This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

RRW - A Robust and Reversible Watermarking Technique for Relational Data

Saman Iftikhar, M. Kamran and Zahid Anwar

Abstract—Advancement in information technology is playing an increasing role in the use of information systems comprising relational databases. These databases are used effectively in collaborative environments for information extraction; consequently, they are vulnerable to security threats concerning ownership rights and data tampering. Watermarking is advocated to enforce ownership rights over shared relational data and for providing a means for tackling data tampering. When ownership rights are enforced using watermarking, the underlying data undergoes certain modifications; as a result of which, the data quality gets compromised. Reversible watermarking is employed to ensure data quality along-with data recovery. However, such techniques are usually not robust against malicious attacks and do not provide any mechanism to selectively watermark a particular attribute by taking into account its role in knowledge discovery. Therefore, reversible watermarking is required that ensures; (i) watermark encoding and decoding by accounting for the role of all the features in knowledge discovery; and, (ii) original data recovery in the presence of active malicious attacks. In this paper, a robust and semi-blind reversible watermarking (RRW) technique for numerical relational data has been proposed that addresses the above objectives. Experimental studies prove the effectiveness of RRW against malicious attacks and show that the proposed technique outperforms existing ones.

Index Terms—Reversible Watermarking, Genetic Algorithm, Data Recovery, Data Quality, Robustness, Numerical Data.

1 Introduction

In the digital world of today, data is excessively being generated due to the increasing use of the Internet and Cloud Computing [1]. Data is stored in different digital formats such as images, audio, video, natural language texts and relational data. Relational data in particular is shared extensively by the owners with research communities and in virtual data storage locations in the Cloud. The purpose is to work in a collaborative environment and make data openly available so that it is useful for knowledge extraction and decision making. Take the case of Walmart- a large multinational retail corporation that has made its sales database available openly over the Internet so that it may be used for the purposes of identifying market trends through data mining [2]. However these openly available datasets make attractive targets for attacks. For example there are documented attack incidents where data containing personal information related to customers using certain Walmart video services [3] was stolen. According to a survey related to the security of outsourced customer data [4], it is reported that 46% of organizations do not consider security and privacy issues while sharing their confidential data. Therefore, 64% organizations have to face data loss repeatedly. Similarly, data breaches in the health care and medical domain are increasing alarmingly [5]. Therefore it is imperative, that in shared environments such as that of the Cloud, security threats that arise from un-trusted parties and relational databases[1] need to be addressed along with the enforcement of ownership rights on behalf of their owners.

Watermarking techniques have historically been used to ensure security in terms of ownership protection and tamper proofing for a wide variety of data formats. This includes images, audio, video [6], [7], [8], [9], [10], natural language processing software [11], relational databases [12], [13] and more. Reversible watermarking techniques can ensure data recovery along with ownership protection. Fingerprinting, data hashing, serial codes are some other techniques used for ownership protection [14]. Fingerprints also called transactional watermarks, are used to monitor and identify digital ownership by watermarking all the copies of contents with different watermarks for different recipients. Primarily this type of digital watermarking tries to identify the source of data leakage by tracing a guilty agent. In hashing, digital contents can be saved by performing a one-way hash function whereby the data contents do not change. If the hash of the original and tampered data is the same, data authenticity can be verified but ownership cannot be proved easily. Serial or classification codes are used for filtering of inappropriate contents over the Internet and are mainly applicable to images,

- *Saman Iftikhar is with the Department of Computing, School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan.*
  E-mail: saman.iftikhar@seecs.edu.pk
- *M. Kamran is with the Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt, Pakistan.*
  E-mail: muhammad.kamran@ciitwah.edu.pk
- *Zahid Anwar is with the Department of Computing, School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan.*
  E-mail: zahid.anwar@seecs.edu.pk

1. The terms, relational databases, data and datasets are used interchangeably in this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

audio and video. Watermarking has the property that it can provide ownership protection over the digital content by marking the data with a watermark unique to the owner. The embedded watermark can subsequently be used for proving and claiming ownership.

Digital watermarking of multimedia content is more commonly known. Particularly image watermarking - a derivative of Steganography is an age-old practice allowing covert transmission of messages from one party to another by exploiting redundancy in common image formats. However the basic process of multimedia watermarking is very different from that used to watermark relational databases because of a fundamental difference in the properties of the data. Multimedia data is highly correlated and continuous whereas relational data is independent and discrete. With the advent of modern copyright protection and information hiding techniques, database watermarking can be used to enforce ownership rights of relational data. However a major drawback of these techniques is that they modify the data to a very large extent which often results in the loss of data quality. There is a strong need to preserve the data quality in watermarked data so that it is of sufficiently high quality and fit for use in decision making as well as in planning processes in different application domains. Data quality can be defined as the appropriateness of data for its intended applications.

Reversible watermarking tries to overcome the problem of data quality degradation by allowing recovery of original data along with the embedded watermark information. This paper proposes one such reversible watermarking technique that keeps the data useful for knowledge discovery. Data modifications are allowed to such extent that the quality of the data before embedding watermark information and after extracting, is acceptable for knowledge extraction process. Consequently, knowledge discovery becomes successful in decision support systems where high quality data recovery is imperative. Reversible watermarking techniques are already available in literature; however, to the best of our knowledge, no work has been conducted on overcoming the problems of reversible watermarking techniques in the presence of malicious attacks. Achieving robustness (attack resilience) in the presence of reversibility (ability to recover the watermark and the original data) is a challenging task. These two features may be potentially conflicting because a reversible watermark string also makes it an easier target for attack. Therefore, we try to find the most appropriate watermark bandwidth that ensures maximum watermark robustness without significant loss of information that may result by watermarking. To this end, we model the bandwidth optimization as a constraints optimization problem.

Constrained Optimization (CO) [15], [16] allows one to optimize a single or multiple objectives with respect to certain variables that are bounded by some constraints. Our motivated problem is also a CO problem, whereby the research communities want to share databases over the public Internet or a Cloud environment for their knowledge discovery processes. Ownership rights of these databases need to protected from malicious recipients; in the presence of data quality constraint. Recent research studies enunciate that computational intelligence techniques, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are a promising branch of evolutionary computation that model hard constrained optimization problems [17] using biological inspired computing algorithms. Digital Watermarking can also be modeled as an optimization problem as demonstrated by some recent research works [18] and [19] that use PSO for watermarking different data formats and the results are quite encouraging. GA - an optimization algorithm is employed in the robust and reversible watermarking technique (RRW) proposed in this paper to achieve an optimal solution that is feasible for the problem at hand and does not violate the defined constraints. An optimal watermark value is created through the GA and inserted into the selected feature of the relational database in such a way that the data quality remains intact.

Our focus is to develop an information model through a statistical measure that identifies such features that do not have a significant effect on the decision making process. Mutual Information [20], [21], a well known information theory (concept), statistically measures the amount of information that one feature contains about the other features in a database. In RRW, mutual information is used to select a suitable (candidate) feature from the database for watermarking. According to literature, existing reversible watermarking techniques, do not take into account the mutual information measure for determining relative importance of features. In RRW, the knowledge of mutual information for every candidate feature is also employed to compute the watermark information. Thus, it is ensured that the data quality will not be affected. Consequently, RRW provides a robust solution for data recovery that is reversible and resilient against heavy attacks.

