A Robust Multiple Watermarking Technique for Information Recovery

Shampa Chakraverty[1], Om Prakash Verma[2], Vidhi Khanduja[1], Rakshita Tandon[1], Sahil Goel[1]

1 Department of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India
2 Department of Information Technology, Delhi Technological University, Delhi, India.
apmahs.nsit@gmail.com, opverma@dce.ac.in,vidhikhanduja9@gmail.com, raktandon@nsitonline.in, sahilnsit10@gmail.com

***Abstract*— Digital databases serve as the vehicles for compiling, disseminating and utilizing all forms of information that are pivotal for societal development. A major challenge that needs to be tackled is to recover crucial information that may be lost due to malicious attacks on database integrity. In the domain of digital watermarking, past research has focused on robust watermarking for establishing database ownership and fragile watermarking for tamper detection. In this paper, we propose a new technique for multiple watermarking of relational databases that provides a unified solution to two major security concerns; ownership identification and information recovery. In order to resolve ownership conflicts a secure watermark is embedded using a secret key known only to the database owner. Another watermark encapsulates granular information on user-specified crucial attributes in a manner such that the perturbed or lost data can be regenerated conveniently later. Theoretical analysis shows that the probability of successful regeneration of tampered/lost data improves dramatically as we increase the number of candidate attributes for embedding the watermark. We experimentally verify that the proposed technique is robust enough to extract the watermark accurately even after 100% tuple addition or alteration and after 98% tuple deletion.**

***Index Terms*— Data Recovery, Digital Watermarking, Right Protection, Robustness, Tamper Detection**

I. INTRODUCTION

With Information and Communication Technology (ICT) growing in leaps and bounds, the availability of information has taken centre-stage in the progress of mankind. A major proportion of the internet content is dynamically generated from databases. This has driven the capabilities, sizes and performance of databases to grow in exponential magnitudes. In this scenario where end users are demanding more and more information to be available on the net, data providers are burdened to supply accurate data and at the same time ensure its security. A major threat faced is that unauthorized and illicit copies of true data can be easily generated and distributed using the same enabling technologies. To counter such attacks, legal as well as technological solutions are being devised to assert data ownership.

Digital watermarking, a technique that was originally developed for establishing proof of ownership of digital data analogous to print watermarking, has come of age. Today, it is utilized not only for establishing proof of ownership, but for other objectives such as tamper detection, reversibility, data recovery and data provenance.

The basic principle of watermarking is to make imperceptible changes to the data in such a way that the data does not become unusable. Robust watermarks resist modifications, thereby serving as copyright indicators. Fragile watermarks become noticeable after the slightest modification. They serve to prove the fact that tampering has occurred in a suspect database.

Apart from protection from illicit copies and tampering, we also need to protect our database from potential data loss when it is compromised with either due to malicious attacks or due to noise introduced during network transfers. For many practical applications, it is not the raw data *per se*, but the information conveyed by it that is relevant and must be preserved at all cost. The information conveyed by all the tuples under an attribute is the degree of discernability in the values. If the attribute data is clustered with the desired level of granularity, the individual tuple values can be replaced by a representative value for the cluster it belongs. This motivates us to follow an information-centric data recovery using a watermarking technique. Unless the lost critical information is recovered, a compromised database may be rendered virtually useless.

Most existing watermarking schemes are designed to serve a single purpose such as robustness or tamper detection. We adopt a dual watermarking scheme that establishes ownership as well as allows data recovery so that there is no contention about who is eligible to recover the lost data. In our proposed scheme, the user-specified important attributes are partitioned into cohesive categories and their identifiers are used to prepare and insert the watermark in candidate attributes. In essence the salient information that is encapsulated in the data can be regenerated afterwards. The contributions of this paper are summarized below:

1. The proposed scheme regenerates crucial information encoded in the data in the event of both illegal alterations in the data as well as deletion of data.
2. The granularity of the recoverable information is decided beforehand by the user. We illustrate the use of unsupervised Machine Learning in discovering salient information contained in the data by using K-means clustering where the number of clusters is user-specified.

