Towards a Systematic Approach of Relational Database Watermarking

Monica Vladoiu *monica@unde.ro*
UPG University of Ploiesti
Ploiesti, Romania

Gabriela Moise *gmoise@upg-ploiesti.ro*
UPG University of Ploiesti
Ploiesti, Romania

Zoran Constantinescu *zoran@unde.ro*
UPG University of Ploiesti
Ploiesti, Romania

Abstract

Nowadays more and more data of socio-technical systems become available online to anyone interested to access it or process it (without data alteration or copyright infringement). Generally, these data are stored in relational databases. However, to comply with this new paradigm new models of data access and security are necessary. One upcoming trend for relational databases is to watermark the database instance, i.e. to compute a secret code, which can be either embedded directly into the database or registered to a trusted authority. Current watermarking schemes only apply to either a particular database relation or index and, generally, distort the data. In this paper, we propose a methodology for distortion-free watermarking of both the database schema and instance that takes into account the database semantics, its dynamic, and also ensuring various security levels within the database. A possible scenario on using this methodology on a real-world database is also available.

Keywords: database security; database consistency and integrity; watermarking methodology.

1. Introduction

Today more and more data of socio-technical systems becomes available to general public or to particular user groups. Users may access it or process it in any way that does not result in data alteration or copyright infringement. For example, on the European Union open data portal, data collections from several domains are available, ranging from economics, finance, trade, industry etc. to politics, social questions, employment and working conditions, education and communications, law, international organizations, and so on [9]. Data sets for intrusion detection are available on MIT Lincoln Laboratory's website [21]. Economic and financial data sets are available for analysis and research on the website of The World Bank [8]. However, collecting and managing data into databases, data warehouses, or scaling them up to big data, may raise important economical and strategic concerns for enterprises and public institutions as well. In that context, information leaks, thefts (confidentiality, traceability) or data corruption (integrity, authenticity), intentional or not, represent a real danger. This is even more critical in complex socio-technical systems that includes people, work processes, institutional and cultural factors, along with interconnected devices and services that provide for seamless integration of digital infrastructure into everyday life. The applications running in such systems often exploit physical location and other context information about users and resources to enhance the user experience. New types of interaction among users, as well as between the virtual and physical

worlds, arise from the need to share resources and collaborate. Moreover, existing policies and mechanisms may not provide adequate guarantees to deal with new exposures and vulnerabilities introduced by the emerging paradigms, in which it is difficult to separate physical security from digital security. For example, information systems involved in pervasive computing need to provide high availability (data available from anywhere, anytime, and by anybody), and high scalability and elasticity (new resources can be allocated on-demand if necessary). However, they raise many questions in terms of security and privacy, especially about data confidentiality and integrity. For instance, in case of ubiquitous health monitoring, computer assisted rehabilitation, emergency medical response systems, or recording confidential medical data, providing for protection of both organizations and end users, while fulfilling their goal is a major challenge. Therefore, new approaches of ensuring data and information security become necessary, database watermarking being one of them.

Usually, most of the data in nowadays information systems are relational and, therefore, are stored in relational databases. The most common security issues are related to ensuring data integrity, tamper detection, and copyright protection. Digital watermarking methods have become lately more appealing for data protection and security both in public or private databases that can be accessed online. Watermarking consists in insertion of some security message into a host (e.g., an image, an audio signal, or a database) by slightly modifying it based on the principle of controlled distortion. Watermarking has significant advantages when compared with other security techniques. For example, a common method to detect alteration of a database consists in using a digital signature scheme, which provides for security services such as message integrity and authentication, nonrepudiation, and protection against message tampering. However, this scheme allows to establish whether a database has been modified or not, but it does not provide for identifying or recovering the accomplished modifications nor for their location or type. Moreover, the integrity verification failure will result in uselessness of the entire database. In addition, if some modifications to the database are necessary, a new signature has to be computed, while the previous one is discarded. Finally, the generation and verification of the signatures is computationally intensive [18]. Applying encryption techniques on databases could be another solution for ensuring security of data collections. However, this is not feasible because, first, the time of encryption or decryption of large databases is prohibitively long, as these operations are computationally intensive as well. Second, the majority of users (including even application developers and database administrators) are not familiar with cryptographic techniques, and, finally, fully homomorphic encryption is still to come [22]. Also, similarly to digital signatures, if some database modifications take place, a new encryption of the data is necessary, which affects database performance. Therefore, due to the above shortcomings, a more realistic approach consists in using a watermarking technique, which generates a secret code based on the data themselves and then either embed it directly into the database or register it to a trusted authority. Watermarking has been first introduced for securing multimedia objects and extended later to relational databases. Watermarking has another important advantage when compared with the above protection mechanisms, because it provides also for *a posteriori data protection*, while those offer only a priori protection and once they are bypassed, data are no longer protected. This means that under watermarking protection, the data can be manipulated in various ways, while staying protected by means of a watermark that provides for protection of owner rights, data integrity, data traceability, etc. [6].