RRW mainly comprises a (1) data preprocessing phase, (2) watermark encoding phase, (3) attacker channel, (4) watermark decoding phase and (5) data recovery phase. In data preprocessing phase, secret parameters are defined and strategies are used to analyze and rank features to watermark. An optimum watermark string is created in this phase by employing GA - an optimization scheme - that ensures reversibility without data quality loss. In the watermark encoding phase, the watermark information is embedded in the selected feature(s). Two parameters, β the optimized value from the GA and η_r a change matrix are used in the watermark encoding and decoding phases. Finally, the watermarked data for intended recipients is generated. The attacker channel comprises subset alteration, subset deletion and subset insertion attacks generated by the adversary. These malicious attacks modify the original data and try to degrade its quality. In the watermark decoding phase the embedded watermark is decoded from the suspicious

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

data. In order to achieve this the preprocessing step is performed again, and decoding strategies (feature selection on the basis of *MI*, β the optimized value from the GA and η_r the change matrix) are used to recover the watermark. Semi-blind nature of RRW is used mainly for data reversibility in case of heavy attacks (attacks that may target large number of tuples). Original data is recovered in data recovery phase, through post processing steps for error correction and recovery.

The major contributions of our work are: (1) the design of an intelligent reversible watermarking technique for relational data that ensures data recovery without compromising data quality, and (2) a robust data recovery scheme that is resilient against subset alteration, subset deletion and subset insertion attacks. RRW detects the watermark fully and recovers the original data. The robustness of RRW is evaluated through attack analysis, considering the attacker channel. It is worth mentioning that sound watermarking techniques exploit redundancy in the data to embed the watermark in a manner that it does not impact the overall size. For example consider the case of image watermarking that embeds information in the least significant bit (LSB) of pixel values which is imperceptible as well as it does not effect the overall image size. This is desirable because changes in the size of the original data can compromise the presence or absence of a watermark. Similarly one other contribution of our technique is that it avoids detection by keeping the size of the original relational data unaltered.

The subsequent sections of the paper are structured as follows: In Section 2, a brief overview of related research is provided by emphasizing the different directions of our work. In Section 3, the proposed scheme is described in detail. In Section 4, experiments and results are discussed concisely through a short case study of medical data. Finally, the paper is concluded with a discussion to future research in Section 5.

2 Related Work

The first irreversible watermarking technique for relational databases was proposed by Agrawal et al. in [12]. Similarly, the first reversible watermarking scheme for relational databases was proposed in [22]. In this technique, histogram expansion is used for reversible watermarking of relational database. Zhang et al. proposed a method of distribution of error between two evenly distributed variables and selected some initial nonzero digits of errors to form histograms. Histogram expansion technique is used to reversibly watermark the selected nonzero initial digits of errors. This technique is keeps track of overhead information to authenticate data quality. However, this technique is not robust against heavy attacks (attacks that may target large number of tuples).

Difference Expansion Watermarking techniques (DEW), [23], [24], [25] exploit methods of arithmetic operations on numeric features and perform transformations. The watermark information is normally embedded in the LSB of features of relational databases to minimize distortions. Whereas, in RRW, a GA based optimum value is embedded in the selected feature of the dataset with the objective of preserving the data quality while minimizing the data distortions as a result of watermark embedding. Another reversible watermarking technique proposed in [26] is based on difference expansion and support vector regression (SVR) prediction to protect the database from being tampered. The intention behind the design of these techniques is to provide ownership proof. Such techniques are vulnerable to modification attacks as any change in the expanded value will fail to detect watermark information and the original data.

Genetic Algorithm based on Difference Expansion watermarking (GADEW) technique is used in a proposed robust and reversible solution for relational databases [27]. GADEW improves upon the drawbacks mentioned above by minimizing distortions in the data, increasing watermark capacity and lowering false positive rate. To this end, a GA is employed to increase watermark capacity and minimize introduced distortion. This is because the watermark capacity increases with the increase in number of features and the GA runs on more features to search the optimum one for watermarking. However, watermark capacity decreases with the increase in watermarked tuples. GADEW used the distortion measures (AWD and TWD) to control distortions in the resultant data. In this context, the robustness of GADEW can be compromised when AWD and TWD are given high values.

Prediction-error expansion watermarking techniques (PEEW) like [28], [29], [30], [31], [32], [33] incorporate a predictor as apposed to a difference operator to select candidate pixels or features for embedding of watermark information. The PEEW proposed technique by Farfoura et al. is fragile against malicious attacks as the watermark information is embedded in the fractional part of numeric features only. In this particular scenario, the scheme works because the intention of the attacker is to preserve the usefulness of the data; otherwise, he can easily compromise the fractional part. RRW is robust, as the watermark information is embedded in the values of numeric features, to make the scheme resilient against such attacks.

In [34], the authors proposed a robust, blind, resilient and reversible, image based watermarking scheme for large scale databases. The bit string of an image is used as a watermark where one bit from the bit string is embedded in all tuples of a single partition and the same process is repeated for the rest of the partitions. This technique demonstrates a remarkable decrease in watermark detection rate during various types of heavy attacks, and the database tuples get highly distorted. In RRW, a GA is used to generate a parameter that controls the data distortions to make sure that the data quality remains intact after watermarking. Moreover, the semi-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

blind nature of the technique allows robustness against heavy attacks and also for regeneration of the original dataset after watermark decoding.

Gupta and Pieprzyks' [23], proposed reversible watermarking technique introduces distortions as a result of the embedding process. Changes in the data are controlled by placing certain bounds on LSB. On the contrary, to limit the distortions, the data outside the limited bounds is left un-watermarked. As a result, the watermark robustness gets compromised. However, RRW has no such limitations.

The reversible watermarking techniques DEW, GADEW and PEEW, proposed in [23], [27], [28] respectively, are not robust and reversible against heavy attacks. Features are selected in these techniques for watermarking without considering their importance in knowledge discovery. RRW is robust and reversible and copes with the above mentioned problems and data quality is preserved by taking into account the importance of the features in knowledge discovery.

In RRW, all the tuples of the selected feature can be marked thanks to the selection of a low distortion watermark; therefore, the attacker will have to attack all the tuples to corrupt the watermark to mitigate the effect of the majority voting scheme. Attacking all the tuples is not a viable option for the attacker because he has no knowledge of the original data or the usability constraints and that would completely compromise its usefulness. Moreover, since RRW can afford to embed watermark bits in all or a large fraction of the tuples of the selected feature; it achieves high robustness against heavy attacks. However, marking all tuples is not a requirement. RRW is configurable in that the data owner can choose a fraction for watermarking if it is required. RRW outperforms existing state of the art reversible watermarking techniques including DEW, GADEW and PEEW. These techniques embed the watermark in partitions of the data to ensure minimum distortion; therefore, recover original data with degraded data quality and lack robustness. RRW has overcome drawbacks of these techniques and is also resilient against heavy attacks.

3 RRW Architecture

This section discusses RRW for reversible watermarking of relational databases that improves data recovery ratio. The main architecture of RRW is presented in figure 1. RRW includes the following four major phases: (1) watermark preprocessing; (2) watermark encoding; (3) watermark decoding; and (4) data recovery.

The watermark preprocessing phase computes different parameters for calculation of an optimal watermark. These parameters are used for watermark encoding and decoding.

The main focus of watermark encoding phase is to embed watermark information in such a way that it does not affect the data quality. During watermark embedding, data gets modified according to the available

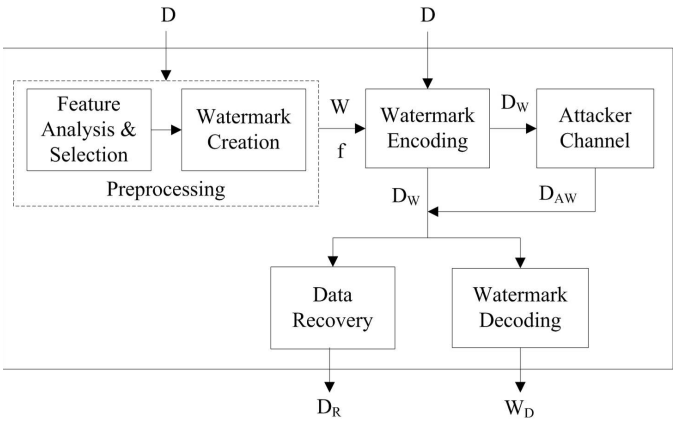


Fig. 1. Main Architecture of RRW

bandwidth (or capacity) of the watermark information. The bandwidth of the watermark should be sufficiently large to ensure robustness but not so large that it destroys the data quality. The data owner decides the amount of data modification such that the quality is not compromised for a particular database application before-hand and therefore defines usability constraints λ to introduce tolerable distortion into the data. The datasets used in our research, are taken from UCI machine learning repository, including, Cleveland Heart Disease dataset [35], MAGIC Gamma Telescope dataset [36] and PAMAP2 Physical Activity Monitoring dataset [37]. It is to note that these datasets were used for evaluation purposes and that RRW is not dependent on any particular dataset or feature. Numerical features can be taken under consideration from any dataset and a suitable feature is determined to embed watermark on the basis of mutual information.

