Middle-East Journal of Scientific Research 24 (4): 1517-1523, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.04.23379

RWTT-Reversible Watermarking Technique Based on Time-Stamping in Relational Database

M. Parameswari, M. Vidhya, V. Sharmila and J. Jainee Sharmila

Department of CSE & IT, Kings Engineering College, Chennai, India

Abstract: Watermarking method is to recognizable pattern used to identify authenticity. Intentionally introduced pattern in the data is hard to find and destroy, robust against malicious attack. WATERMARKING, without any exception, has been used for ownership protection of a number of data formats—images, video, audio, software, XML documents, geographic information system (GIS) related data, text documents, relational databases and so on—that are used in different application domains. Recently, intelligent mining techniques are being used on data, extracted from relational databases, to detect interesting patterns (generally hidden in the data) that provide significant support to decision makers in making effective, accurate and relevant decisions; as a result, sharing of data between its owners and legitimate users. The owner of the Relational Database embeds the watermark data, the distortions in the original data1 are kept within certain limits, which are defined by the usability constraints, to preserve the knowledge contained in the data. The proposed algorithm embeds every bit of a multi bit watermark (generated from date-time) in each selected row (in a numeric attribute) with the objective of having maximum robustness even if an attacker is somehow able to successfully corrupt the watermark in some selected part of the data set.

Key words: Reversible Watermarking · Data Partitioning Algorithm · Canny Edge Detection Algorithm · Robustness · Numerical Data · Data Recovery · Binary Conversion · Time stamp

INTRODUCTION

Reversible Watermarking Techniques appeared in literature during the last five years approximately is presented in this paper. Watermarking method is to recognizable pattern used to identify authenticity [1]. Intentionally introduced pattern in the data is hard to find and destroy, robust against malicious attack. watermarking, without any exception, has been used for ownership protection of a number of data formats—images, video, audio, software, XML documents, geographic information system (GIS) related data, text documents, relational databases and so on—that are used in different application domains. Recently, intelligent mining techniques are being used on data, extracted from relational databases [2], to detect interesting patterns (generally hidden in the data) that provide significant support to decision makers in making effective, accurate and relevant decisions; as a result, sharing of data between its owners and legitimate users [3]. The owner of the Relational Database embeds the watermark data, the distortions in the original data1 are kept within certain limits, which are defined by the usability constraints, to preserve the knowledge contained in the data. The proposed algorithm embeds every bit of a multibit watermark (generated from date-time) in each selected row (in a numeric attribute) with the objective of having maximum robustness even if an attacker is somehow able to successfully corrupt the watermark in some selected part of the data set [4].

Related Work: The first irreversible watermarking technique for Relational Databases was proposed by Agrawal and Kiernan in R. Agrawal and J. Kiernan, “Watermarking relational databases,” in the year 2002. Similarly, the first reversible watermarking scheme for relational databases was proposed by Y. Zhang, B. Yang and X.M. Niu,“Reversible watermarking for relational database authentication,” in the year 2006. 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

Corresponding Author: M. Parameswari, Department of CSE & IT, Kings Engineering College, Chennai, India.

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), G. Gupta and J. Pieprzyk, “Reversible and blind database watermarking using difference expansion,” in the year 2008, A. M. Alattar, “Reversible watermark using difference expansion of triplets,” in the year 2003, G. Gupta and J. Pieprzyk, “Database relation watermarking resilient against secondary watermarking attacks,” in the year 2009 and those 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 by J.-N. Chang and H.-C. Wu, “Reversible fragile database watermarking technology using difference expansion based on SVR prediction,” in the year 2012 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. Algorithm based on difference expansion watermarking (GADEW) technique is used in a proposed robust and reversible solution for relational databases in the year 2013. 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 M. E. Farfoura and S.-J. Horng, proposed “A novel blind reversible method for watermarking relational databases,”in the year 2010, D. M. Thodi and J. J. Rodriguez, proposed “Prediction-error based reversible watermarking,” in the year 2004, D. M. Thodi and J. J. Rodriguez, proposed “Reversible watermarking by prediction-error expansion,” in the year 2004, D. M. Thodi and J. J. Rodriguez, proposed “Expansion embedding techniques for reversible watermarking,” in the year 2007, M. E. Farfoura, S.-J. Horng, J.-L. Lai, R.-S. Run, R.-J. Chen and M. K. Khan, proposed “A blind reversible method for watermarking relational databases based on a time-stamping protocol,” in the year 2012, X. Li, B. Yang and T. Zeng, proposed “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” in the year 2011 which 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 and Horng 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.