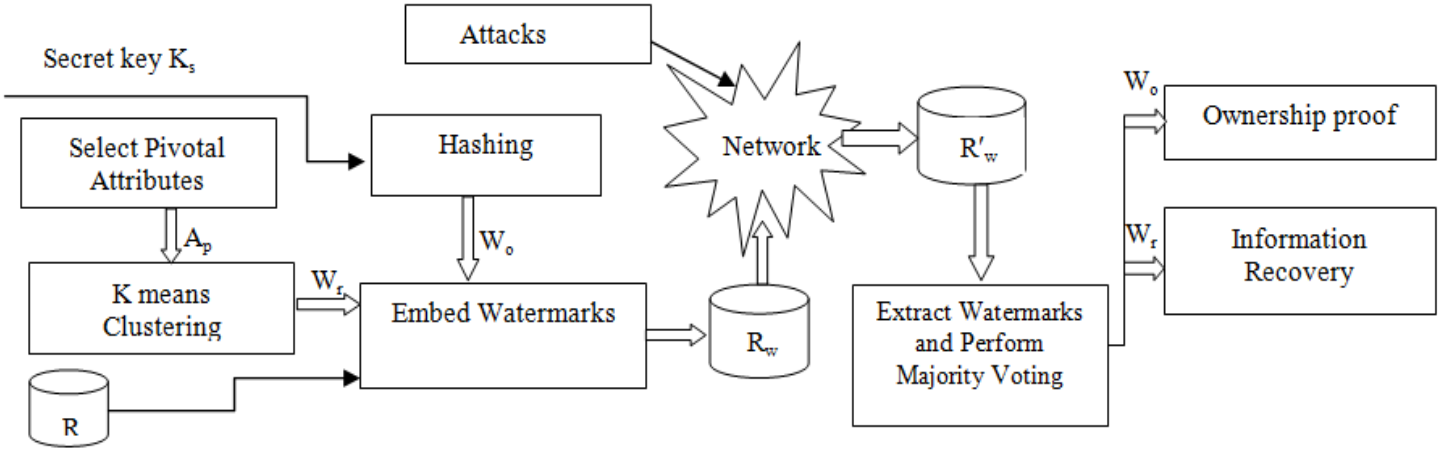

Figure 1. The Proposed Watermarking Model

3. Our scheme combines proof of ownership along with data recovery within the same watermarking framework by embedding multiple watermarks.

The rest of the paper is organized as follows: Section II gives an overview of the related work. Section III explains in detail our proposed watermarking scheme. Section IV presents the Attacker model followed by experiments performed on the database to check for robustness and category regeneration in section V. Finally section VI concludes the paper with a summary and suggestions for future work.

II. RELATED WORK

A body of research is centered around robust watermarking targeting proof of ownership [1-4]. Agrawal and Kiernan first proposed a simple and efficient robust technique for embedding a watermark securely in the Least Significant Bits (LSB) of numeric attributes [1]. Extending from here, Farfoura *et al.* proposed a blind and reversible watermarking technique for relational databases. Reversibility allows one to recover the original data completely from the watermarked database after authenticating with a time-stamp protocol [2]. Robust watermarking techniques on non-numeric and categorical attributes have been investigated in [3,4]. Sun *et al.* proposed multiple watermarking technique that uses two different images as copyright information to be embedded into the relational database [5]. This robust scheme targeted copyright protection only.

Another direction of research on watermarking has been towards fragile watermarking techniques that aim at proof of integrity. Li *et al.* proposed a distortion-less watermarking scheme for databases with categorical attributes that detects and localizes maliciously induced modifications [6]. Kamel *et al.* presented a fragile scheme with R-tree data structures [7]. Recently, Guo proposed a distortion-less watermarking in which tampered data can be localized up to the block level [8]. The technique works by manipulating the order and grouping information among database entities.