After watermarking, the data is released to the intended recipients over a communication channel that is assumed to be insecure and termed as the "attacker channel" in this research domain. The data may undergo several malicious attacks in the attacker channel. The efficiency and effectiveness of RRW is described through robustness analysis determined by its response to subset insertion, alteration and deletion attacks.

The Watermark decoding phase recovers watermark information effectively for detection of the embedded watermark. Data recovery phase mainly comprises the important task of successful recovery of the original data. For a quick reference, Table 1 lists the notations used in this paper.

In subsequent sections, different phases of RRW are discussed.

3.1 Watermark Preprocessing Phase

In the preprocessing phase, two important tasks are accomplished: (1) selection of a suitable feature for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Notations Used in the Paper

Symbol	Description	Symbol	Description
D	Original database	b	The watermark bit
D_W	Watermarked database	$min(a_w)$	The minimum value a of a feature after watermarking
R	Total number of tuples/rows/records in a table (or dataset)	r	A tuple in the database table
η_r	Detected amount of percentage change in encoding	∇	A matrix containing percent change in data values
$max(a)$	The maximum value a of a feature	ζ	The watermark encoder used for watermark encoding
A	a feature/column/attribute selected for watermarking (D)	$D_W^{'}$	A watermarked database after the malicious attacks
$min(a)$	The minimum value of a feature	$max(a_w)$	The maximum value of a feature a after watermarking
l	The length of the watermark	η_{d_r}	Detected amount of change in the value of a feature after an attack on the watermark bit b
w	Watermark bits	η_{Δ_r}	The difference between the changes detected in the value of a feature during the encoding and decoding process
A_W	The watermarked feature	W_D	Decoded watermark
F	Total number of features in the database	D_W	D watermarked by the proposed scheme
dtW	detected watermark bit	D_r	Recovered Data
MI_O	Mutual information of original data	MI_W	Mutual information of watermarked data
Acc_O	Classification accuracy of original data	Acc_W	Classification accuracy of watermarked data
Sn_O	Sensitivity of original data	Sn_W	Sensitivity of watermarked data
Sp_O	Specificity of original data	Sp_W	Specificity of watermarked data
Δ_{MI}	Change in value of mutual information	a	value of feature A
β	An optimized value to watermark a feature	λ	The usability constraints defined by the data owner
ζ	The watermark decoder used for watermark decoding	RRW	proposed Robust and Reversible Watermarking technique
μ_D	Mean of the original data of RRW	σ_D	Variance of the original data of RRW
μ_{D_W}	Mean of the watermarked data of RRW	σ_{D_W}	Variance of the watermarked data of RRW
μ_d	Mean of the original data of PEEW Technique	σ_d	Variance of the original data of PEEW Technique
μ_{d_w}	Mean of the watermarked data of PEEW Technique	σ_{d_w}	Variance of the watermarked data of PEEW Technique

watermark embedding; (2) calculation of an optimal watermark with the help of an optimization technique.

3.1.1 Feature Analysis and Selection

For developing a decisive information model of various features of the dataset, all the features are ranked according to their importance in information extraction, subject to their mutual dependence on other features. For this purpose, mutual information (*MI*), is exploited, that is an important statistical measure for computation of mutual dependence of two random variables. Mutual information of every feature with all other features is calculated by using equation 1.

$$MI(A,B) = \sum_{a}\sum_{b} P_{AB}(a,b) log \frac{P_{AB}(a,b)}{P_A(a)P_B(b)}. \quad (1)$$

Where *MI(A,B)* measures the degree of correlation of features by measuring the marginal probability distributions as $P_A(a)$, $P_B(b)$ and the joint probability distribution $P_{AB}(a,b)$.

Then *MI* of one feature with all other features is computed using the relation:

$$MI_{f_i} = (MI_{f_{ij}}). \quad (2)$$

Where i, j = 1,2 ..., f_t with i $\neq$ j, and f_t is the total number of features.

The value of *MI* of each feature is then used to rank the features. The attacker can try and predict the feature with the lowest *MI* in an attempt to guess which feature has been watermarked. To deceive the attacker for this particular scenario, a secret threshold can be used for selecting the feature for watermark embedding. In this context, the data owner can define a secret threshold based on *MI* of all the features in the database. The feature(s) having *MI* lower than that threshold can be selected for watermarking. The attacker will not attack the features having large *MI* as in that case the usability of the data will be compromised. Therefore, he will be forced to attack the feature(s) with lower *MI* without concrete knowledge (due to the use of secret threshold) of which features have been watermarked.

3.1.2 Watermark Creation through Genetic Algorithm

For the creation of optimal watermark information, that needs to be embedded in the original data, we use an evolutionary technique; Genetic Algorithm (GA). GA is a population-based computational model, basically inspired from genetic evolution [38]. GA evolves a potential solution to an optimization problem by searching the possible solution space. In the search of optimal solution, the GA follows an iterative mechanism to evolve a population of chromosomes. The GA preserves essential information through the application of basic genetic operations to these chromosomes that include: selection, crossover, mutation and replacement. The GA evaluates the quality of each candidate chromosome by employing a fitness function. The evolutionary mechanism of the GA continues through a number of generations, until some termination criteria is met.

During watermark creation phase, we employed the following major steps of the GA for getting optimal watermark information:

1) Initial random population of binary strings called chromosomes is generated. Gene values of each chromosome represents l-bit watermark string as shown in Fig. 2.
2) Fitness of each chromosome is evaluated by employing a constrained optimized fitness function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

discussed in Section 3.1.2.1.

3) Tournament selection mechanism is applied to get the most appropriate individuals as parent chromosomes.
4) Genetic operations of crossover and mutation are performed on parent chromosomes to create off-springs. A single point crossover operator is applied to evolve high quality individuals, inheriting parental characteristics, by exchanging information between two or more chromosomes. A uniform mutation operator is applied to bring diversity in population through small random changes in gene values of binary chromosomes. The values of crossover fraction and mutation rate are set empirically and are given in Table 4.
5) Elitism strategy is applied to hire two individuals with best fitness value; as elites to the next generation without genetic changes.
6) Remaining population of the next generation is created by replacing less fit individuals of the previous generation with the most fit newly created off-springs.
7) Step 2 to 6 are repeated until MI_O and MI_W reach approximately equal values for a certain number of generations.
8) Both, optimal watermark information string and best fitness value (β) is returned after the fulfillment of the termination criteria.

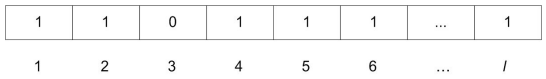


Fig. 2. The Structure of GA Chromosome Used in the Proposed RRW technique for Watermark Creation

3.1.2.1 Constrained Optimized Fitness Function: In the proposed scheme, the GA is populated with a constrained fitness function to acquire an optimal change in data that will ensure data quality while embedding the watermark. The gene values of chromosomes are used to decide which action will be performed on selected feature by using a value of β. Initially, some suitable value of β is set to embed a bearable change in feature values.