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 1134 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 4, APRIL 2015 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.

In Reversible Watermarking Technique Based on Time-Stamping in Relational Database, we implement a new approach to generate the watermark bits from UTC (Coordinated Universal Time) date time which is the primary time standard used to synchronize the time all over the world. A robust watermark algorithm is used to embed watermark bits into the data set of Database Owner. The watermark embedding algorithm takes a secret key (Ks) and the watermark bits (W) as input and converts a data set D into watermarked data set DW. A cryptographic hash function MD5 is applied on the selected data set to select only those tuples which have an even hash value. The Watermarking process includes Encoding and Decoding Phase. The Encoding phase consist of Data partitioning, Selection of data set for watermarking, Watermark embedding process. Decoding phase consist also these process to extract the Watermarked content.

Proposed System

RWTT Architecture: This section discusses RWTT for Reversible watermarking Technique for relational database that improves data recovery using timestamping these includes Data partitioning, Tuple selection, UTC Date_Time Bit Generator, Edge Detection Authentication, Marked Tuples Identification and Watermark Extraction phase.

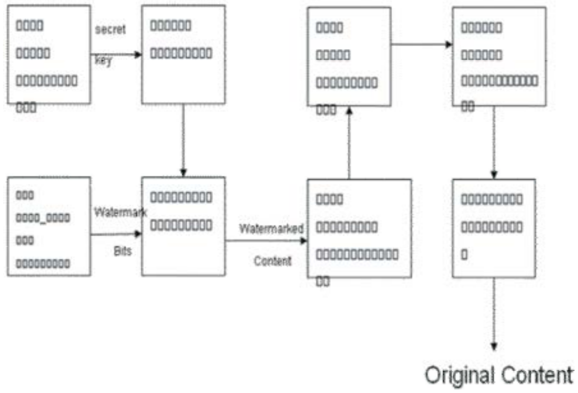
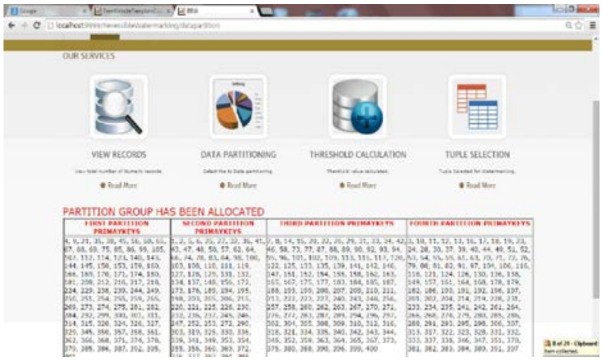


Fig. 1: Main architecture of RWTT

Data Partitioning: Data Group Partitioning includes the Data partitioning Relational Numerical Database Watermarking. Data Partitioning comes under Watermark Encoding Phase which has been done by owner of the DataBase(ie) Admin. The data partitioning algorithm partitions the data set into logical groups by using data partitioning algorithm [5, 6].

$$par(r)=H(ks||H(r.Pk||ks))\bmod m$$

Where r: PK is the primary key of the tuple r,H() is a cryptographic hash, Function Message Digest (MD5),|| is the concatenation, ks is a secret key. Logical groups or Partitions has been arrived after applied this algorithm. Admin has to decide the group's length that is m.