The utility of watermarking in recovering tampered data is illustrated in works reported on audio and image content recovery [9,10,11]. However, data recovery from compromised databases has not yet been fully explored. In [12], Khataeimaragheh *et.al* proposed a fragile watermarking technique that can detect and correct distortions in relational databases by embedding watermarks created from each attribute value, thereby recovering true data. There are certain serious shortcomings in such an approach. Firstly, it can only be used to detect and recover *altered* data and precludes proper recovery from deletions. Secondly, the probability of accurately detecting, localizing and hence rectifying errors reduces drastically when the number of errors exceeds two.

In sharp contrast, we follow an approach in which the endeavor is to safeguard the salient information that is encapsulated in important attributes of the database. The core information in a database is captured by applying an unsupervised learning algorithm such as *k means clustering* on the selected attributes [13]. The owner is free to dictate the granularity of this information by specifying the number of clusters. This approach deciphers data in terms of the information it represents and helps recover from altered as well as deleted data. Furthermore, our scheme is equipped with ownership proof.

III. PROPOSED SCHEME

Figure 1 depicts the flow of the proposed watermarking scheme. Information is framed in the form of a relational database R containing a primary key attribute K_p and a set of other attributes $\boldsymbol{A}$ where, $|\boldsymbol{A}|=\eta$ i.e. total number of attributes in R. We divide our watermarking model into following steps:

A. ***Select pivotal attributes for clustering:*** The owner first decides a set of pivotal attributes $\boldsymbol{A_p \subseteq A}$ which need to be information protected. $\boldsymbol{A_p}$ is partitioned into cohesive clusters by applying *k means clustering* on the dataset. The individual tuples of pivotal attributes are thus mapped to a smaller set of cluster-ids which are used as watermarks. The cluster-ids and the representative data points i.e. centroid of each cluster are saved with the owner or with a trusted entity. Each cluster represents a granule of information. The same information can be restored later when tampering is detected. In this manner, the usefulness of the database is preserved.

The onus of specifying the number of clusters rests on the owner. Thus she has complete control over the granularity of the information that is encoded into the watermark. The optimal choice of k will strike a balance between the degree of compression and the accuracy of recovered data.

Figure 2 shows cluster formation on the two selected pivotal attributes of the national geochemical survey (NGS) database we experimented on [14]. We chose the *pH-value* of the water content of the sample and grain size as the pivotal attributes. By applying *k means clustering* on these two attributes, we are

able to extract relevant information with three clusters. Even though we have shown the results for a two attributes, the same principles apply to any number of attributes.

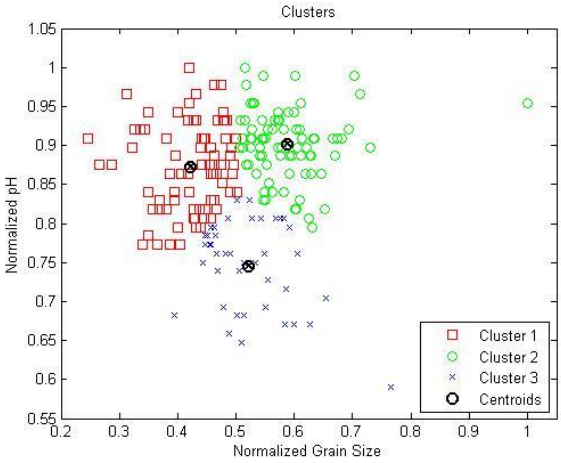

Figure 2. Formation of clusters by taking two pivotal attributes pH value and Grain size.

B. Embed Watermarks: The algorithm for embedding the watermarks is described in the pseudocode *EmbedWatermarks(.)* given in Figure 3. In this step we create two different watermarks, one for information recovery (W_r) and other for ownership proof (W_o).