However, most of the existing watermarking schemes for relational databases address only *some particular tables (relations) and/or indexes* and, what is worst, *alter the data* breaking this way an important pillar of relational databases, i.e. data consistency. Also, they ignore *the data semantics* when watermarking [1, 2, 3, 4, 5, 11, 13, 15, 24, 25, 26, 27]. In our view, for increasing the efficacy of detection and localization of attacks, adequate watermarking needs to approach *the whole database, without breaking neither data consistency nor integrity that are taken for granted in relational databases*, this being a totally new approach up to our knowledge. The main contribution of this work is twofold: first, it emphasizes the need to consider the database as a whole when watermarking and, consequently, to address more database elements such as database schema, data dictionary, relationships, integrity constraints, etc., *while still ensuring the kind of data consistency and integrity expected of relational*

databases. Secondly, it introduces a general methodology for relational database watermarking (which is also applied in a working scenario). The structure of the paper is as follows: Section 2 includes the basics of watermarking relational databases, a taxonomy of possible attacks, and a set of features expected from watermarking schemes, while Section 3 includes the related work. Section 4 presents the proposed methodology. Section 5 presents a scenario of applying this methodology to a portion of a real-world database. Our main conclusions and future work ideas are shown in the final section.

2. Watermarking Relational Databases

In this section, basic watermarking schemes for relational databases are shown. A taxonomy of possible attacks on such databases and a set of features expected from digital watermarking schemes able to resist to such attacks are included as well.

2.1. Basic Watermarking Schemes for Relational Databases

In a nutshell, a watermark is a secret code that can be hidden in multimedia objects, files, databases, etc. The watermark contains the information necessary for proving ownership or tamper detection. Generally, digital watermarking of a database consists of two stages:

- *Watermark Embedding* that includes both the watermark generation and its storage, either into the database itself or to a trusted authority;
- *Watermark Verification* that consists in extraction (computation) of the current watermark from the database and checking it against the watermark that is either hidden into the database or stored at a trusted authority.

In the *Watermarking Embedding Stage* (Fig. 1), first, a watermark is generated based on the original (unmarked) database, using a key (k_{pr}), which is private in most of the techniques and which is known only by the database's owner. Then, the issued watermark is either integrated into the database or transmitted, via a secured channel, to a trusted authority. This way, a watermarked database is obtained. In the *Watermark Verification Stage* (Fig. 2), the trusted entity extracts the watermark from the Suspicious Database (SDB) and compares it with the original watermark information. This trusted entity can be either a legal authority or the database owner. The private key is the same in both stages. Even a user who does not know this key can still make a request for watermark verification to the trusted authority.

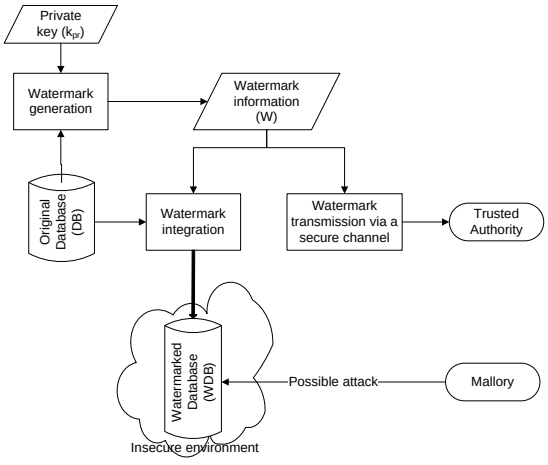

Fig. 1. Watermark Embedding Stage.

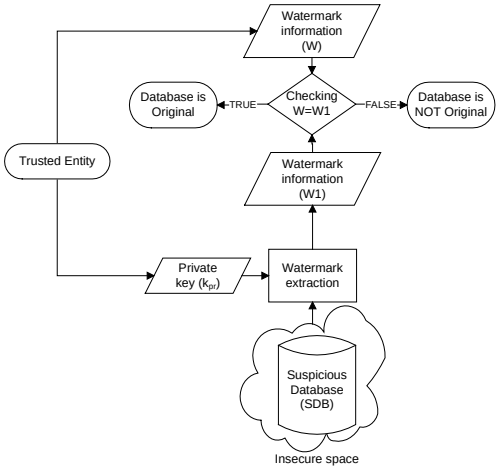

Fig. 2. Watermark Verification Stage.

2.2. Taxonomy of Possible Attacks and Features of Watermarking Schemes

Generally, digital watermarking schemes are designed for integrity verification of database relations (*fragile watermarking*) or copyright protection (*robust watermarking*). Fragile watermarking is used to both detect and localize modifications in database relations. Two main attack types exist, which aim at *the data sets* or *the relation watermark* [10]. The first one refers to alterations of relations through inserting or deleting tuples, changing attribute values, reordering tuples, or changing data value ranges. This kind of attack is successful when a relation is altered, while the embedded watermark remains unaltered. Second type of attack consists in either removal or distortion of the embedded watermark or making it undetectable. In [11], the authors show two kinds of attacks (subset and superset) that act simultaneously on relations and their watermark. Thus, Mallory deletes or updates a subset of tuples or attributes with the intention of losing the watermark (*subset attack*) or new tuples or attributes are added aiming at affecting watermark detection (*superset attack*). In a *brute force attack*, the attacker tries to guess the private information by checking all the possible values of the key. In a *bit attack*, which is directed to the watermark, the attacker attempts to alter one or more bits in the watermarked data aiming to destroy the watermark itself [2]. In a *collusion attack*, many watermarked copies of a relation are compared to each other in order to locate the positions in which they are different, i.e. the locations of altered bits. The watermark information must be carefully chosen and inserted in the same position in all the copies of a relation to achieve resistance to a collusion attack [19]. In a *secondary watermark attack* an attacker can successfully generate his own watermark and embed it in a relation. Using his watermark, he can claim the property over the relation [13]. The collusion attack is also identified as a *comparative attack*, while a secondary watermark attack is also known as a *pseudo-property statement attack* [26].