PARTITION GROUP HAS BEEN ALLOCATED

FIRST PARTITION PRIMAYKEYS	SECOND PARTITION PRIMAYKEYS	THIRD PARTITION PRIMAYKEYS	FOURTH PARTITION PRIMAYKEYS
[illegible]	[illegible]	[illegible]	[illegible]

Fig. 2 Data Partitioning

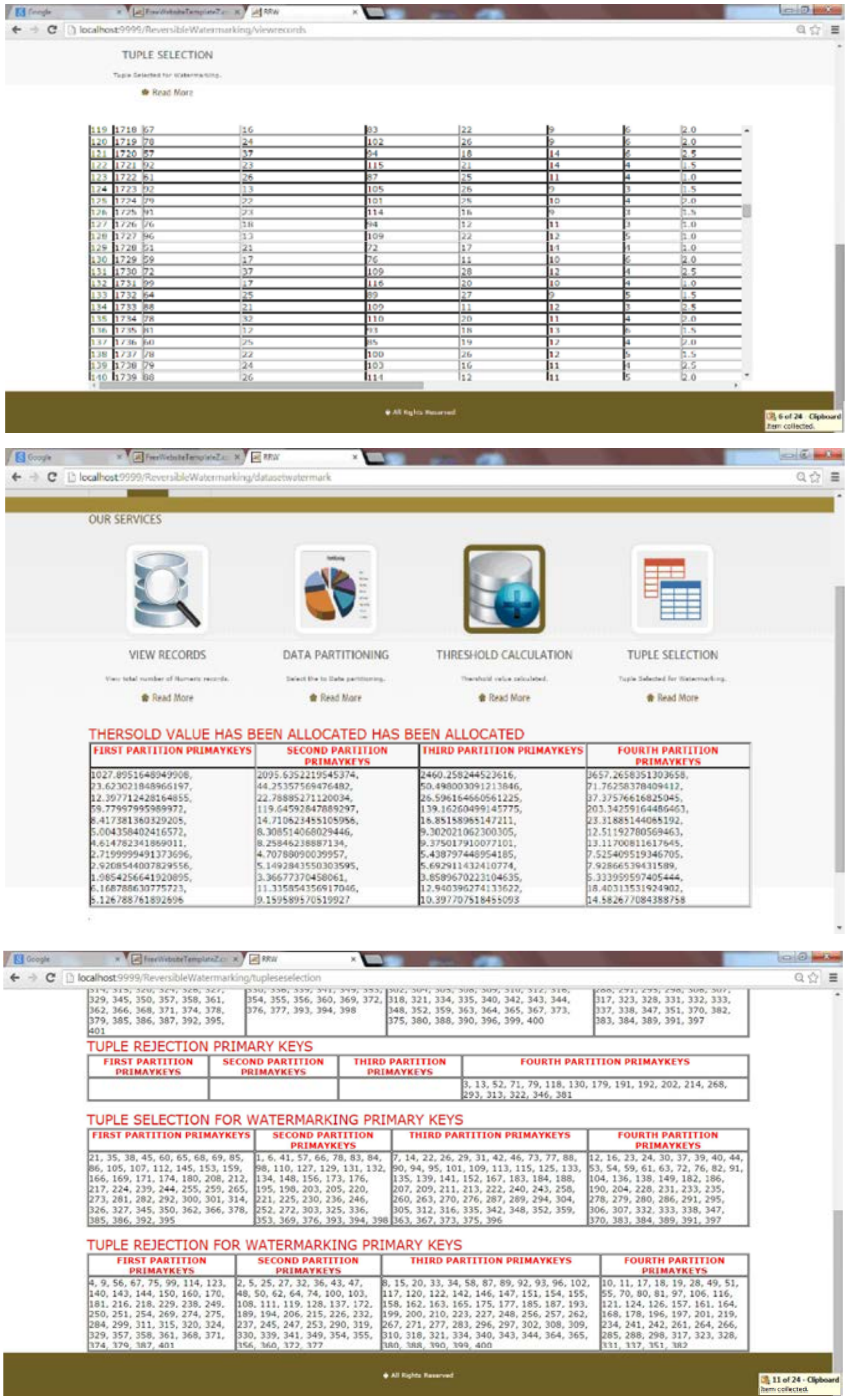
Tuple Selecion: A Tuple is one record or one row in a Relational Database.