The optimal fitness value obtained through GA is basically the change - β - to be embedded in the original data that needs to be watermarked. The purpose of getting an optimum value of β is to justify the amount of change that a feature value can withhold without compromising the data quality. The fitness function can be mathematically expressed as:

$$\begin{aligned} &\max \beta_A, \ \ where \ \ A \epsilon F, \ \ \forall A \\ &subject \ \ to \ \ \lambda \end{aligned} \tag{3}$$

Data quality is ensured by imposing the following usability constraints λ on original and watermarked data given in equation 4.

$$\begin{aligned} \mu_D - \mu_{D_W} &= 0 \\ \sigma_D - \sigma_{D_W} &= 0 \\ MI_W - MI_O &= 0 \end{aligned} \tag{4}$$

where MI_O, μ_D and σ_D denote mutual information, mean and variance of the original data and MI_W, μ_{D_W} and σ_{D_W} represent mutual information, mean and variance of the watermarked data respectively. The minimum and maximum values of the selected feature should also be the same both before and after watermark embedding as denoted in equation 5.

$$\begin{aligned} min(a) &= min(a_w) \\ max(a) &= max(a_w) \end{aligned} \tag{5}$$

In the watermark encoding phase, the optimized value of β is embedded in the particular feature that is selected to be watermarked on the basis of the selection criteria discussed in Section 3.1.1. The gene values of the best chromosome are responsible for deciding which mathematical operation will be performed on the selected feature by using the optimized value of β. Subsequently, this optimal watermark information is decoded from the specified dataset successfully and also needs to be decoded during the decoding process by the data recipient.

3.2 Watermark Encoding Phase

Watermark information calculation is formulated as a CO problem to meet the data quality constraint of the data owner. A GA is used to create optimal watermark information that includes: (1) Optimal chromosomal string (watermark string of length l); and (2) β value. β is a parameter that is computed using GA and represents a tolerable amount of change to embed in the feature values. Once the optimum value of β for each candidate feature A is found, it is saved for use during watermark encoding and decoding. A watermark (bit string) of length l and an optimum value β is used to manipulate the data provided it satisfies the usability constraints λ. The value β is added into every tuple of the selected feature A when a given bit is 0; otherwise, its value is subtracted from the value of the feature.

It is ensured that the mutual information of a feature remains unchanged, when the watermark is inserted into the database. The watermark is inserted into every tuple for the selected feature of the dataset. The data owner can select any number of features for watermark embedding based upon a secret threshold and *MI* of the feature(s).

After finding the optimum value of β, a parameter η_r is calculated according to the equation 6, that represents the percentage change in the watermark encoding. This parameter is calculated for a tuple r as:

$$\eta_r = D_r * \zeta \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

Since the length of the watermark is l; η_r is calculated and β is inserted l times in the database. The length l of the watermark should be carefully chosen. If it is too small, it will make the watermark fragile against attacks, and if it is too large, it might compromise the data quality because the data gets altered for every bit of the watermark. In RRW, the data gets altered for each watermark bit in every tuple. After a number of empirical studies, a length of 16 bits was selected. It is also worth mentioning here that RRW allows a user defined number for watermark bit length.

The watermark encoding algorithm starts the embedding process with the most significant bit MSB of the watermark. For this purpose the algorithm works with one tuple at a time. If the MSB of the watermark is 1, the new value of D_r denoted by D_{W_r} is calculated using equation 7 as follows.

$$D_{W_r} = D_r - \beta \tag{7}$$

In order to embed the second MSB of the watermark, the algorithm is again employed using the same procedure, but the updated value D_r of the feature (that has now become D_{W_r}) is used for calculating new values of η_r and D_{W_r}. If the algorithm encounters a watermark bit that is 0 then the new value of D_{W_r} is calculated using equation 8.

$$D_{W_r} = D_r + \beta \tag{8}$$

The above procedure is repeated until all tuples of the dataset have been watermarked. Algorithm 1 describes the steps involved in the watermark encoding phase.

Algorithm 1 Watermark Encoding

Input: D, w,β
Output: D_W, ∇
  for $w = 1$ to l **do**
    //loop will iterate for all watermark bits w from 1 to length l of the watermark
    for $r = 1$ to R **do**
      //loop will iterate for all tuples of the data
      if $b_{r,w} == 0$ **then**
        // the case when the watermark bit is 0
        changes are calculated by using equation 6
        data is watermarked by using equation 8
        insert η_r *into* ∇
      end if
      if $b_{r,w} == 1$ **then**
        // the case when the watermark bit is 1
        changes are calculated by using equation 6
        data is watermarked by using equation 7
        insert η_r *into* ∇
      end if
    end for
  end for
  return D_W, ∇

3.3 Watermark Decoding Phase

In the watermark decoding process,the first step is to locate the features which have been marked. The process of optimization through GA is not required during this phase. We use a watermark decoder ζ, which calculates the amount of change in the value of a feature that does not affect its data quality. The watermark decoder decodes the watermark by working with one bit at a time. In the decoding phase, η_{d_r} is calculated using equation 9 and represents the percent change detected in the watermarked data. The value of η_{d_r}, η_r and η_{Δ_r} is calculated using the values of tuple r and therefore might be different for every r. The parameter η_{Δ_r} is computed by calculating the difference between the original data change amount η_r and the watermark detected change amount η_{d_r} using equation 10.

$$\eta_{d_r} = D^{'}_W * \zeta \tag{9}$$

$$\eta_{\Delta_r} = \eta_{d_r} - \eta_r \tag{10}$$

The watermark decoding algorithm is presented in Algorithm 2. The decoding phase mainly consists of two steps: Step 1. For every candidate feature A of all the tuples in $D^{'}_W$, the watermark bits are detected starting from the LSB (least significant bit) and moving towards the MSB (most significant bit). The bits are detected in the reverse order compared with the bits encoding order because it is easy to detect the effect of the last encoded bit of the watermark. This process is carried out using the change matrix η_r. Step 2. The bits are then decoded according to the percentage change values of watermarked data. If $\eta_{\Delta_r} \leq 0$, the detected watermark bit will be 1. If $\eta_{\Delta_r} > 0$ and $\eta_{\Delta_r} \leq 1$, the detected watermark bit will be 0. The final watermark information is retrieved through a majority voting scheme using equation 11.

$$W_D \Leftarrow mode(dtW(1, 2, ..., l)) \tag{11}$$

Algorithm 2 Watermark Decoding

Input: D_W or $D^{'}_W$, ∇, l
Output: W_D
  for $r = 1$ to R **do**
    //loop will iterate for all tuples of the data
    for $b = l$ to 1 **do**
      //loop will iterate for all watermark bits b from 1 to length l of the watermark
      $\eta_{d_r} \Leftarrow D^{'}_{W(r)} * \zeta$
      $\eta_{\Delta_r} \Leftarrow \eta_{d_r} - \eta_r$
      if $\eta_{\Delta_r} \leq 0$ **then**
        detected watermark bit (dtW) is 1
      else if $\eta_{\Delta_r} > 0$ and $\eta_{\Delta_r} \leq 1$ **then**
        detected watermark bit (dtW) is 0
      end if
    end for
  end for
  $W_D \Leftarrow mode(dtW(1, 2, ..., l))$
  return W_D

3.4 Data Recovery Phase

After detecting the watermark string, some post processing steps are carried out for error correction and data recovery. The optimized value of β computed through the GA is used for regeneration of original data. The data recovery algorithm is presented in algorithm 3. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

value of a numeric feature is recovered using equations 12 and 13.

$$D_r = D'_{W_r} + \beta \quad (12)$$

$$D_r = D'_{W_r} - \beta \quad (13)$$

Algorithm 3 Data Recovery

Input: D_W or D'_W, b
Output: D_r
  for $r = 1$ to R **do**
    //loop will iterate for all tuples of the data
    for $b = l$ to 1 **do**
      //loop will iterate for all watermark bits b from 1 to length l of the watermark
      if $dtW(r, b) == 1$ **then**
        // 0 or 1 watermark bit is detected from every tuple r
        data is recovered by using equation 12
      else
        data is recovered by using equation 13
      end if
    end for
  end for
  return D_r

4 Results and Discussion

Experiments are conducted on Intel Core i3 with CPU of 2.40GHz and RAM of 2GB. For brevity, heart disease medical dataset, [35] containing more than 300 tuples is selected. RRW was evaluated for: (1) investigating effect on the data quality of the underlying data; (2) robustness against malicious attacks; and (3) restoration of the original data. The data recovery, watermark detection accuracy and effect of RRW on data quality are evaluated using the case study of a heart disease medical dataset. A small set of tuples from the same dataset are also used as an example to illustrate the entire procedure step by step.