Based on the variety of possible attacks on watermarked relational databases, a set of desirable features of digital watermarking schemes have been identified in the literature [1], [7], [10], [11], [24]. Depending on the application, some may be prevalent to others. These digital watermarking features are as follows:

- *Robustness* measures the watermark resistance to different kinds of attacks;
- *Imperceptibility* means accepting slight data distortions during watermarking provided those do not affect the database usefulness;
- *Watermarks undetectability* is provided by the *w*atermarking techniques;
- *Blindness*, i.e. the original database is not needed during watermark verification;
- *Security* is ensured by using a private key, which is known only by the owner and by some trusted users, and, which has an appropriate length necessary to resist to brute force attacks;
- *Publicity* - according to Kerckhoffs' principle, a cryptosystem should remain secure in spite of the fact that all the details about the system are public, except for the secret key [17]. So, a watermarked database should be secure even if all the details about the watermarking schemes are known, with exception of the private information (e.g. the private keys) [1];
- *Incremental watermarking* – in case of a database update only the watermark for the altered data is re-computed [1];
- *Localization of alterations* is possible using the watermark detection algorithm;
- *Negligibility of false positives and false negatives* – a false positive is obtained if a watermark is detected for an unwatermarked relation and a false negative if a valid watermark is not detected for a watermarked relation that has been attacked;
- *Computational feasibility* refers to watermarking schemes (both watermark embedding and verification algorithms) that are computationally feasible;
- *One-way watermark generation algorithm* refers to the fact that given the watermarked database and the algorithm, no private information can be inferred.

3. Related Work

The first digital watermarking scheme for relational databases was proposed by Agrawal and Kiernan in 2002 [2]. Their technique ensures that *some bit positions of some of the attributes of some of the tuples contain specific values*. The tuples, the attributes, the bit positions inside an attribute, and specific bit values are all algorithmically determined under the control of a private key known only to the database owner. *This bit pattern constitutes the watermark.* Only if one has access to the private key can detect the watermark with high probability and it can be detected *even in a small subset of a watermarked relation as long as the sample contains some of the marks*. Moreover, detecting the watermark neither requires access to the original database nor the watermark. This watermarking technique has some of the necessary characteristics, such as robustness, incremental updatability, and blind watermarking, but it *only allows watermarking numerical attributes (with minor alterations to the data).* This scheme also ensures copyright protection. In [1], Agrawal, P. J. Haas, and J. Kiernan propose a modification to their previous method based on using a cryptographically secure pseudorandom sequence generator for the selection of the tuples to be marked. However, using their technique a watermark can be applied to any database relation having attributes *which are such that changes in a few of their values do not affect the applications* [2].

The fragile watermarking scheme proposed in [10] verifies the integrity of a database and detects malicious modifications made to a relation. The watermark inserting algorithm is based on operations of grouping and sorting tuples of a relation and on computing values of hash functions. In this technique, two kinds of watermarks are used: *a tuple watermark* and *an attribute watermark* that, together, form a watermarks' grid that enables the detection of any modification. The algorithm works based on the assumption that each attribute's value *allows changing of at least two of the least significant bits*. In the watermark detection procedure, all tuples are divided into groups and each group is verified. The embedded watermarks form a watermark grid that provides for tamper detection and localization. This scheme has the advantage that allows localization of alterations, but the watermark insertion time is bigger than the similar time in the scheme proposed in [2].

In [14], the authors propose using genetic algorithms for minimization of the distortion-induced effects. Thus, different attributes are checked against the optimal criteria, rather than using less effective attributes for watermark insertion. Given that the distortion caused by difference expansion can facilitate the prediction of the watermarked attribute, distortion tolerance of different attributes is explored. Moreover, their approach is reversible and hence distortion introduced after watermark insertion can be fully restored. To avoid distortions in databases, watermarking scheme that use a certification code [25] or a watermark certificate [12] have been devised. In [12], the watermark is generated from the most significant bits of each attribute of a relation, virtually sorted previously based on the hash values of attributes' names. The watermark verification is done by comparing the watermark with the corresponding certified one. Similarly, in [20], the authors propose a fragile watermarking scheme for database tables that is based on virtual sorting and partitioning of the data in disjunct groups obtained when applying a hash function to both the primary key and the private key. The watermark is integrated by reordering the tuples in each independent group. This schema does not introduce distortions on the data, *but it only works for relations that contain categorical data*. A watermarking scheme based on hierarchical sorting is introduced in [27]. On level 1, the database is divided in rows and columns groups and the watermark information is embedded in each group based on an embedding function. At level 2, the groups defined at level 1 are seen as rows and columns and each of them is also divided in groups of rows and columns. The watermark embedding algorithm does not alter the attributes' values; it only rearranges the entities in the formed groups.

