Journal of Computer Science 4 (12): 1024-1029, 2008
ISSN 1549-3636
© 2008 Science Publications

Robust and Blind Watermarking of Relational Database Systems

[1]Ali Al-Haj and [2]Ashraf Odeh
[1]Princess Sumaya University for Technology, P.O. Box 1928, Al-Jubeiha, 11941 Amman, Jordan
[2]Royal Scientific Society, P.O. Box 1438, Al-Jubeiha, 11941 Amman, Jordan

Abstract: Problem statement: Digital multimedia watermarking technology was suggested in the last decade to embed copyright information in digital objects such images, audio and video. However, the increasing use of relational database systems in many real-life applications created an ever increasing need for watermarking database systems. As a result, watermarking relational database systems is now merging as a research area that deals with the legal issue of copyright protection of database systems. **Approach:** In this study, we proposed an efficient database watermarking algorithm based on inserting binary image watermarks in non-numeric mutli-word attributes of selected database tuples. **Results:** The algorithm is robust as it resists attempts to remove or degrade the embedded watermark and it is blind as it does not require the original database in order to extract the embedded watermark. **Conclusion:** Experimental results demonstrated blindness and the robustness of the algorithm against common database attacks.

Key words: Digital watermarking, relational database systems, copyright protection, ownership verification, usability, robustness, watermark embedding, watermark extraction

INTRODUCTION

Digital images, video and audio are examples of digital assets which have become easily accessible by ordinary people around the world. However, the owners of such digital assets have long been concerned with the copyright of their digital products, since copying and distributing digital assets across the Internet was never easier and possible as its nowadays. Therefore, researchers have been looking for ways to protect the ownership of digital assets for a long time. Digital watermarking technology was suggested lately as an effective solution for protecting the copyright of digital assets[3,9,12]. This technology provides ownership verification of a digital product by inserting imperceptive information into the digital product. Such 'right witness' information is called the watermark and it is inserted in such a way that the usefulness of the product remains, in addition to providing it with robustness against attempts to remove the watermark.

Most watermarking research concentrated on watermarking multimedia data objects such as still images and video[8,13,18] and audio[3,5,15]. However, watermarking of database systems started to receive attention because of the increasing use of database systems in many real-life applications. Examples where database watermarking might be of a crucial importance include protecting rights of outsourced relational databases and allowing the creation of secured and copyright-protected web-based services that enable users to search and access databases remotely[6,10,19].

Due to the different characteristics between images or audio and relational data, there exists no image or audio watermarking method suitable for watermarking relational databases. Therefore, relational database watermarking is, in fact, a process challenged by many factors such as data redundancy fewness, relational data out-of-order and frequent updating. Moreover, database systems watermarking has unique and sometimes complex, requirements that differ from those required for watermarking digital audio-visual products. Due to such unique requirements and challenges, literature on watermarking relational databases is very limited and has focused mainly on embedding short strings of binary bits in randomly selected locations in numerical databases. Most proposed algorithms lack robustness against bit-level attacks such as bit-setting, bit-resetting and bit-flipping. Other database watermarking algorithms embed watermark information in the statistical properties of tuples rather than in the data itself. These algorithms are computation-intensive and still lack solid mathematical formulations.

Corresponding Author: Ali Al-Haj1, Princess Sumaya University for Technology, P.O. Box 1928, Al-Jubeiha, 11941 Amman, Jordan

In this study, a binary image is used to watermark a given relational database system. The watermark image is embedded in non-numeric, multi-word, attributes of a selected number of tuples of the database. The algorithm is robust as it resists attempts to remove or degrade the embedded watermark and it is blind as it does not require the original database in order to extract the embedded watermark.

MATERIALS AND METHODS

Database watermarking research: Watermarking of relational database systems is a relatively new field and thus research literature has been very limited and reported results are insufficient[14,17,27]. Accordingly, we anticipate that advancements in this area will continue, but at a slow pace due to the challenges and unique requirements imposed by the nature of relational databases. In what follows we will describe briefly such unique requirements and challenges. We will also outline classes of database watermarking algorithms that have been proposed in literature.

Unique Requirements of Database Watermarking

Watermarking database systems has unique requirements that differ from those required for watermarking digital image and audio systems[2]. The watermarked database must maintain the following properties:

Usability: That mount of change in the database caused by the watermarking process should not result in degrading the database and making it useless. The amount of allowable change differs from one database to another, depending on the nature of stored records.

Robustness: Watermarks embedded in the database should be robust against attacks to erase them. That is, the database watermarking algorithm must be developed in such a way to make it difficult for an adversary to remove or alter the watermark beyond detection without destroying usability of the database.

Blindness: Watermark extraction should neither require the knowledge of the original un-watermarked database nor the watermark itself. This property is critical as it allows the watermark to be detected in a copy of the database relation, irrespective of later updates to the original relation.

Structure: A database is made of inter-related tuples. The tuples that are joined before the watermarking process should not be altered during watermarking. Moreover, scale and classification must be considered during the watermarking process since they have impact on the semantics of the database.