In this phase to Select the Particular tuples For embedding Watermarked Content. Threshold Computation is a method computed for each attribute. If the value of any attribute of a tuple is above its respective computed threshold, it is selected for Encoding Process [7]. The data selection threshold for an attribute is calculated using the Threshold equation.

$$T=c^{*}\ Mean+\ Standard\ Deviation$$

c is the confidential factor with a value between 0 and 1. The confidential factor c is kept secret to make it very difficult for an attacker to guess the selected tuples in which the watermark is inserted [8]. We select only those tuples, during the encoding process, whose values are above T. Collect Selected tuples for Encoding and apply Hash Value Computation [9].

This step achieves two objectives: 1) it further enhances the watermark security by hiding the identity of the watermarked tuples from an intruder; and 2) it further reduces the number of to-be-watermarked tuples to limit distortions in the data set.If the Hash Value Computation is Satisfied Select the tuples for Watermarking bits from Selected tuples for Encoding process.



localhost:9999/ReversibleWatermarking/viewrecords

TUPLE SELECTION

Tuple Selected for Watermarking.

119	1718	67	16	83	22	9	6	2.0
120	1719	78	24	102	26	9	6	2.0
121	1720	57	37	94	18	14	6	2.5
122	1721	92	23	115	21	14	4	1.5
123	1722	61	26	87	25	11	4	1.0
124	1723	92	13	105	26	9	3	1.5
125	1724	79	22	101	25	10	4	2.0
126	1725	91	23	114	16	9	3	1.5
127	1726	76	18	94	12	11	3	1.0
128	1727	96	13	109	22	12	5	1.0
129	1728	51	21	72	17	14	4	1.0
130	1729	59	17	76	11	10	6	2.0
131	1730	72	37	109	28	12	4	2.5
132	1731	99	17	116	20	10	4	1.0
133	1732	64	25	89	27	9	5	1.5
134	1733	88	21	109	11	12	3	2.5
135	1734	78	32	110	20	11	4	2.0
136	1735	81	12	93	18	13	6	1.5
137	1736	60	25	85	19	12	4	2.0
138	1737	78	22	100	26	12	5	1.5
139	1738	79	24	103	16	11	4	2.5
140	1739	88	26	114	12	11	5	2.0

localhost:9999/ReversibleWatermarking/datasetwatermark

OUR SERVICES

VIEW RECORDS

View total number of Numeric records.

DATA PARTITIONING

Select the to Data partitioning.

THRESHOLD CALCULATION

Threshold value calculated.

TUPLE SELECTION

Tuple Selected for Watermarking.

THERSOLD VALUE HAS BEEN ALLOCATED HAS BEEN ALLOCATED

FIRST PARTITION PRIMAYKEYS	SECOND PARTITION PRIMAYKEYS	THIRD PARTITION PRIMAYKEYS	FOURTH PARTITION PRIMAYKEYS
1027.8951648949908, 23.623021848966197, 12.397712428164855, 59.77997995989972, 8.417381360329205, 5.004358402416572, 4.614782341869011, 2.7199999491373696, 2.9208544007829556, 1.9854256641920895, 6.168788630775723, 5.126788761892696	2095.6352219545374, 44.25357569476482, 22.78885271120034, 119.64592847889297, 14.710623455105956, 8.308514068029446, 8.25846238887134, 4.70788090039957, 5.149284355030359 5, 3.36677370458061, 11.335854356917046, 9.15958957051992 7	2460.258244523616, 50.498003091213846, 26.596164660561225, 139.16260499145775, 16.85158965147211, 9.302021062300305, 9.375017910077101, 5.438797448954185, 5.692911432410774, 3.858967022310463 5, 12.940396274133622, 10.397707518455093	3657.2658351303658, 71.76258378409412, 37.37576616825045, 203.34259164486463, 23.31885144065192, 12.51192780569463, 13.11700811617645, 7.525409519346705, 7.92866539431589, 5.333959597405444, 18.40313531924902, 14.582677084388758