A relatively new watermarking technique for relational databases that marks both the tables and the relevant indexes is introduced in [15]. It is based on the R-tree data structures that store the data. The main goal of this technique is to identify unauthorized updates to the R-trees. Authorized modifications update also the watermark as needed. The watermark is hidden in the set of entries (or minimum bounding rectangles) by re-arranging them inside the R-tree's nodes

in a specific way that corresponds to the value of the watermark W. This re-arrangement is performed relatively to an initial secret reference order. The watermark information includes all these pieces of information that can be binary images as in [4], indexes, and so on. In [4] the watermark information for a tuple is the bit string <1,1,1,0> and more such strings make a binary image that correspond to a tuple partition (they use the term partition there for a group of tuples). Watermark information is obtained from the hash values computed from both the values of some attributes and the primary key of each table.
A technique that aims at minimizing distortions in the original data set is proposed in [16]. It generates the watermark bits by taking date-time stamp as an input. Distortion minimization is based on enforcing a tolerance level on attributes' alterations when watermarking a table.

4. Towards a Methodological Approach of Relational Database Watermarking

All the schemas presented in the previous section focus on *marking some relations, independently from each other, and some indexes, some use rearrangements and grouping of tuples based on hash values, while some can be used only on numerical value attributes that can stand minor alterations or only on categorical data. Also, they are all focused on database instance without considering the database schema.* In our view, these approaches are only the first step toward systematic database watermarking, which will be performed entirely by the database management system like transaction management. The core idea of relational databases consists in storing both real-world entity types (sets) and their relationships and, therefore, database watermarking should be driven by these main elements. Moreover, database watermarking needs to take into account both database schema and instance, data semantics and dynamics, the metadata in the data dictionary, and database tuning elements. Of course, neither watermarking each and every table and each and every index nor watermarking only some of them without considering the others can be appropriate.

4.1. A Methodology for Relational Database Watermarking

In this section, we introduce our methodology for relational database watermarking (Fig. 3). The main premise is that in order to fully comply with the relational data model, *no distortions of the data are accepted*. The word "table" is used generically for actual data tables, but also for indexes, tables of the data dictionary, etc.

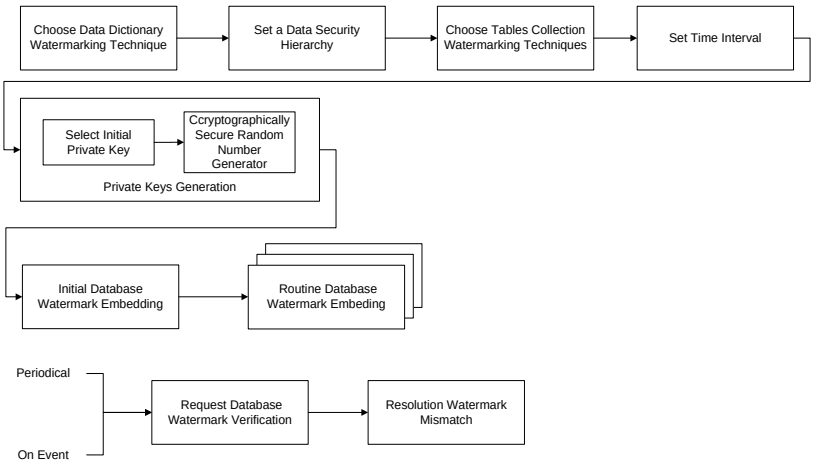

Fig. 3. A methodology for relational database watermarking

Choose the Watermark Technique for the Data Dictionary: the data dictionary is vital for relational databases. Any successful attack on it would have catastrophic consequences for the entire database. A strong distortion-free watermarking scheme is used to generate the watermark, which is then recorded at the database owner or at a trusted authority, or both.

Set a Data Security Hierarchy: all data are not created equal with respect to security requirements, so a security hierarchy has to be established. On each level, tables with same security requirements will be found. This stage is necessary because watermarking a table or more can be a very time (and other resources) consuming activity, and therefore, the least security-critical tables should be marked not as often and/or not as much as the most critical ones. The hierarchy presented in Fig. 4 includes, on the level 0 the most critical parts of the data dictionary, on the levels 1 and 2, some other parts of it along with very important data, and on level 3 only regular data. The tables are clustered together according to the real-world relationships between their stored data (in terms of The Entity Relationship Model).

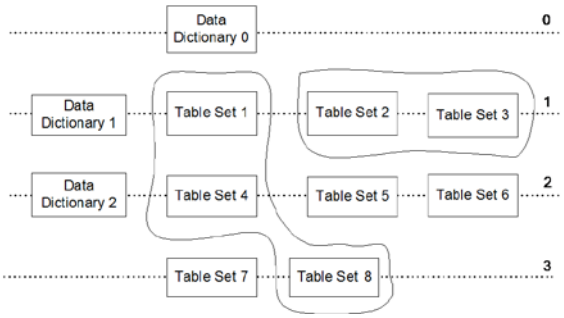

Fig. 4. Table clusters in the data security hierarchy.