Creation of W_r: The watermark $W_r(t)$ for recovering the information in the tuple t is the designated cluster-id of its pivotal attributes $\boldsymbol{A_p(t)}$. W_r is the combined watermark for all the tuples $W_r=W_r(1).W_r(2).W_r(t)...W_r(\eta_t)$, where η_t is total number of tuples in database.

We assume that the total number of categories for any attribute is less than 100. Therefore the cluster-ids are 2-digit integers. The watermark W_r(t) of each tuple t is hidden within some algorithmically selected secret positions of the other selected numeric candidate attributes of the same tuple repeatedly (Line 5 to 9).

Creation of W_o: The watermark embedded for ownership proof W_o is the hash of secret key which is known only to the owner of the database (Line 2). This, hash is embedded as watermark in several algorithmically selected secret positions of the database (Lines 6 and 7). The algorithm can use any of the available cryptographic hash functions such as MD4, MD5 and SHA-1 [16]. As discussed in [15], due to the secret key, the hash function's unique properties and multiple insertions this watermark is inherently robust against various attacks. Therefore it can be used to establish ownership claims and detect illicit copies.

Both kinds of watermarks are embedded separately into the fractional parts of specifically selected numeric attributes $\boldsymbol{A_w \subseteq A}$ where $\boldsymbol{A_w \cap A_p = \Phi}$. These candidate attributes are selected by the owner so that they can tolerate a certain amount of distortions caused by watermarking. The changes are made in the fractional part so as to minimize its impact. We assume a constant length α for the fractional part of the numeric attributes.

In order to determine the secret positions for embedding both watermarks, the cryptographic hash function is applied on the concatenation of the secret key and the tuple's primary key to obtain the result $H(t)$ (Line 2). This secure hash $H(t)$ decides (i) which of the attributes that are earmarked for embedding W_r and which ones are chosen for embedding W_o (Lines 5 to 9). Only that candidate attribute A_i which satisfies $mod(H(t), \beta)==i$, is selected for embedding W_o. All others are selected for embedding $W_r(t)$. To enhance the level of security, we store a single hex character of W_o in each tuple as determined by $p=mod(H(t), N)$, N being the length of the hash $H(t)$ and (ii) the secret positions within the fractional part where the watermark is to be embedded in each case by the function $mod(H(t), \alpha)$, where α is the length of the fractional part. (Lines 3,7 and 9). We need minimum of two members in set $\boldsymbol{A_w}$ to embed both W_o and W_r. However, we recommend $|\boldsymbol{A_w}|>=4$ i.e. cardinality of $\boldsymbol{A_w}$ be minimum four as one attribute will be utilized to embed W_o and rest three can be utilized to embed W_r thrice to allow triple redundancy based majority voting. As the number of candidate attributes increases, probability of successful information recovery also increases due to majority voting applied in extraction of watermark process.

EmbedWatermarks(.)
Input: Database: R, Candidate Attributes: $\boldsymbol{A_w}$, Recovery watermark of $\boldsymbol{A_w}$: W_r, Number of candidate attributes: β, Length of fractional part of numeric candidate attributes: α, Primary key attribute: K_p, Secret key: K_s, Type of hash function: *Hash(.)*, Length of *Hash*: N.
Output: Watermarked Database R_w

1. **For** every tuple $t \varepsilon R$, **repeat** steps 2-10
2. Calculate $H(t)=Hash(K_p(t)||K_s)$, $W_o=Hash(K_s)$
3. Calculate $x=mod(H(t), \alpha)$
4. **For** each candidate attribute $A_i \varepsilon \boldsymbol{A_w}$ repeat 5-10
5. **If** $(mod(H(t),\beta) = = i)$ perform steps 6 and 7
6. Calculate $p=mod(H(t),N)$
7. Embed hextodec($W_o(p)$) at the positions x and $x+1$ in the fractional part of $A_i(t)$
8. **Else**
9. Embed $W_r(t)$ at the positions x and $x+1$ in the fractional part of $A_i(t)$.
10. **End If**

Figure 3. Pseudo code for embedding watermarks in the Relational Database R.