localhost:9999/ReversibleWatermarking/tupleseselection

329, 345, 350, 357, 358, 361, 362, 366, 368, 371, 374, 378, 379, 385, 386, 387, 392, 395, 401	354, 355, 356, 360, 369, 372, 376, 377, 393, 394, 398	318, 321, 334, 335, 340, 342, 343, 344, 348, 352, 359, 363, 364, 365, 367, 373, 375, 380, 388, 390, 396, 399, 400	317, 323, 328, 331, 332, 333, 337, 338, 347, 351, 370, 382, 383, 384, 389, 391, 397

TUPLE REJECTION PRIMARY KEYS

FIRST PARTITION PRIMAYKEYS	SECOND PARTITION PRIMAYKEYS	THIRD PARTITION PRIMAYKEYS	FOURTH PARTITION PRIMAYKEYS
			3, 13, 52, 71, 79, 118, 130, 179, 191, 192, 202, 214, 268, 293, 313, 322, 346, 381

TUPLE SELECTION FOR WATERMARKING PRIMARY KEYS

FIRST PARTITION PRIMAYKEYS	SECOND PARTITION PRIMAYKEYS	THIRD PARTITION PRIMAYKEYS	FOURTH PARTITION PRIMAYKEYS
21, 35, 38, 45, 60, 65, 68, 69, 85, 86, 105, 107, 112, 145, 153, 159, 166, 169, 171, 174, 180, 208, 212, 217, 224, 239, 244, 255, 259, 265, 273, 281, 282, 292, 300, 301, 314, 326, 327, 345, 350, 362, 366, 378, 385, 386, 392, 395	1, 6, 41, 57, 66, 78, 83, 84, 98, 110, 127, 129, 131, 132, 134, 148, 156, 173, 176, 195, 198, 203, 205, 220, 221, 225, 230, 236, 246, 252, 272, 303, 325, 336, 353, 369, 376, 393, 394, 398	7, 14, 22, 26, 29, 31, 42, 46, 73, 77, 88, 90, 94, 95, 101, 109, 113, 115, 125, 133, 135, 139, 141, 152, 167, 183, 184, 188, 207, 209, 211, 213, 222, 240, 243, 258, 260, 263, 270, 276, 287, 289, 294, 304, 305, 312, 316, 335, 342, 348, 352, 359, 363, 367, 373, 375, 396	12, 16, 23, 24, 30, 37, 39, 40, 44, 53, 54, 59, 61, 63, 72, 76, 82, 91, 104, 136, 138, 149, 182, 186, 190, 204, 228, 231, 233, 235, 278, 279, 280, 286, 291, 295, 306, 307, 332, 333, 338, 347, 370, 383, 384, 389, 391, 397

TUPLE REJECTION FOR WATERMARKING PRIMARY KEYS

FIRST PARTITION PRIMAYKEYS	SECOND PARTITION PRIMAYKEYS	THIRD PARTITION PRIMAYKEYS	FOURTH PARTITION PRIMAYKEYS
4, 9, 56, 67, 75, 99, 114, 123, 140, 143, 144, 150, 160, 170, 181, 216, 218, 229, 238, 249, 250, 251, 254, 269, 274, 275, 284, 299, 311, 315, 320, 324, 329, 357, 358, 361, 368, 371, 374, 379, 387, 401	2, 5, 25, 27, 32, 36, 43, 47, 48, 50, 62, 64, 74, 100, 103, 108, 111, 119, 128, 137, 172, 189, 194, 206, 215, 226, 232, 237, 245, 247, 253, 290, 319, 330, 339, 341, 349, 354, 355, 356, 360, 372, 377	8, 15, 20, 33, 34, 58, 87, 89, 92, 93, 96, 102, 117, 120, 122, 142, 146, 147, 151, 154, 155, 158, 162, 163, 165, 175, 177, 185, 187, 193, 199, 200, 210, 223, 227, 248, 256, 257, 262, 267, 271, 277, 283, 296, 297, 302, 308, 309, 310, 318, 321, 334, 340, 343, 344, 364, 365, 380, 388, 390, 399, 400	10, 11, 17, 18, 19, 28, 49, 51, 55, 70, 80, 81, 97, 106, 116, 121, 124, 126, 157, 161, 164, 168, 178, 196, 197, 201, 219, 234, 241, 242, 261, 264, 266, 285, 288, 298, 317, 323, 328, 331, 337, 351, 382