Choose the Appropriate Watermarking Techniques: our approach is to *cascade watermarking* the tables from a cluster following the relationships between the data. In the watermark generation step, watermarkers for tables, tuples, attributes, and also for the entire database will be computed based on a hash function that depends on the private keys and/or the primary keys. Hash values are used in watermarking because they provide for special security properties: preimage resistance, second preimage resistance, and collision resistance [23]. In case of normal data manipulation, only the watermark information for the database elements involved in modifications will be recomputed. Two watermarking schemes are needed: a strong initial one that is used for watermarking the initial version of a database and a routine one that watermarks database elements periodically based on their security level. In the *strong watermarking* scheme all the relations are considered security-critical and the watermarking is performed based on all the data in tables. Of course, this is time and computing power consuming, but it is worthwhile doing that from time to time. In the *routine watermarking* scheme, the marking is made using a different scheme and based on the security level of each table. For example, for tables with high security needs (found on the top levels of the hierarchy) the watermarking scheme is based on all the data in the tables, while for ordinary ones the watermarking can be based only on some of their data. The tables are watermarked in cascade based on the relationships between the data and on dynamic of data changes. For instance, in case of an one-to-one(many) relationship between the relations R_i and R_j, when a tuple in the relation R_i is selected to be marked, the corresponding tuple(s) in the relation R_j will be selected for marking as well, provided that it has not been marked already. This way, the tuple watermarks are generated. The selection of the attributes to be marked is based on their use frequency, for example, the unique number that identifies a company is used for fiscal issues far more often than any other numerical values. A relation's watermark depends on both the tuples' watermarks and the attributes' watermarks. A database watermark is an aggregation function of the relations' watermarks. As alterations of the actual data are not accepted in our methodology, in this stage only distortion-free schemes may be used. Such schemes generally use a secret virtual partitioning of tuples in groups based on the secret key, and then generate each group's watermark and the whole table's watermark, and, finally, store the watermark information [3,4,5].

Schedule Watermarking Operations: timestamps are used for scheduling watermarking operations (embedding or verifying). They will be included in the database security procedures and will depend on the content and functionality of a database, on the importance of the stored data, on the categories of users etc.

Private keys' generation: the initial private key is chosen so that the desired security level is obtained, e.g. it has to be long enough to resist to brute-force attacks. Private keys are generated using various methods for cryptographically secure random number generation having as a seed the initial private key. One or more private keys may be used according to the watermarking scheme. During the database lifecycle many private keys will be used.

Initial and Routine Database Watermark Embedding: a database is watermarked first after its initialization and afterwards periodically, by its database administrators, in accordance with the security policies and procedures. Therefore, this methodology needs to be customized for each database to be watermarked. For instance, the time interval between successive routine watermarking is set based on the *database semantics and dynamic* and also on *the needed security level*. Thus, when the number of updated records in database tables reaches a database specific threshold, the routine watermarking is performed (incrementally whenever possible). This approach can provide for increased security in case of (portions of) databases with critical security needs, while some others may not be watermarked quite as often. Watermarks' logging is necessary for easy monitoring of security breaches and attacks.

Request Watermark Verification: watermark detection is performed at various levels of the database. First, the database watermark is computed and checked and, in case of a mismatch, the watermarks of relations are verified. When a relation watermark mismatches, the tuple and attribute watermarks are checked as well. This approach provide for localization of unauthorized alteration within the database. Watermark detection is performed either periodically, in accordance with the security policies and procedures or in case of an attack suspicion. The recovery to a consistent state relies on identifying the tuples and columns altered by the attack and recovering them using both the available backups and specific techniques depending on the watermarking schemes used.

Resolution Watermark Mismatch: in case of a watermark detection mismatch, a specific procedure is activated that generates an alert message and/or block the access to the entire database or only to some portion of it (the affected part), and then analyze the attack (its source, its effects, if it still produces effects etc.). Further on, corrections to the database are applied to get the database in the last consistent state before the attack.

4.2. Additional Features for Database Watermarking Schemes

In addition to the database watermarking features presented in Section 2.2, to make possible to use our methodology some extra features are required, such as *extended robustness* with respect to ensuring data integrity, preserving relation schemes and data constraints, and so on; *data semantics-driven*, i.e. the watermarking process has to be driven by the semantics of the data and not by some particular data in some relation instances; *distortion-free* because allowing distortions may violate the data consistency; *private key strength* that requires choosing private keys that are able to resist to brute force attacks; *time effectiveness* with respect to both watermark generation and detection; *watermark logging* for scrutinizing of security breaches and attacks and easily detecting the malicious users; *adaptive watermarking* based on the database semantics and dynamic and also on the needed security level.

The proposed methodology provides for watermarking both database schema and instance, taking into account both data semantics and dynamics, as well as its importance based on a data security hierarchy, enforcing data consistency in accordance with the relational data model, saving computational power and other resources because only the altered records that represent related data across database tables are cascade watermarked, while still ensuring the security requirements, narrowing down quickly to the groups and records affected by unauthorized database alterations, using in both public and private databases, extra security for databases, as a whole, and, consequently, it can be a first step towards the systematic database watermarking, the main goal being to have the database management system performing the watermarking similarly to transaction management. Relying on distortion-free schemes is a significant shortcoming for the time being, as the scarcity of such schemes that are easy to implement is a fact.