C. Extract Watermarks: Figure 4 shows the pseudo-code that describes the watermark extraction process from the suspected watermarked database R_w'. Lines 1 to 7 are same steps as done in *Embedwatermarks(.)* function. For each tuple t, first their embedded watermarks are extracted. If $mod(H(t),\beta)== i$ then it is single hex character of W_o and stored in matrix M_p (Line 8). Otherwise it is $W_r(t)$ and stored in another matrix M which is cleaned up initially for each tuple (Line 5 and 10). Some of the $W_r(t)$ may be obliterated due to attacks. Hence a majority voting on all the values of M is performed (Line 12)

to get the correct cluster-id. Finally when all tuples have been processed, a majority voting on M_p for each hex character of W_o' outputs the correct character (Line 13). The embedded watermark W_o is then compared to the extracted watermark W_o'. If both the values are equivalent, then we can conclude that the suspected copy is a pirated one and hence, ownership can be claimed (Lines 14 to 16).

ExtractWatermarks(.)
Input: Suspected watermarked Database: R_w' , Candidate Attributes: $\boldsymbol{A}_w$, Number of candidate attributes: β, Length of fractional part of numeric candidate attributes: α, Primary key attribute K_p, Secret key K_s, Type of hash function: *Hash(.),* Length of *Hash*: *N*.
Output: Information recovery watermark W_r.

1. **For** every tuple $t \in R_w'$, **repeat** steps 2-13
2. Calculate $H(t)=Hash(K_p(t)||K_s)$,$W_o=Hash(K_s)$
3. Calculate $x=mod(H(t), \alpha)$.
4. **For** each candidate attribute $A_i \in \boldsymbol{A}_w$ repeat 4-12
5. Clear matrix *M*.
6. **If** $(mod(H(t),\beta) = = i)$ perform steps 7 and 8
7. Calculate $p=mod(H(t),N)$
8. Save the value present at the x^{th} and $x+1^{th}$ position in the fractional part of $A_i(t)$ in matrix M_p (where $p \in [1,N]$)
9. **Else**
10. Save the value present at the x^{th} and $x+1^{th}$ position in the fractional part of $A_i(t)$ in matrix *M*.
11. **End If**
12. Perform majority voting on *M* to obtain the cluster-id of tuple *t*.
13. Perform majority voting on each of the matrices M_p (where $p \in$ [1,N]) to obtain the decimal character at position *p*. Convert the decimal character into hexadecimal and save in $W_o'(p)$.
14. Compare W_o and W_o'
15. *If* found equivalent, *then* the suspected copy is a pirated one.
16. The cluster-ids of all the tuples are combined to form W_r.

Figure 4. Pseudo code for extracting watermarks from suspected watermarked database R_w'

IV. ATTACKER MODEL

Suppose Alice owns a database *R* which is freely available on the internet. Alice has embedded watermarks W_o and W_r into the database using our proposed algorithm. A malicious attacker Mallory may try to destroy these watermarks to claim the database to be hers and/or delete/alter important information from the database so as to reduce the usability of the database. In such an event, Alice can recover the watermark W_o from the database due to the majority voting mechanism applied in our algorithm and prove that the database was owned by her originally. She can then proceed to recover information.
To destroy either or both the watermarks W_o and W_r, Mallory can resort to following different kinds of attacks.