Fig. 3: Selection of Tuples

Watermark Embedding: The watermark generating function takes date-time stamp as an input and then generates watermark bits b1b2... bn from this date-time stamp [10]. These bits are given as input to the watermark encoding function. The date-time stamp "might" also help to identify additive attacks in which an attacker wants to rewatermark the data set. To construct a watermarked data set, these watermark bits are embedded in the original data set by using watermark embedding algorithm. Watermark generated from date-time in each selected row. The watermark bits are embedded in the selected tuples using a robust watermarking function. Our technique embeds each bit of the watermark in every selected tuple of each partition.

EDGE Detection Authentication and Watermark Decoding: Edge detection Authentication is proposed as an alternative solution to text based. It is mainly depends on images rather than alphanumerical. The main argument here is that pass-images from the challenge set and then he/she will be authenticated users are better at recognizing and memorizing pictures. During Registration phase Admin has to provide some images to the user. In the registration phase the user is supposed to choose the pass-images for the verification phase. That image has to be Stored in Server For that Specific User. During Login phase Admin has to converting the raw image to a gray scale followed by Edge detection image. The idea here is the user will have a challenge set which contains decoy and pass-images. The decoy images are randomly generated by the scheme during the verification process. On the other hand, pass-image will be the users selected images. Basically authentication is simple; a legitimate user needs to correctly identify pass-images from the challenge set and then he/she will be authenticated. Watermark Extraction process in the Decoding phase. The Watermarked Content has to be extracted only by legitimate user to give the proper ownership. If the User ownership content is matched by the Admin generated content, Decoding process has to done. Otherwise it's not done.

Watermark Detection: Our experiments showed that when more than 80 percent of the data was deleted, the water mark was detected with 100 percent accuracy. We compared RRW with well known reversible watermarking techniques for detecting the watermark information after such attacks. RRW has shown 100 percent accuracy when up to 90 percent 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.