5. A Scenario of Using the Methodology

Our goal here is to see how this methodology works on a portion of a real database (called wqdb) that is a part of a quote management information system (both for products and services). The database has 30 tables, each of them holding hundreds of tuples. As this is a legacy database that has not been normalized yet, the relations are in various normal forms. The relations Quotes and Items_quotes contain data about the product quotes, while the relations Quotes_services and Items_ quotes_services include data about the service quotes. The Products and Services tables contain, respectively, information about the available products and services. The Customers table contains data about the clients. The main relation schemas are as follows (the primary keys are underlined; some fields are left out):

Customers (customer_code, customer_name, uic, no_reg_commerce, country, city, street, no, phone, fax, email, contact_person, bank, account). The fields that are not self-explanatory mean, respectively: uic (company unique identification code), no_reg_commerce (trade register identifier); *Products* (catalogue_code, short_name, description, warranty_period); *Quotes* (no_quote, version, state, date_quote, id_user, customer_code, delivery_term, valid, standard, total_value, global_discount, quote_value, advance). The fields that are not self-explanatory mean, respectively: state (i.e. sent to client, under processing, etc.), id_user (the employee who made the quote), valid (when true, an order based on the quote was placed), standard (quality_standard), quote_value (the quote value after applying the discount to the total value); *Services* (service_code, service_name, wqdb_lst_price); *Quotes_services* (no_quote, ver, area, state, date_quote, id_user, customer_code, delivery_tem, valid, standard, total_value, global_discount, quote_value, advance) – this contains data about the quotes that correspond to the services provided by the company, the meaning of the fields being similar to the above ones; *Items_quotes* (no_quote, ver, no_item, catalogue_code, name_for_quote, purchase_price, supplyer_discount, supplyer_list_price, wqdb_discount, wqdb_list_price, quote_discount, quote_price, quantity, value) – this corresponds to an entry to the quotes' catalogue for products, the meanings of the non self-explanatory fields being as follows: no_item (entry index), name_for_quote (the name of the product), wqdb_discount (the minimum discount for that product), quote_discount (the actual discount, as a percent), quote_price (as wqdb_list_price - wqdb_list_price*quote_discount), and value (as quantity*quote_price); *Items_quotes_ services* (no_quote, ver, no_item, service_code, value_service – this corresponds to an entry to the quotes' service catalogue.

According to our methodology, after initializing a database, a strong cascade fragile watermarking is applied to *all the data* (so, initially, everything is watermarked). Periodically, a routine watermarking scheme is applied, as follows. *First step is to choose a watermarking scheme for the data dictionary*. Taking into account the data dictionary importance, that scheme uses all its rows and columns. We used a part of the data dictionary with 116 records and 23 columns and two kinds of watermarks, one for tuples and one for attributes, which, together, form a watermarks' grid (as in [6]). A private key has been generated using a cryptographically secure pseudo-random number generator. Usually, large dimension keys (more than 128 bits) are used because they provide for a reasonable level of security. In our case, the generated key was (3832765987080). However, several private keys can be used and the same key may be used throughout the whole procedure or not. Further on, for each pair (tuple, column) of the dictionary a hash value is computed, based on the tuple data, the column data, and the selected private key. Basically, the watermark information associated with the data dictionary consists of two sequences of hash values, one for tuples and one for columns that will be stored at a trusted authority. In the verification stage, the watermark information is extracted from the suspicious data dictionary and verified against the stored watermark information. If any difference exists then the database has been altered and, therefore, the procedures for watermark mismatch resolution are launched. We have chosen this fragile watermarking technique because it allows locating the modification in case of an unauthorized alteration.

Set a Data Security Hierarchy: this hierarchy takes into account both the importance of the data stored in the database tables and the operating mode on the database (some data may be accessed from within the organization, some not, various categories of users etc.) – Fig. 5.

Choose the Appropriate Watermarking Techniques: further on, a cascade watermarking only on the tables Quotes, Customers, and Item_quotes is illustrated. The connectivity of the relationships between Customers and Quotes, and, respectively, Quotes and Item-quotes, is one-to-many. We started the watermarking with the table Quotes, based on the fact that in the real world, many customers may exist without having associated quotes yet. Suppose that the size of these three tables is as follows: Quotes has 1600 records, Items_quotes has 3400 records, and Customers has 500 records. The cascade watermark embedding works this way: when a tuple of Quotes is selected to be marked, the corresponding tuples from Items_quotes and Customers will be marked as well, following the relationship between the data.