A. ***Attacks against ownership proof:*** In these attacks Mallory may try to destroy W_o. We give theoretical analysis to prove robustness against these attacks.

i. In *tuple addition attack*, Mallory tries to demolish the watermark W_o by adding spurious tuples in the database. In this way, she hopes to overshadow the watermark embedded in the database by adding new non-watermarked tuples. In our technique, the watermark bits have been embedded repeatedly into each tuple and then a majority voting mechanism has been adopted to extract the correct watermark back. This ensures that our watermark is safeguarded from this kind of an attack.

ii. In *tuple alteration attack*, Mallory tries to destroy the watermark W_o completely by randomly altering some data bits such that the data is not completely rendered useless. Multiple embedding of the watermark bits in each tuple along with the majority voting scheme ensure that our technique remains impervious to this attack.

iii. In *tuple deletion attack*, Mallory tries to delete tuples from the database with an intention of deleting all the watermarked tuples. In our technique, we have embedded watermark bits W_o in each tuple and hence to delete the entire watermark, she must destroy a significant portion of the database. This ensures that Mallory cannot erase the watermark completely without destroying a considerable portion of the database. Deleting such a huge part of the database will render the database useless.

B. ***Attack against Information Recovery:*** *M*allory may also try to delete or alter some pivotal data of the database not with an intention to delete the watermark but only to reduce the usability of the database. In our technique, we provide a scheme to regenerate information of lost/altered pivotal data. If Mallory deletes/alters any of the pivotal attribute from $\boldsymbol{A}_p$, then Alice can accurately regenerate the categories of each tuple by extracting the embedded information from other candidate attributes. In addition, Mallory may alter the attributes containing the embedded information. Out of total number of attributes η, the number of attributes containing the category information for each tuple will be β-1. We have applied majority voting, therefore, considering worst case scenario, the category information can be lost if at least $(\beta-1)/2+ 1 = (\beta+1)/2$ correct attributes are altered. Thus, the Mallory must accurately choose $(\beta+1)/2$ attributes and alter all of them. The number of ways of choosing $(\beta+1)/2$ attributes out of η attributes is given by (1).

$$ {}^{\eta}C_{(\beta+1)/2} \quad (1)$$

Where, C is combination of $(\beta+1)/2$ attributes out of η attributes. The total number of ways of correctly choosing the $(\beta+1)/2$ attributes is given by combination of $(\beta+1)/2$ attributes out of β-1 attributes. Hence, the probability of correctly choosing $(\beta+1)/2$ attributes is given by (2)

$$p_1 = \frac{{}^{\beta-1}C_{(\beta+1)/2}}{{}^{\eta}C_{(\beta+1)/2}} \quad (2)$$

For a given tuple, for each candidate attribute there are α possible positions where the bits are embedded. The probability of choosing the 2 correct consecutive locations for a single attribute of a tuple is given by $1/(\alpha-1)$. Hence the probability of altering $(\beta+1)/2$ attributes of a given tuple will be given by (3)

$$p_2 = \left(\frac{1}{\alpha-1}\right)^{\frac{\beta+1}{2}} \quad (3)$$

The probability of failure of regenerating the category of a particular tuple can therefore be stated as (4) using (2) and (3).

$$p = p_1 * p_2 = \frac{{}^{\beta-1}C_{(\beta+1)/2}}{{}^{\eta}C_{(\beta+1)/2}} \left(\frac{1}{\alpha-1}\right)^{(\beta+1)/2} \quad (4)$$

This probability can be reduced by choosing large values of β and α. Hence, for α=5 and β =40, η =60, p= 1.4954e-17 which is practically zero.

V. Experiments

The experiments were validated on an Intel Core™ i7 2.30 GHz system in MATLAB 7.8.0. The database [14] of a national geochemical analysis of stream sediments and soils conducted in the US has been used. The database contains information about soil samples at different geological locations. Pivotal attributes *pH*, *grain size*, *orgn_pct* (estimated % organics) are considered as elements of $\boldsymbol{A_p}$ and is being categorised into 16 categories (N_c =16). The candidate attributes $\boldsymbol{A_w}$ chosen are *Al_ICP40, Ca_ICP40, Fe_ICP40, K_ICP40, Na_ICP40, P_ICP40, Ti_ICP40, Ag_ICP40, As_ICP40, Mg_ICP40* which give the amount of aluminium, calcium, iron, potassium, sodium, phosphorous, titanium, silver, arsenic and magnesium respectively present in the soil samples. Our database contains 16,000 tuples with β=10 and α=4. We have used the SHA-1 algorithm thus N=40. In the proposed technique, the time taken to embed the watermark in each tuple is approximately 0.01 seconds.

