortion to underlying data, (ii) the type of the cover where mark is embedded, and (iii) the type of the watermark information. Most of the distortion-based watermarking techniques mainly aim at protecting the ownership, whereas distortion-free watermarking techniques mostly are fragile and aim at maintaining integrity of the database information. Although we classify the schemes based on different watermark information, most of the numerical distortion-based schemes follow almost similar steps to identify the candidate bit positions for the watermark. Finally, we observe that the usability of the watermarked database and queries still remains an open issue for future research.

Acknowledgement

Work partially supported by Italian MIUR COFIN'07 project "SOFT", and by RAS L.R. 7/2007 Project TESLA.

References

[Abdel-Hamid et al., 2004] Abdel-Hamid, A. T., Tahar, S., and Aboulhamid, E. M. (2004). A survey on ip watermarking techniques. *Design Automation for Embedded Systems*, 9(3):211–227.

[Agrawal et al., 2003a] Agrawal, R., Haas, P. J., and Kiernan, J. (2003a). A system for watermarking relational databases. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data (SIGMOD '03)*, pages 674–674, San Diego, California. ACM Press.

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

[Agrawal and Kiernan, 2002] Agrawal, R. and Kiernan, J. (2002). Watermarking relational databases. In *Proceedings of the 28th international conference on Very Large Data Bases (VLDB '02)*, pages 155–166, Hong Kong, China. VLDB Endowment.

[Agrawal and Srikant, 2000] Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. *ACM SIGMOD Record*, 29(2):439–450.

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

[Bertino et al., 2005] Bertino, E., Ooi, B. C., Yang, Y., and Deng, R. H. (2005). Privacy and ownership preserving of outsourced medical data. In *Proceedings of the 21st International Conference on Data Engineering (ICDE '05)*, pages 521–532, Tokyo, Japan. IEEE Computer Society.

[Bhattacharya and Cortesi, 2009a] Bhattacharya, S. and Cortesi, A. (2009a). A distortion free watermark framework for relational databases. In *Proceedings of the 4th International Conference on Software and Data Technologies (ICSOFT '09)*, pages 229–234, Sofia, Bulgaria. INSTICC Press.

[Bhattacharya and Cortesi, 2009b] Bhattacharya, S. and Cortesi, A. (2009b). A generic distortion free watermarking technique for relational databases. In *Proceedings of the 5th International Conference on Information Systems Security (ICISS '09)*, pages 252–264, Kolkata, India. Springer LNCS, Volume 5905.

[Bhattacharya and Cortesi, 2010] Bhattacharya, S. and Cortesi, A. (2010). Database authentication by distortion-free watermarking. In *Proceedings of the 5th International Conference on Software and Data Technologies (ICSOFT '10)*, pages 219–226, Athens, Greece. INSTICC Press.

[Cui et al., 2006] Cui, X., Qin, X., Sheng, G., and Zheng, J. (2006). A robust algorithm for watermark numeric relational databases. In *Proceedings of the 2010 International conference on Intelligent computing (ICIC '06)*, pages 810–815, Kunming, China. Springer Lecture Notes in Control and Information Sciences.

[Guo et al., 2006a] Guo, F., Wang, J., and Li, D. (2006a). Fingerprinting relational databases. In *Proceedings of the 2006 ACM symposium on Applied computing (SAC '06)*, pages 487–492, Dijon, France. ACM Press.

[Guo et al., 2005] Guo, F., Wang, J., Zhang, Z., Ye, X., and Li, D. (2005). An improved algorithm to watermark numeric relational data. In *Proceedings of the 6th International Workshop on Information Security applications (WISA '05)*, pages 138–149, Jeju Island, Korea. Springer LNCS, Volume 3786.

[Guo et al., 2006b] Guo, H., Li, Y., Liua, A., and Jajodia, S. (2006b). A fragile watermarking scheme for detecting malicious modifications of database relations. *Information Sciences*, 176:1350–1378.

[Gupta and Pieprzyk, 2009] Gupta, G. and Pieprzyk, J. (2009). Database relation watermarking resilient against secondary watermarking attacks. In *Proceedings of the 5th International Conference on Information Systems Security (ICISS '09)*, pages 222–236, Kolkata, India. Springer LNCS, Volume 5905.

[Hacigumus et al., 2002] Hacigumus, H., Iyer, B., and Mehrotra, S. (2002). Providing database as a service. In *Proceedings of the 18th International Conference on Data Engineering (ICDE '02)*, pages 29–38, San Jose, California, USA. IEEE Computer Society.