Private keys' generation: this stage is important in our methodology because choosing keys cryptographically unsecure can lead to database vulnerability to brute force attack. To show how our methodology works, two different keys have been used: one for the cascade watermarking of the database and one for watermarking the data dictionary. Prime numbers are fundamental for key generators and, therefore, we used the prime number 991 for exemplification (to ensure real protection, a much larger number would be necessary). Using it as a seed, we generated an integer **g** that is the number of clusters to be formed. For efficiency reasons, this number need to be much smaller than the minimum between the number of tuples of Quotes and Customers The **g** number (obtained using the C# Random class) was 79, which is conveniently smaller than 500 (the minimum). Next, we computed a hash value based on the concatenation between the values of the primary key and the seed. Based on this hash value, the group to which each tuple belongs is determined. Cascading further, the same number of groups is taken into account and the tuples of Items_quotes are grouped following the relationship between the data in Quotes and the data in Items_quotes. E.g., for the record in Quotes identified by no_quote=975 and ver=3, we computed a hash value (9753991) mod 79 and the obtained result, 47 is the group number in which this record will belong. Moreover, all the records in Items_quotes for which no_quote=975 and ver=3 will belong in group 47 of the table Items_quotes. The Customers' clusters are constituted similarly, based on the equality between the values of customer_code in Quotes with custo-mer_code in Customers. For the record with no_quote=975 and ver=3, the customer_code is 131. As a result, the record of Customers identified by customer_code=131 belongs to group 47. In fact, for each table, all the tuples have been partitioned this way (Fig. 6). The selection of the attributes that will contribute to the hash values is made based on the dependencies between the data (that show data semantics) and, in some cases, all of them may be necessary.

During the stage of *Request Watermark Verification* in case of tamper detection, first the altered group is identified based on group watermark information, and then the altered tuple based on tuple watermark information. During the stage of *Resolution Watermark Mismatch* procedures to get the database into a consistent state are triggered, along with other specific ones such as establishing the attacker's identity, identifying the attack type, updating the watermarking schemes and procedures, and, of course, getting the whole database updated.

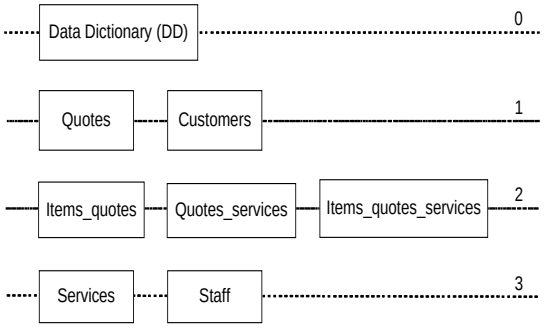

Fig. 5. The data security hierarchy of wqdb database.

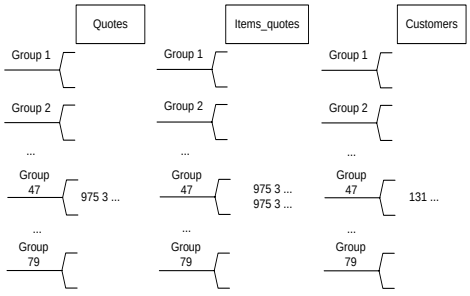

Fig. 6. Cascade watermarking in wqdb database.

6. Conclusion and Future Work

Expansion of the digital world based on more and more sophisticated pervasive systems both outwards, to the environment we live in, and inwards, to our inner self, opens up new challenges for ensuring both security and safety within this complex and pervasive world. Current trends in development of information systems consist in linking users and databases within socio-technical systems, and in increasing the importance of organizational factors. However, cyber attacks have increased as well and they are more and more ingenious every day. The attackers can be outsiders or insiders. Sometimes, trusted persons can steal or modify data in an indistinctive way. For example, a bank employee can transfer periodically a infinitesimal amount of money from client accounts to a secret account he or she has access to; clients may not realize that is happening, the amount being really insignificant, but in time, the damage may be important. Copyright protection is also a challenge within the emerging open world. Watermarking has been used successfully mainly for files that contain images, multimedia, or documents. Recently, significant progress has taken place also in case of data stored in tables. A challenge still remains with respect to relational databases. In more and more domains, such as research, education, health etc., databases are made public and anyone can use their content. Therefore, they are more and more vulnerable. But also private databases that are networked and, therefore easily accessible online by various users, including mischievous ones, are equally vulnerable. Many of the database security issues can be approached by using database watermarking in conjunction with other security techniques (data encryption, digital signatures, digital certificates, access control, identity control, etc.).

The main contribution of this paper is dual and consists in designing from scratch a methodology for relational database watermarking and put it to work on a portion of a real database. The main lesson learned is that *a database can be watermarked as a whole, while still ensuring the kind of data consistency and integrity expected of relational databases*. Furthermore, whenever possible, the watermarking scheme for a relational database should be devised during the database design stage. We are currently working to integrate this methodology as a service offered by the database management system itself.

Nevertheless, using such a methodology is only a first step towards a systematic approach of relational database watermarking. The days when the watermarking procedures will be provided by the database management system itself, in a similar way with transaction control, are still to come. Future work is possible in the meantime with regard to performing some experiments with various (types of) databases and database management systems, evaluating performance and overheads, clarifying some methodological aspects, identifying the most appropriate watermarking schemes with respect to various database categories, its scalability, and, finally, to documenting the methodology better to facilitate experimenting with it.