Robustness of RRW is demonstrated through an extensive attack analysis. Our results have shown 100% accuracy in both watermark detection and data recovery. The experiments performed, demonstrate data recovery in best case as well as in worst case scenarios where Mallory tries to insert, alter and delete 10%, 20%, 30%, 40%, 50% and up-to 100% of the data and the results are plotted in the graphs below. After such attacks, RRW recovered 100%, 65.02% and 50.50% tuples with 50% insertion, alteration and deletion attacks on data respectively. In another experiment, 37% of tuples have been recovered with 100% distortion in data. The effect of RRW on the statistical measures reported in the GA fitness function as in equation 4 is analyzed with the original as well as watermarked data. Results are compared with effect on the statistical measures of the PEEW technique before and after watermarking. RRW is also compared with DEW, GADEW and PEEW techniques for watermark detection accuracy with subset insertion, subset alteration and subset deletion attacks. In all these scenarios, RRW technique has shown better results.

The computational time of RRW is $(l * R * A)$ where l is the watermark length, R is the total number of tuples in the dataset and A is the feature selected for watermarking. The number of tuples are usually much larger as compared to the number of features in databases and the watermark length l; so, $A << R$ and $l << R$. Therefore, for large databases, (R termed as $'n'$) the time complexity of RRW for watermark insertion and detection is $O(n)$ (it is worth mentioning here that the time for computing the GA based optimal value is not included in this calculation because it is part of pre-processing and same is the case for GADEW). For datasets involving large number of features or large number of tuples, the data owner may use a separate machine, with high computation power, for watermarking the datasets. This is a small price to pay for an improved sense of privacy against data theft (false claim of ownership can be tackled by watermark encoding and decoding). Computational time of PEEW is also $O(n)$ as this technique also needs to process all the tuples. DEW marks less number of tuples based on whether they meet a certain threshold; however, it still needs to process all tuples leading to a time complexity of $O(n)$. For GADEW, the time complexity is $(C*G*R*A)$ where C is the number of chromosomes, G is the number of generations involved in calculating the total cost, R is the number of tuples and A is the number of features. In general this is again $O(n)$.

4.1 A Case Study of Medical Data Recovery

4.1.1 Phase 1: Preprocessing

Consider the medical dataset of heart disease classification problem; for brevity, here only two features (A1 and A2) and one feature (A14 - Class Labels) with four tuples are shown in Table 2. The class label is divided into five classes, where 0 corresponds to absence of heart disease and 1,2,3,4 corresponds to four different types of heart diseases on the basis of cp (chest pain) type naming: typical angina, atypical angina, non-anginal pain, asymptomatic respectively. In order to select the candidate feature for watermarking, mutual information of the whole dataset is calculated. The MI_O and MI_W of the watermarked feature $A1$ has been provided in the Table 3 ($A1$ was selected as the candidate feature for watermarking after calculating the *MI* of all features). Its statistical relationship (*MI*) with itself has not been shown in the table.

TABLE 2
D (Original Data) for watermarking

A1	A2	...	Class Labels
63.0	M	...	1
67.0	F	...	0
37.0	M	...	3
41.0	M	...	4

An optimum value of β = 0.29 is calculated by using an optimization scheme. This value might be different for different databases. We have performed sufficient number of experiments to find the most reliable set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
Mutual Information of Selected Feature before and after watermarking

Sr no	Name of features	MI_O	MI_W	Δ_{MI}
A1	Age	-	-	-
A2	Sex	0.1175	0.1175	0
A3	cp	0.3919	0.3919	0
A4	trestbps	1.9705	1.9705	0
A5	chol	3.8996	3.8996	0
A6	fbs	0.1113	0.1113	0
A7	restecg	0.1763	0.1763	0
A8	thalach	3.0698	3.0698	0
A9	exang	0.1396	0.1396	0
A10	oldpeak	0.4373	0.4373	0
A11	slope	0.1925	0.1925	0
A12	ca	0.4259	0.4259	0
A13	thal	0.2191	0.2191	0
A14	Class Labels	0.3631	0.3631	0

of parameters for the genetic algorithm. The detailed set of GA parameters found reliable are given in table 4. The preprocessing phase, takes approximately 1.7 milliseconds for computing an optimum value and a watermark string for the entire medical dataset [35]. Figure 3 demonstrates the time incurred in this phase for increasing dataset sizes (number of tuples) where l was kept fixed at 16-bits and the number of features were 14.

TABLE 4
Genetic Algorithm parameters

GA Parameter	Value
No. of Generations	100
Population Size	50
Chromosome Length	3
Selection Mechanism	Tournament Selection Tournament Size = 5
Crossover	Type: Single Point Fraction: 0.7
Mutation	Type: Uniform Rate: 0.1
Elitism Count	2

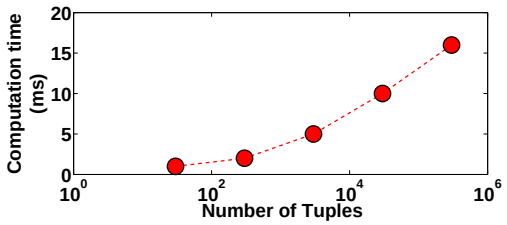


Fig. 3. Computation Time for Genetic Algorithm

4.1.2 Phase 2: Watermark Encoding

We report an example where the owner inserts a 3-bits long watermark 110 into the dataset and ζ = 10% change is incorporated into the tuple values. We use this high value of ζ just to make the difference between the original and the marked data clear for the readers of this paper; however, in real practice, relatively smaller values of ζ are used. The values of η_r to be embedded into the values D_r of feature $A1$ are calculated using equation 6. The watermark encoding algorithm starts the embedding process with the most significant bit (MSB) of the watermark. For this purpose, the algorithm works with one tuple at a time. Suppose the MSB of the watermark is 1; therefore, the new value of D_r (denoted by D_{W_r}) will be calculated using equation 7. In order to embed the second MSB of the watermark, the algorithm is again run using the same procedure, but the updated value D_r of the feature $A1$ is used for calculating new values of η_r and D_{W_r}. If the algorithm encounters a bit of the watermark that is 0 then the new value D_{W_r} of $A1$ will be calculated using equation 8. The above procedure is repeated until all tuples of the dataset are watermarked. Four tuples from the original data are shown in table 2 for brevity. The watermark encoding process is explained in tables 5 - 7. These three tables have represented the manipulation of a 3-bits long watermark string (110), on all four values of feature $A1$, one-by-one.