A. Robustness Analysis: To ensure that our database is not stolen by any malicious third party, we need to make sure that the watermark embedded i.e. W_o in our database can withstand several attacks. The watermark must hence be robust. In the following section, we demonstrate the robustness of our technique against different attacks discussed above.

i. Tuple Addition: In this attack, Mallory tries to demolish the watermark W_o by adding θ% tuples in the database. We have performed the experiment by varying values of θ from 10 to 100. The results are recorded in Table I. The percentage watermark extracted accurately is 100 even when θ is made 100. Due to redundancy introduced and majority voting, embedded watermark is fully extracted even on 100% tuple addition. Hence, proposed technique is robust against this attack.

Table I
Resilience to Tuple addition and Alteration attack

θ	%W_o extracted Correctly	γ
10	100	10
40	100	40
80	100	80
90	100	90
100	100	100

ii. Tuple Alteration: In this attack, Mallory tries to demolish the watermark W_o by by altering γ % tuples in the database. We have performed the experiment by varying values of γ from 10 to 100. For altering the numeric attributes, we have randomly chosen any position from the fractional part of the number and replaced by a random digit. The results are recorded in the table I. The percentage watermark extracted accurately is 100 even for γ =100. The redundancy in embedding the watermark bits and majority voting scheme ensures 100% watermark recovery.

iii. Tuple Deletion: In this attack Mallory tries to delete the watermark W_o by deleting $\boldsymbol{\lambda}$% tuples from the database. We have performed the experiment by varying $\boldsymbol{\lambda}$ from 10 to 100. Experiments show that even on 98% deletion, we are able to extract the entire watermark. The database would be rendered useless if more than 98% of the tuples are deleted. Figure 5 shows the results of this experiment.

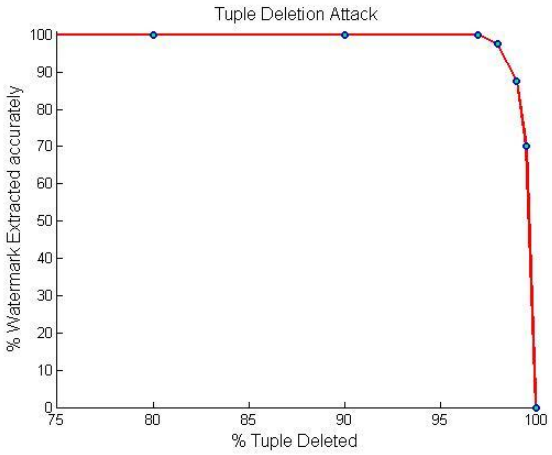

Figure 5. Tuple Deletion Attack

B. Information Recovery: If Mallory alters or deletes the values of any of pivotal attributes from $\boldsymbol{A_p}$ with an intention of reducing the usability of the database, then Alice can recover the correct category of each tuple with the help of embedded category information W_r. If Mallory tries to alter the embedded category information as well, even then most of the category information can be regenerated due to the redundancy introduced while embedding this information. Figure 6 shows a plot of percentage of correct information

extracted against percentage of tuples altered. This is observed by varying the number of perturbed candidate attributes per tuple (β=10).