References

1. Agrawal, R., Haas, P. J., Kiernan, J.: Watermarking Relational Data: Framework, Algorithms and Analysis. Int J Very Large Databases 12(2), 157–169 (2003)
2. Agrawal, R., Kiernan, L.: Watermarking Relational Databases. In: Proc. of 28th Int. Conf. on Very Large Data Bases, pp. 155–166. VLDB Endowment (2002)
3. Bhattacharya S. and Cortesi, A.: Distortion-Free Authentication Watermarking Software and Data Technology. In: 5th International Conference ICSOFT Revised Selected Papers, vol. 170, pp. 205-219 (2013)
4. Bhattacharya, S., Cortesi, A.: A Generic Distortion Free Watermarking Technique for Relational Databases. In: Proceedings of the 5th Int. Conference on Information Systems Security, LNCS, vol 5905, pp. 252--264, Springer, Heidelberg (2009)
5. Camara, L., Junyi, L., Renfa, L., Wenyong Xie, X.: Distortion-Free Watermarking Approach for Relational Database Integrity Checking. Math Probl Eng (2014)
6. Contreras, J. F., Coatrieux, G.: Protection of Relational Databases by Means of Watermarking: Recent Advances and Challenges, Advances in Security in Computing and Communications. Jaydip Sen (ed.), InTech. 101-123 (2017)

7. Cox, I. J., Miller, M. L., Bloom, J. A.: Watermarking applications and their properties. In: Proceedings of the International Conference on Information Technology: Coding and Computing, pp. 6-10. IEEE (2000)
8. Development Research Group, World Bank, Research Program Datasets, http://econ.worldbank.org/research, Accessed July, 21, 2017
9. EU Open Data Portal, http://open-data.europa.eu/en/data/#,Accessed March, 24, 2018
10. Guo, H. et al.: A Fragile Watermarking Scheme for Detecting Malicious Modifications of Database Relations. Inform Sciences 176(10), 1350–1378 (2006)
11. Halder, R., Pal, S., and Cortesi, A.: Watermarking Techniques for Relational Databases: Survey, Classification and Comparison. J Univers Comput Sci 16(21), 3164-3190 (2010)
12. Hamadou, A., Sun, S., Gao, L., Shah, S. A.: A Fragile Zero-Watermarking Technique for Authentication of Relational Databases. Int J of Digital Content Technology and its Applications 5(5), 189-200 (2011)
13. Iqbal, S., Rauf, A., Javed, H., Ahmed, S.: Distortion Free Algorithm to Handle Secondary Watermark Attack in Relational Databases. In: Proc. of the 2nd International Conference on Information Management and Evaluation, pp. 214-222. Academic Conferences and Publishing International Limited (2011)
14. Jawad, K., Khan, A.: Genetic Algorithm and Difference Expansion Based Reversible Watermarking for Relational Databases. J Syst Softw 86(11), 2742-2753 (2013)
15. Kamel, I.: A Schema for Protecting the Integrity of Databases. Computers and Security 28(7), 698 – 709 (2009)
16. Kamran, M., Suhail, S., Farooq, M.: A Robust, Distortion Minimizing Technique for Watermarking Relational Databases Using Once-for-All Usability Constraints. IEEE T Knowl Data En 25 (12), 2694-2707 (2013)
17. Kerckhoffs, A.: La Cryptographie Militaire. J des Sciences Militaires IX, 5–38 (1883)
18. Khataeimaragheh, H., Rashidi, H.: A Novel Watermarking Scheme for Detecting and Recovering Distortions in Database Tables. Int J DB Manag Syst 2(3), 1-11 (2010)
19. Lafaye, J., Gross-Amblard, D., Constantin, C., Guerrouani, M.: Watermill: An Optimized Fingerprinting System for Databases under Constraints. IEEE T Knowl Data En 20(4), 532-546 (2008)
20. Li, Y., Guo, H., Jajodia, S.: Tamper Detection and Localization for Categorical Data using Fragile Watermarks. In: Proceedings of the 4th ACM Workshop on Digital Rights Management, pp. 73–82. ACM New York (2004)
21. MIT Lincoln Laboratory, DARPA Intrusion Detection Data Sets, http://www.ll.mit.edu/ideval/data/index.html, Accessed March, 23, 2018
22. Popa, R. A., Redfield, C. M. S., Zeldovich, N., and Balakrishnan, H.: CryptDB: Protecting Confidentiality with Encrypted Query Processing. In: Proc. of the 23rd ACM Symp. on Operating Systems Principles, pp. 85-100. ACM New York (2011)
23. Rogaway, P., Shrimpton T.: Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In: Roy B., Meier W. (eds) Fast Software Encryption 2004. LNCS, vol 3017, pp. 371-388. Springer, Berlin, Heidelberg (2004)
24. Tao, H., Chongmin, L., Zain, J. M., Abdalla, A. N.: Robust Image Watermarking Theories and Techniques: A Review. J. Appl Res Technol 12(1), 122–138 (2014)
25. Tsai, M. H., Tseng, H. Y, Lai, C. Y.: A Database Watermarking Technique for Temper Detection. In: Online Proceedings of the 2006 Joint Conference on Information Sciences, Atlantis Press (2006)
26. Tzouramanis, T.: A Robust Watermarking Scheme for Relational Databases. In: Proceedings of 6th International Conference on Internet Technology and Secured Transactions, pp. 783-790. IEEE (2011)
27. Wenyue, Z., Jie, G.: A New Watermarking Scheme for Database Authentication. Presented at International Conference on Energy System and Electric Power (2011)