TABLE 5
Watermark Encoding with Watermark Bits 110, Started from First Bit (1)

A1	Class Labels	η_r
62.71	1	6.3
66.71	0	6.7
36.71	3	3.7
40.71	4	4.1

TABLE 6
Data Encoded with 2nd Bit (1)

A1	Class Labels	η_r
62.42	1	6.271
66.42	0	6.671
36.42	3	3.671
40.42	4	4.071

TABLE 7
Data Encoded with the 3rd Bit (0) (D_W - Watermarked Data)

A1	Class Labels	η_r
62.71	1	6.242
66.71	0	6.642
36.71	3	3.642
40.71	4	4.042

4.1.3 Phase 3: Watermark Decoding

In the decoding phase - $A1$ - the watermarked feature has been selected after calculating *MI*. The embedded bits are decoded in reverse order in which they are encoded. The values of η_r are used, that are saved by the data owner during the watermark encoding stage in a matrix, to calculate the value of η_{d_r} for D_{W_r} by calculating equation 9. The percentage change in data of 0.1 is used both for encoding and decoding. The value of η_r is used to compute its difference from η_{d_r} using equation 10.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

The value of η_{Δ_r} is compared with the value of watermark decoding operator (having values between 0 and 1) to decode the last embedded watermark bit. After decoding the last embedded bit for all the tuples, majority vote is taken and if the output of the majority voting process for that bit is 1 then 1 is saved as the detected bit and the values of feature $A1$ in all the tuples is updated using equation 12. If the output of the majority voting is 0 then 0 is saved as the detected bit and the values of feature $A1$ in all the tuples is updated using equation 13.

Watermark decoding is performed in the reverse order of the watermark embedding process; the last embedded bit is decoded first and so on. Therefore, in the example of watermark decoding the watermark bits are decoded after applying the majority voting scheme on every tuple of the dataset. In our case these are 110, which is the same as the embedded watermark bits. The decoding process is explained in tables 8 - 10. These three tables have shown the detection of a 3-bits long watermark string (110), from all four values of feature $A1$, one-by-one.

TABLE 8
Majority Bit = 0, Detected Watermark Bit = 0

A1	Class Labels	η_{d_r}	η_r	η_{Δ_r}	Detected Bits
62.42	1	6.271	6.242	0.029	0
66.42	0	6.671	6.642	0.029	0
36.42	3	3.671	3.642	0.029	0
40.42	4	4.071	4.042	0.029	0

TABLE 9
Majority Bit = 1, Detected Watermark Bit = 1

A1	Class Labels	η_{d_r}	η_r	η_{Δ_r}	Detected Bits
62.71	1	6.242	6.271	-0.029	1
66.71	0	6.642	6.671	-0.029	1
36.71	3	3.642	3.671	-0.029	1
40.71	4	4.042	4.071	-0.029	1

TABLE 10
Majority Bit = 1, Detected Watermark Bit = 1

A1	Class Labels	η_{d_r}	η_r	η_{Δ_r}	Detected Bits
63	1	6.271	6.3	-0.029	1
67	0	6.671	6.7	-0.029	1
37	3	2.671	3.7	-0.029	1
41	4	4.071	4.1	-0.029	1

4.1.4 Phase 4: Data Recovery

The original data is recovered along with watermark bits from watermarked data $A1_w$. The recovered data is the same as shown in table 2.

4.2 Robustness Analysis

In this experimental study, the watermarked data is subjected to different types of attacks. Suppose, Alice is the owner of the data and she embeds a watermark W in the tuples which results in D_W with the intent of ownership protection. Bob, the recipient, requires that the data quality should remain intact for information extraction process. Alice is now willing to share watermarked data D_W with Bob. In the meantime, Mallory (the intruder) wants to corrupt the watermarked data from the dataset so that Bob is unable to detect the watermark and use the data for knowledge extraction process.

In this study, the following assumptions are considered: (1) Mallory is unable to access original data; and (2) Mallory does not have access to secret parameters like candidate feature of the dataset, percentage data change ζ, watermark length l, and the watermark encoder/decoder β. These assumptions will make the task of Mallory very difficult and she has to corrupt the watermark in such a manner that the data quality remains intact. Therefore, in a worst case scenario, even if she is able to successfully corrupt the original data, the proposed scheme is robust enough for its successful recovery. The watermark is inserted in all the tuples of the selected feature in the dataset, thus the watermark (and hence the original data) can be recovered from the remaining tuples if Mallory is somehow able to successfully corrupt the watermark in some tuples.

In the robustness study, experiments were performed with three types of attacks: (1) Insertion; (2) Deletion; and (3) Alteration. The original data recovery is reported for RRW only and watermark detection rate is compared with DEW [23], PEEW [28], and GADEW [27] techniques with different fractions of tuples. RRW is highly robust as compared to PEEW, GADEW and DEW techniques in the analysis of the three types of above mentioned attacks. In RRW all the tuples of the selected feature were watermarked to achieve robustness. GADEW watermarks a large number of tuples in two particular features as compared to DEW and thus improves robustness in scenarios of attribute-wise and tuple-wise multi-faceted attacks. However, the dependency of robustness of GADEW on AWD and TWD requires a trade-off between the watermark robustness and the data distortion introduced during watermark embedding. As a result, the decoding accuracy of GADEW degrades if data distortions are minimized. In particular, its performance is not satisfactory in case of heavy attacks because watermarked tuples are affected by such attacks; consequently, it becomes difficult to decode the watermark correctly from the remaining un-affected tuples. On the contrary, RRW watermarks all the tuples of the selected feature and still provides a way of controlling data distortions while ensuring data quality after watermark embedding.

4.2.1 Insertion Attacks

In this particular type of attack, the insertion of new tuples by Mallory did not damage the data quality and watermark information, because, she is not disturbing the original tuples. Mallory may insert a number of α duplicate tuples or randomly generated fake tuples into

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

the database. RRW is resilient against these types of attacks.

4.2.1.1 Data Recovery: When Mallory tries to insert 50% tuples within the range of values of the watermarked feature, 100% data is recovered as shown in figure 4. The reason for successful data recovery is the marking of all the tuples (majority vote) and use of η_r.

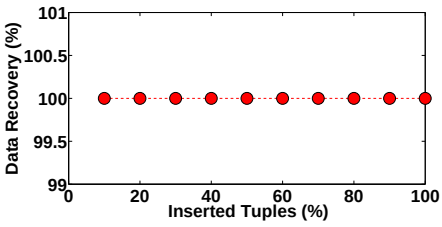


Fig. 4. Data Recovery with RRW after Insertion Attacks

4.2.1.2 Watermark Detection: RRW is compared with the most recent techniques namely DEW, GADEW and PEEW for analyzing its watermark decoding accuracy. Watermark detection accuracy is 100% in RRW, 89% to 94% in DEW, 100% GADEW and 98% to 100% in PEEW technique as shown in figure 5.

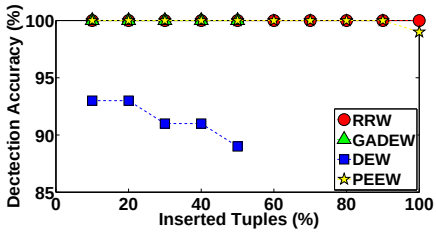


Fig. 5. A Comparison of Watermark Decoding Accuracy of RRW with DEW, GADEW and PEEW after Insertion Attacks

4.2.2 Deletion Attacks

In this type of attack, Mallory might delete α subset of tuples from the watermarked database. In the decoding phase, the watermark information and original data is recovered from the rest of the data. If Mallory deletes α tuples from the dataset and mutual information of the features in the database is changed, the data quality of the features gets compromised and as a result the knowledge extraction process makes wrong decisions. Since, the attacker wants to disturb the data usefulness, α tuples will be deleted from the database such that the data quality is unaffected. Consequently, the ranking of features is not disturbed after such attacks and the data remains useful.

4.2.2.1 Data Recovery: The original data is recovered with more than 50% accuracy in case 50% data was deleted as shown in figure 6. The reason for the success of the proposed scheme is the ability of being able to embed a low distortion watermark in all the tuples and applying a majority voting scheme.