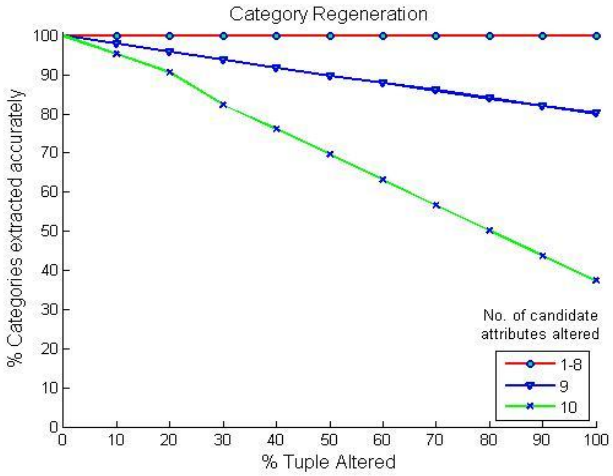

Figure 6. Category Regeneration

VI. CONCLUSION

In this paper, we have proposed a novel generic watermarking technique which not only solves the problem of ownership claims but regenerates information of pivotal attributes in the database. This is achieved by embedding multiple watermarks robustly. In the proposed technique information retrieval of crucial data is achieved through k-means clustering. Theoretical analysis show that the probability of failure of generating lost data from the database is reduced drastically as we increase the number of candidate attributes and the potential locations for embedding the watermark. Experiments demonstrate that our technique is robust to several attacks and even on 100% tuple addition or tuple alteration, the entire watermark is extracted. Even when 98% of the database is deleted, we are able to recover the watermark without any loss. Future work will focus on regenerating true data of the pivotal attributes for each tuple on alteration or deletion of data. We are working on extending this scheme for information systems that can embed the watermark in various data types attributes.

REFERENCES

[1]. R.Agrawal & J.Kiernan, "Watermarking relational databases", In proc. of the 28th Very Large DataBases conference (VLDB), 2002, Vol. 28, pp.155–166.

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

[3].V.Khanduja, A.Khandelwal, A.Madharaia, D.Saraf, T.Kumar, "A Robust Watermarking Approach for Non-Numeric Relational Database", IEEE International Conference on Communication, Information & Computing Technology (ICCICT) 2012, pp. 1 – 5.

[4]. R.Sion "Proving ownership over categorical data", 20th International Conference of Data Engineering, 2004 ,pp.584 – 595.

[5] Jianhua Sun,Zaihui Cao, Zhongyan Hu, "Multiple Watermarking Relational Databases Using Image", IEEE International conference on MultiMedia and Information Technology, 2008, pp-373-376.

[6]. Y.Li, H.Guo and S.Jajodia, "Tamper Detection and Localization for Categorical Data Using Fragile Watermarks," In Proc. of the 4th ACM Workshop on Digital Rights Management, Washington DC, USA, 2004, pp. 73-82,2004.

[7]. I.Kamel,"A schema for protecting the integrity of databases", Computer and Security, 2009,pp. 698-709.

[8]. J.Guo, "Fragile Watermarking Scheme for Tamper Detection of Relational Database", International Conference on Computer and Management – CAMAN (2011), pp. 1- 4.

[9]. F.Chen, H.He & H.Wang, "A fragile watermarking scheme for audio detection and recovery" IEEE conference on Image and Signal Processing, (CISP),Vol. 5, 2008, pp. 135-138.

[10]. R.Chamlaw, I.Usman, & Khan,"A Dual watermarking method for secure image authentication and recovery",13th International IEEE Multi-topic Conference, 2009, pp. 1-4.

[11]. M.Chen, & X.Sun, "A digital image watermarking of self- recovery base on the SPIHT algorithm", 2nd International Conference on IEEE Signal Processing Systems (ICSPS),Vol.2, 2010, pp.621-624.

[12]. H. Khataeimaragheh and H. Rashidi, "A Novel Watermarking Scheme For Detecting And Recovering Distortions In Database Tables", International Journal of Database Management Systems(Aug. 2010), pp.1-11.

[13]. J.A.Hartigan & M.A.Wong, "A k-means clustering algorithm" , Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979 , pp. 100-108.

[14].National geochemical Database, http://mrdata.usgs.gov/geochem.

[15] M. Shehab, E. Bertino, and A. Ghafoor, "Watermarking Relational Databases Using Optimization-Based Techniques," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 1, Jan. 2008, pp. 116-129,.

[16]. Schneier, B. , John Wiley, Applied cryptography, New York (1996).