Configuring the Channels That Are Available: Any TV will have a list of channels available and XleTView also offers this functionality. Unlike a real TV, you need to tell it what channels are available and you can do this by editing the config/channels.xmlfile. The default version of this file looks like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CHANNELS>
  <CHANNEL>
    <NAME>0</NAME>
    <MEDIA>config/defaultbg.jpg</MEDIA>
  </CHANNEL>
</CHANNELS>
```

As you can see, each channel definition consists of two parts. The < NAME> element contains the channel name or number that will be assigned to this channel. The< MEDIA> element tells, XleTView what it should display in the background when that channel is selected. This can either be a JPEG image (which should be 720 pixels wide by 576 pixels high) or it can be an AVI file if you prefer a moving background. Please note that only some types of AVI file are supported - see the section on using video with XleT View for more details. When you first start XleTView, it will display the channel listed first in the channels.xml file. In version 0.3.6 of XleTView it is not possible to change the channel using the keys on the remote.

RESULTS AND DISCUSSION

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets. The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM. Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

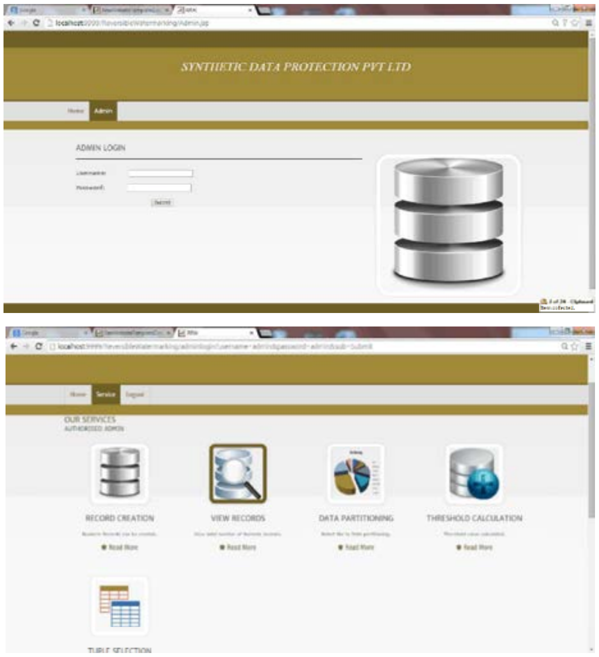


Fig. 4: Analysis of Data Protection

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 a duplicate tuples or randomly generated fake tuples into the database. RRW is resilient against these types of attacks. Data recovery. When Mallory tries to insert 50 percent tuples within the range of values of the watermarked feature, 100 percent data is recovered. The reason for successful data recovery is the marking of all the tuples (majority vote) and use of hr. 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 percent in RRW, 89 to 94 percent in DEW, 100 percent GADEW and 98 to 100 percent in PEEW technique.

Deletion Attacks: In this type of attack, Mallory might delete a 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 a 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 useful-ness, a 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. The original data is recovered with more than 50 percent accuracy in case 50 percent data was deleted. 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.

Alteration Attacks: In such attacks, Mallory can modify the value of the watermarked feature within a certain range. Mallory can make random or ?xed alterations within the range of minimum to maximum values. When Mallory alters a 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. RRW demonstrated more than 65 percent of data recovery when half of the tuples get altered.

Again, in the presence of such attacks, RRW provided 100 percent accuracy with 90 percent of attacked tuples whereas DEW, GADEW and PEEW gave less accuracy. It is observed that RRW provides 100 percent watermark detection even with 90 percent 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.

CONCLUSION

We achieved to maintain the ownership of Relational Database and also minimizing distortion in the watermarked content proves the effectiveness of RRW against malicious attacks.

REFERENCES

1. Wong, P.W., 1998. A public key watermark for image verification and authentication, in Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, IEEE, 1: 455-459.

2. Wong, P.W. and N. Memon, 2001. Secret and public key image watermarking schemes for image authentication and ownership verification," Image Processing, IEEE Transactions on, 10(10): 1593-1601.
3. Petitcolas, F.A., 2000. Watermarking schemes evaluation, Signal Processing Magazine, IEEE, 17(5): 58-64.
4. Agrawal, R. and J. Kiernan, 2002. Watermarking relational databases, in Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, pp: 155-166.
5. Sion, R., M. Atallah and S. Prabhakar, 2005. Rights protection for categorical data, Knowledge and Data Engineering, IEEE Transactions on, 17(7): 912-926.
6. Subramanya, S. and B.K. Yi, 2006. Digital rights management, Potentials, IEEE, 25(2):: 31-34.
7. Gill, P.E., W. Murray and M.A. Saunders, 2005. Snopt: An sqp algorithm for large-scale constrained optimization, SIAM review, 47(1): 99-131.
8. Parsopoulos, K.E. and M.N. Vrahatis, 2002. Particle swarm optimization method for constrained optimization problems, Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies, 76: 214-220.
9. Hassan, R., B. Cohanim, O. De Weck and G. Venter, 2005. A Comparison Of Particle Swarm Optimization And The Genetic Algorithm, in 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. American Institute of Aeronautics and Astronautics, pp: 1-13.
10. Brassil, J.T., S. Low and N.F. Maxemchuk, 1999. Copyright protection for the electronic distribution of text documents, Proceedings of the IEEE, 87(7): 1181-1196.