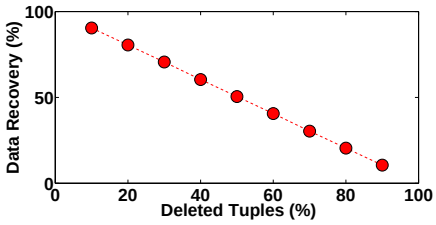


Fig. 6. Data Recovery with RRW after Deletion Attacks

4.2.2.2 Watermark Detection: Our experiments showed that when more than 80% of the data was deleted, the watermark was detected with 100% accuracy. We compared RRW with well known reversible watermarking techniques for detecting the watermark information after such attacks. RRW has shown 100% accuracy when up to 90% tuples were deleted while DEW, GADEW and PEEW were less accurate when a larger number of tuples were attacked. The results of this study are reported in figure 7.

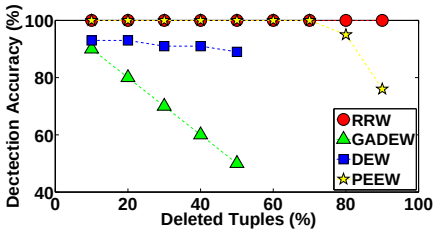


Fig. 7. A Comparison of RRW against DEW, GADEW and PEEW for Resilience to Deletion Attacks

4.2.3 Alteration Attacks

In such attacks, Mallory can modify the value of the watermarked feature within a certain range. Mallory can make random or fixed alterations within the range of minimum $min(a)$ to maximum values $max(a)$. When Mallory alters α tuples, the watermark decoder helps the decoding process to successfully recover the original data and the watermark from unaltered tuples and some of the altered tuples if data usability is not affected after the attacks.

4.2.3.1 Data Recovery: RRW demonstrated more than 65% of data recovery when half of the tuples get altered (see figure 8).

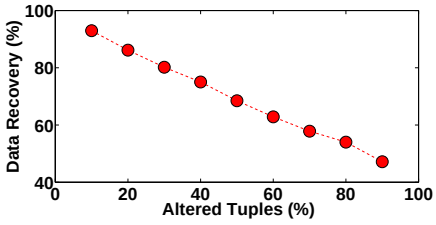


Fig. 8. Data Recovery with RRW after Alteration Attacks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

4.2.3.2 Watermark Detection: Again, in the presence of such attacks, RRW provided 100% accuracy with 90% of attacked tuples whereas DEW, GADEW and PEEW gave less accuracy as shown in figure 9.

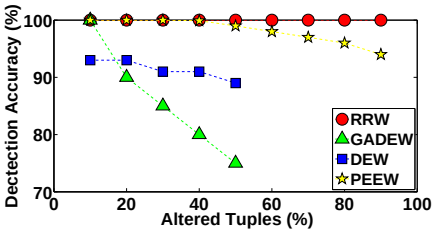


Fig. 9. Watermark decoding accuracy with DEW, GADEW and PEEW after Alteration Attacks

To summarize, a comparative study of RRW with other state of the art techniques is given in table 11. It is observed that RRW provides 100% watermark detection even with 90% attacked tuples while other techniques do not give such good results. As far as the data recovery is concerned, even in the presence of heavy attacks most of the data was recovered with high degree of accuracy.

4.3 Effect on Statistical and Classification Measures

The effect of RRW on data quality is evaluated with usability constraints defined in the fitness function in equation 3. For this purpose, the calculation of mutual information of the original data MI_O and the watermarked data MI_W is carried out and is shown in table 3. The proposed reversible watermarking (RRW) technique is also compared with PEEW technique by computing other statistical measures constrained in fitness function such as mean and variance (see equation 4). Our reported mean and variance of the original selected feature is 54.439 and 81.697 and, the mean and variance of the watermarked feature is 55.019 and 81.697 respectively. The reported value of variance is the same before and after watermarking the data, while we can notice a difference of 0.58 in mean value which is negligible as compared to that of PEEW [28] (bold in the table). The values for mean and variance of some features taken from datasets used by RRW and PEEW before or after applying watermarking techniques are given in table 12.

Moreover, the effect of RRW on different classification results of the data, before and after watermarking the dataset is evaluated through a data mining application. Various classifiers are used for this purpose; such as, Support Vector Machine (SVM) with linear and RBF kernel functions [39], K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), and Naive Bayes classifier to classify the heart disease data [35]. It is found that the classification measures - including accuracy, sensitivity and specificity - remain almost same before and after applying RRW. The data is classified into 5 classes on the basis of cardiac symptoms where 0 corresponds to absence of heart disease and 1,2,3,4 correspond to four different types of heart diseases on the basis of the type of chest pain including: typical angina, atypical angina, non-anginal pain, asymptomatic respectively. The classification results of the data obtained through various classifiers before and after applying RRW are reported in table 13.

5 Conclusions

Irreversible watermarking techniques make changes in the data to such an extent that data quality gets compromised. Reversible watermarking techniques are used to cater to such scenarios because they are able to recover original data from watermarked data and ensure data quality to some extent. However, these techniques are not robust against malicious attacks – particularly those techniques that target some selected tuples for watermarking. In this paper, a novel robust and reversible technique for watermarking numerical data of relational databases is presented. The main contribution of this work is that it allows recovery of a large portion of the data even after being subjected to malicious attacks. RRW is also evaluated through attack analysis where the watermark is detected with maximum decoding accuracy in different scenarios. A number of experiments have been conducted with different number of tuples attacked. The results of the experimental study show that, even if an intruder deletes, adds or alters up to 50% of tuples, RRW is able to recover both the embedded watermark and the original data. RRW is compared with recently proposed state-of-the-art techniques such as DEW, GADEW and PEEW to demonstrate that RRW outperforms all of them on different performance merits.

One of our future concerns is to watermark shared databases in distributed environments where different members share their data in various proportions. We also plan to extend RRW for non-numeric data stores.

References

[1] Y.-C. Liu, Y.-T. Ma, H.-S. Zhang, D.-Y. Li, and G.-S. Chen, "A method for trust management in cloud computing: Data coloring by cloud watermarking," *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 280–285, 2011.

[2] "Walmart to start sharing its sales data," http://www.nypost.com/p/news/business/walmart-opens-up, last updated: 09:55 AM on February 4, 2012, last accessed: July, 20 2013.

[3] "Identity theft watch," http://www.scambook.com/blog/2013/04/identity-theft-watch-customer-passwords-stolen-from-walmart-vudu-video-service/, last updated: April 11, 2013, last accessed: July, 20 2013.

[4] "Securing outsourced consumer data," http://www.databreaches.net/securing-outsourced-consumer-data/, last updated: February 26, 2013, last accessed: July, 20 2013.

[5] "As patients' records go digital, theft and hacking problems grow," http://www.kaiserhealthnews.org/Stories/2012/June/04/electronic-health-records-theft-hacking.aspx, last updated: June 03, 2012, last accessed: July, 20 2013.

[6] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1673–1687, 1997.

[7] I. Cox, M. Miller, J. Bloom, and M. Miller, *Digital watermarking*. Morgan Kaufmann, 2001.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

TABLE 11
Robustness Analysis of RRW with GADEW, DEW, PEEW

	RRW	**GADEW**	**DEW**	**PEEW**
Data distortions	Negligible	Less	Less	Less
Resilience against insertion attacks	100%	89% to 94%	100%	98% to 100%
Resilience against alteration attacks	100%	74% to 100%	89% to 94%	95% to 100%
Resilience against deletion attacks	100%	50% to 90%	89% to 94%	76% to 100%
Watermark detection with or without attacks	100%	89%	94%	95%
Original data recovery without attacks	100%	89%	94%	98%
Original data recovery with 50% attacks	65%	37%	40%	50%

TABLE 12
Effect of RRW and PEEW on Various Statistical Measures (Mean μ_D, Variance σ_D of RRW, and Mean μ_d, Variance σ_d of PEEW)