[Halder and Cortesi, 2010a] Halder, R. and Cortesi, A. (2010a). A persistent public watermarking of relational databases. In *Proceedings of the 6th International Conference on Information Systems Security (ICISS '10)*, pages 216–230, Gandhinagar, Gujarat, India. Springer LNCS, Volume 6503.

[Halder and Cortesi, 2010b] Halder, R. and Cortesi, A. (2010b). Persistent watermarking of relational databases. In *Proceedings of the IEEE International Conference on Advances in Communication, Network, and Computing (CNC '10)*, pages 46–52, Calicut, Kerala, India. IEEE Computer Society.

[Halder et al., 2009] Halder, R., Dasgupta, P., Naskar, S., and Sarma, S. S. (2009). An internet-based ip protection scheme for circuit designs using linear feedback shift register (lfsr)-based locking. In *Proceedings of the 22nd Annual Symposium on Integrated Circuits and System Design(SBCCI '09)*, Natal, Brazil. ACM Press.

[Hu and Grefen, 2002] Hu, J. and Grefen, P. (2002). Component based system framework for dynamic b2b interaction. In *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC '02)*, pages 557–562, Oxford, England. IEEE Computer Society.

[Hu et al., 2005] Hu, T., Chen, G., Chen, K., and Dong, J. (2005). Garwm: Towards a generalized and adaptive watermark scheme for relational data. In *Proceedings of the 6th International Conference in Advances in Web-Age Information Management (WAIM '05)*, pages 380–391, Hangzhou, China. Springer LNCS, Volume 3739.

[Hu et al., 2009] Hu, Z., Cao, Z., and Sun, J. (2009). An image based algorithm for watermarking relational databases. In *Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09)*, pages 425–428, Zhangjiajie, Hunan, China. IEEE Computer Society.

[Huang et al., 2009] Huang, K., Yue, M., Chen, P., He, Y., and Chen, X. (2009). A cluster-based watermarking technique for relational database. In *Proceedings of the 1st International Workshop on Database Technology and Applications (DBTA '09)*, pages 107–110, Wuhan, China. IEEE Press.

[Huang et al., 2004] Huang, M., Cao, J., Peng, Z., and Fang, Y. (2004). A new watermark mechanism for relational data. In *Proceedings of the 4th International Conference on Computer and Information Technology (CIT '04)*, pages 946–950, Wuhan, China. IEEE Computer Society.

[Kamel, 2009] Kamel, I. (2009). A schema for protecting the integrity of databases. *Computers & Security*, 28:698–709.

[Kerckhoffs, 1983] Kerckhoffs, A. (1983). La cryptographie militaire. *Journal des Sciences Militaires*, 9:5–38.

[Khanna and Zane, 2000] Khanna, S. and Zane, F. (2000). Watermarking maps: hiding information in structured data. In *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms (SODA '00)*, pages 596–605, San Francisco, California, United States. Society for Industrial and Applied Mathematics.

[Lafaye, 2007] Lafaye, J. (2007). An analysis of database watermarking security. In *Proceedings of the 3rd International Symposium on Information Assurance and Security (IAS '07)*, pages 462–467, Manchester, United Kingdom. IEEE Computer Society.

[Lee and Jung, 2001] Lee, S. and Jung, S. (2001). A survey of watermarking techniques applied to multimedia. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '01)*, pages 272–277, Pusan, South Korea. IEEE Press.

[Li, 2007] Li, Y. (2007). *Database Watermarking: A Systematic View.* Springer Verlag.

[Li and Deng, 2006] Li, Y. and Deng, R. H. (2006). Publicly verifiable ownership protection for relational databases. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security (ASIACCS '06)*, pages 78–89, Taipei, Taiwan. ACM Press.

[Li et al., 2004] Li, Y., Guo, H., and Jajodia, S. (2004). Tamper detection and localization for categorical data using fragile watermarks. In *Proceedings of the 4th ACM workshop on Digital rights management (DRM '04)*, pages 73–82, Washington, DC, USA. ACM Press.

[Li et al., 2003] Li, Y., Swarup, V., and Jajodia, S. (2003). Constructing a virtual primary key for fingerprinting relational data. In *Proceedings of the 3rd ACM workshop on Digital rights management (DRM '03)*, pages 133–141, Washington, DC, USA. ACM Press.

[Li et al., 2005] Li, Y., Swarup, V., and Jajodia, S. (2005). Fingerprinting relational databases: schemes and specialties. *IEEE Transactions on Dependable and Secure Computing*, 2:34–45.