Features	μ_D	σ_D	μ_{D_W}	σ_{D_W}	μ_d	σ_d	μ_{d_w}	σ_{d_w}
A1	54.4389	81.6974	55.019	81.697	134.92	28787.73	134.91	28787.89
A2	0.6799	0.2184	0.6799	0.2184	299.64	**141878.7**	299.6426	**141879.09**
A3	3.1584	0.9218	3.1584	0.9218	2.54	**0.0312**	2.5442	**0.03294**
A4	131.6898	309.7511	131.6898	309.75	54.3	**478.2**	54.3024	**478.339**
A5	246.6931	2.68e+03	246.6931	2.6e+03	156.92	31006.42	156.92244	31006.526
A6	0.1485	0.1269	0.1485	0.1269	**35.49**	**204.2**	**35.4297**	**204.326**
A7	0.9901	0.9900	0.9901	0.9900	4.54	**0.030**	4.54305	**3.030882**
A8	149.6073	523.2658	149.6073	523.2658	38.9	1201.8	38.920716	1201.8555

TABLE 13
Effect of RRW on Various Classification Measures ((Accuracy Acc, Sensitivity Sn and Specificity Sp)

$Classifier$	$Classes$	Acc_O	Acc_W	Sn_O	Sn_W	Sp_O	Sp_W
Linear SVM	0	83.44	83.82	88.12	86.87	77.86	78.96
	1	81.84	81.85	0	0	100	100
	2	88.24	88.24	0	0	100	100
	3	88.56	88.26	2.50	5.83	100	100
	4	95.64	95.64	0	0	100	100
RBF SVM	0	85.17	85.23	90	90.62	79.56	79.01
	1	81.83	81.83	0	0	100	100
	2	88.23	88.25	0	0	100	100
	3	88.24	88.23	0	0	100	100
	4	95.64	95.65	0	0	100	100
NB	0	84.22	83.23	88.13	86.88	78.68	78.52
	1	57.81	57.93	53.33	52.33	60.10	60.83
	2	73.46	73.04	82.50	81.67	72.45	72.14
	3	76.65	75.73	89.17	82.50	74.78	74.09
	4	77.49	76.39	80	85	76.81	76.02
KNN	0	57.21	59.97	64.37	66.87	48.79	51.87
	1	69.69	68.31	9.66	8.67	83.05	81.45
	2	78.40	78.53	16.66	19.17	86.5	86.28
	3	83.85	83.23	18.33	18.33	92.71	93.13
	4	91.92	92.92	15	10	95.42	95.42
LDA	0	83.76	83.76	86.25	86.87	79.67	80.33
	1	59.96	60.59	59.00	57.33	62.93	62.63
	2	71.03	71.08	78.33	64.16	71.42	72.14
	3	76.51	75.73	60.83	61.66	75.97	76.30
	4	80.84	81.85	50	50	81.70	81.99

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2014.2349911, IEEE Transactions on Knowledge and Data Engineering

[8] P. W. Wong, "A public key watermark for image verification and authentication," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1. IEEE, 1998, pp. 455–459.

[9] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1593–1601, 2001.

[10] F. A. Petitcolas, "Watermarking schemes evaluation," *Signal Processing Magazine, IEEE*, vol. 17, no. 5, pp. 58–64, 2000.

[11] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, 1999.

[12] R. Agrawal and J. Kiernan, "Watermarking relational databases," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 155–166.

[13] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for categorical data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 7, pp. 912–926, 2005.

[14] S. Subramanya and B. K. Yi, "Digital rights management," *Potentials, IEEE*, vol. 25, no. 2, pp. 31–34, 2006.

[15] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[16] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method for constrained optimization problems," *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, vol. 76, pp. 214–220, 2002.

[17] R. Hassan, B. Cohanim, O. De Weck, and G. Venter, "A Comparison Of Particle Swarm Optimization And The Genetic Algorithm," in *46 th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2005, pp. 1–13.

[18] Y.-R. Wang, W.-H. Lin, and L. Yang, "An intelligent watermarking method based on particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8024 – 8029, 2011.

[19] M. Kamran and M. Farooq, "An information-preserving watermarking scheme for right protection of emr systems," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 11, pp. 1950–1962, 2012.

[20] T. M. Cover, J. A. Thomas, and J. Kieffer, "Elements of information theory," *SIAM Review*, vol. 36, no. 3, pp. 509–510, 1994.

[21] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley-interscience, 2012.

[22] Y. Zhang, B. Yang, and X.-M. Niu, "Reversible watermarking for relational database authentication," *Journal of Computers*, vol. 17, no. 2, pp. 59–66, 2006.

[23] G. Gupta and J. Pieprzyk, "Reversible and blind database watermarking using difference expansion," in *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 24.

[24] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1. IEEE, 2003, pp. I–501.

[25] G. Gupta and J. Pieprzyk, "Database relation watermarking resilient against secondary watermarking attacks," in *Information Systems Security*. Springer, 2009, pp. 222–236.

[26] J.-N. Chang and H.-C. Wu, "Reversible fragile database watermarking technology using difference expansion based on svr prediction," in *Computer, Consumer and Control (IS3C), 2012 International Symposium on*. IEEE, 2012, pp. 690–693.

[27] K. Jawad and A. Khan, "Genetic algorithm and difference expansion based reversible watermarking for relational databases," *Journal of Systems and Software*, 2013.

[28] M. E. Farfoura and S.-J. Horng, "A novel blind reversible method for watermarking relational databases," in *Parallel and Distributed Processing with Applications (ISPA), 2010 International Symposium on*. IEEE, 2010, pp. 563–569.

[29] D. M. Thodi and J. J. Rodriguez, "Prediction-error based reversible watermarking," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 3. IEEE, 2004, pp. 1549–1552.

[30] D. M. Thodi and J. J. Rodriguez, "Reversible watermarking by prediction-error expansion," in *Image Analysis and Interpretation, 2004. 6th IEEE Southwest Symposium on*. IEEE, 2004, pp. 21–25.

[31] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *Image Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 721–730, 2007.

[32] M. E. Farfoura, S.-J. Horng, J.-L. Lai, R.-S. Run, R.-J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3185–3196, 2012.

[33] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3524–3533, 2011.

[34] E. Sonnleitner, "A robust watermarking approach for large databases," in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*. IEEE, 2012, pp. 1–6.

[35] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan, "Biased minimax probability machine for medical diagnosis," in *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics (AIM04)*, 2004.

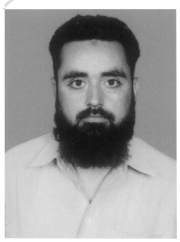
[36] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[37] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 108–109.

[38] M. Mitchell, "An introduction to genetic algorithms mit press," *Cambridge, Massachusetts. London, England*, 1996.

[39] S. Bhatia, P. Prakash, and G. Pillai, "Svm based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features," in *Proceedings of the World Congress on Engineering and Computer Science, WCECS*, 2008, pp. 22–24.

Saman Iftikhar received her M.S degree in Information Technology in 2008 from National University of Sciences and Technology (NUST), Islamabad, Pakistan. She has worked as Research Assistant in Health Life Horizon (HLH) project funded by Information and Communication Technology Research and Development (ICT R&D) from 2008 to 2011 in NUST. Her research interests include information security, distributed computing, semantic web and cloud computing.



M. Kamran received the M.S. and Ph.D. degrees in computer science, in 2008 and 2012, respectively, from National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. Currently he is working as Assistant Professor in Computer Science Department of COMSATS Institute of Information Technology, Wah Cantt, Pakistan. His research interests include machine learning, evolutionary computations techniques, data security, health informatics, big data analytics and decision support systems.

Zahid Anwar received his Ph.D. and M.S. degrees in Computer Sciences in 2008 and 2005 respectively from the University of Illinois at Urbana-Champaign, USA. Zahid has worked as a software engineer and researcher at IBM, New York, USA, Intel, Oregon, USA, Motorola, Chicago, USA and the National Center for Supercomputing Applications (NCSA), Urbana, Illinois, USA on various projects related to information security and operating system design. Zahid holds post-doctorate experience from Concordia University, Montreal, Canada. Currently he is a faculty member at the School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.