[Liu et al., 2004] Liu, S., Wang, S., Deng, R. H., and Shao, W. (2004). A block oriented fingerprinting scheme in relational database. In *Proceedings of the 7th International Conference in Information Security and Cryptology (ICISC '04)*, pages 455–466, Springer LNCS, Volume 3506. Seoul, Korea.

[Meng et al., 2008] Meng, M., Cui, X., and Cui, H. (2008). The approach for optimization in watermark signal of relational databases by using genetic algorithms. In *Proceedings of the 2008 International Conference on Computer Science and Information Technology (ICCSIT '08)*, pages 448–452, Singapore. IEEE Computer Society.

[Potdar et al., 2005] Potdar, V. M., Han, S., and Chang, E. (2005). A survey of digital image watermarking techniques. In *Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN '05)*, pages 709–716, Peth, Australia. IEEE Press.

[Pournaghshband, 2008] Pournaghshband, V. (2008). A new watermarking approach for relational data. In *Proceedings of the 46th Annual Southeast Regional Conference on XX (ACM-SE '08)*, pages 127–131, Auburn, Alabama. ACM Press.

[Prasannakumari, 2009] Prasannakumari, V. (2009). A robust tamperproof watermarking for data integrity in relational databases. *Research Journal of Information Technology*, 1:115–121.

[Qin et al., 2006] Qin, Z., Ying, Y., Jia-jin, L., and Yi-shu, L. (2006). Watermark based copyright protection of outsourced database. In *Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS'06)*, pages 301–308, Delhi, India. IEEE Computer Society.

[Sion, 2004] Sion, R. (2004). Proving ownership over categorical data. In *Proceedings of the 20th International Conference on Data Engineering (ICDE 04)*, pages 584–595, Boston, MA, USA. IEEE Computer Society.

[Sion et al., 2005] Sion, R., Atallah, M., and Prabhakar, S. (2005). Rights protection for categorical data. *IEEE Transactions on Knowledge and Data Engineering*, 17:912–926.

[Tsai et al., 2007] Tsai, M., Hsu, F., Chang, J., and Wu, H. (2007). Fragile database watermarking for malicious tamper detection using support vector regression. In *Proceedings of the 3rd International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP '07)*, pages 493–496, Splendor Kaohsiung, Taiwan. IEEE Computer Society.

[Tsai et al., 2006] Tsai, M., Tseng, H., and Lai, C. (2006). A database watermarking technique for temper detection. In *Proceedings of the 2006 Joint Conference on Information Sciences (JCIS '06)*, Kaohsiung, Taiwan. Atlantis Press.

[Wang et al., 2008a] Wang, C., Wang, J., Zhou, M., Chen, G., and Li, D. (2008a). Atbam: An arnold transform based method on watermarking relational data. In *Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering (MUE '08)*, pages 263–270, Beijing, China. IEEE Computer Society.

[Wang et al., 2008b] Wang, H., Cui, X., and Cao, Z. (2008b). A speech based algorithm for watermarking relational databases. In *Proceedings of the 2008 International Symposiums on Information Processing (ISIP '08)*, pages 603–606, Moscow, Russia. IEEE Computer Society.

[Xiao et al., 2007] Xiao, X., Sun, X., and Chen, M. (2007). Second-lsb-dependent robust watermarking for relational database. In *Proceedings of the 3rd International Symposium on Information Assurance and Security (IAS '07)*, pages 292–300, Manchester, United Kingdom. IEEE Computer Society.

[Xinchun et al., 2007] Xinchun, C., Xiaolin, Q., and Gang, S. (2007). A weighted algorithm for watermarking relational databases. *Wuhan University Journal of Natural Science*, (1):79–82.

[Zhang et al., 2005] Zhang, Y., Niu, X., and Zhao, D. (2005). A method of protecting relational databases copyright with cloud watermark. *International Journal of Information and Communication Engineering*, 1:337–341.

[Zhang et al., 2006] Zhang, Y., Niu, X., Zhao, D., Li, J., and Liu, S. (2006). Relational databases watermark technique based on content characteristic. In *Proceedings of the 1st International Conference on Innovative Computing, Information and Control (ICICIC '06)*, pages 677–680, Beijing, China. IEEE Computer Society.

[Zhou et al., 2007] Zhou, X., Huang, M., and Peng, Z. (2007). An additive-attack-proof watermarking mechanism for databases' copyrights protection using image. In *Proceedings of the 2007 ACM symposium on Applied computing (SAC '07)*, pages 254–258, Seoul, Korea. ACM Press.