Security: Choice of he watermarked tuples, attributes, bit positions should be secret and be only known through the knowledge of a secret-key. Owner of the database should be the only one who has knowledge of a secret-key.

Incremental watermarking: After a database has been watermarked, the watermarking algorithm should compute the watermark values for added or modified tuples only. The already unaltered watermarked tuples should not be watermarked again.

Challenges of database watermarking: Watermarking relational database is challenged by the following factors[28]:

Few redundant data: A relational database is made up of tuples, each indicating an independent object. Therefore, watermarks basically have no places to hide.

Out-of-order relational data: Tuples of a relational database have no fixed location. This makes building a corresponding relative is very difficult in relational databases.

Frequent updating: Insertion, dropping, updating of operation of relational database is very frequent. Without malicious intention, users often casually drop some tuples or attributes. On the other hand, the pirate can add or substitute the tuples and attributes.

Existing database watermarking methods: There has been a few proposed relational database watermarking algorithms. Published algorithms can be classified as bit-level watermarking algorithms, statistical-property watermarking algorithms and image-based watermarking algorithms. The three classes of algorithms operate on numeric attributes of relational databases. A brief description of each class is given:

Bit-level watermarking algorithms: In these algorithms, certain attributes of a selected subset of tuples are chosen to hide watermark bit information. Attribute selection is based on the value of a hash function. For each selected attribute, some bit positions will be marked amongst a predetermined number of least significant bits of the attribute[1,2,4,7,11,16,28].

Statistical-property watermarking algorithms: In theses algorithms, watermark bits are not encoded in the data itself, but rather in actual data distribution properties of s subset of tuples. The complete set of tuples making up the database is partitioned into a maximal number of unique, nonintersecting subsets of tuples. For each selected subset of tuples, a watermark bit is embedded by making minor changes to some of the data values in the tuples, in such a way to make the subset's average and variance values reach two possible values depending whether the watermark bitis 0 or 1[20-25].

Image-based watermarking algorithms: In these algorithms, an image is used to watermark the database. The image is transformed into bits which represent the watermark bits. The bits are embedded in carefully chosen locations in database and if recovered correctly can be used to reconstruct the embedded image[26,28]. This class of watermarking methods can be considered as a sub-class of the bit-level watermarking class.

Proposed watermarking algorithm: In our proposed algorithm, a binary image is used to watermark relational databases. The bits of the image are segmented into short binary strings that are encoded in non-numeric, multi-word attributes of selected tuples of the database. The embedding process of each short string is based on creating a double-space at a location determined by the decimal equivalent of the short string. Extraction of a short string is done by counting number of single-spaces between two separated double-space locations. The image watermark is then constructed by converting the decimals into binary strings. A major advantage of using the space-based watermarking is the large bit-capacity available for hiding the watermark. This facilitates embedding large watermarks or multiple small watermarks. This is in contrast to bit-based algorithms where watermark bits have limited potential locations that can be used to hide bits without being subjected to removal or destruction. Our proposed algorithm has two procedures: watermark embedding procedure and watermark extraction procedure. The two procedures are described below.

Watermark embedding procedure: The watermark embedding procedure consists of the following operational steps:

Step 1: Arrange the watermark image into m strings each of n bits length

Step 2: Divide the database logically into sub-sets of tuples. A sub-set has m tuples

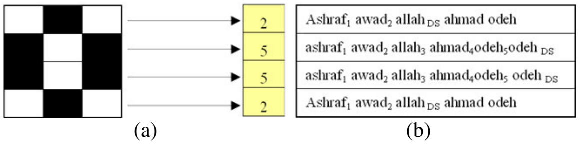

2
5
5
2
Ashraf$_1$ awad$_2$ allah$_{DS}$ ahmad odeh
ashraf$_1$ awad$_2$ allah$_3$ ahmad$_4$odeh$_5$odeh $_{DS}$
ashraf$_1$ awad$_2$ allah$_3$ ahmad$_4$odeh$_5$ odeh $_{DS}$
Ashraf$_1$ awad$_2$ allah$_{DS}$ ahmad odeh
(a)
(b)

Fig. 1: (a): Binary image watermark and its decimal equivalent vector, (b): Watermarking example; where subscripts represent space number and DS corresponds to double space

Step 3: Embed the m short stings of the watermark image into each m-tuple sub-set

Step 4: Embed the n-bit binary string in the corresponding tuple of a sub-set as follows:

- Find the decimal equivalent of the string. Let the decimal equivalent be d
- Embed the decimal number d in a pre-selected non-numeric, multi-word attribute by creating a double-space after d words of the attribute

Step 5: Repeat step 4 for each tuple in the subset

Step 6: Repeat steps 4 and 5 for each subset of the database under watermarking

An illustration of embedding the binary watermark into a sub-set of tuples is shown in Fig. 1. The watermark is a of 4×3 binary image. Each of the four 3-bit binary strings is transformed into its decimal equivalent as shown in Fig. 1a and embedded in the 4-tuple sub-set, as shown in Fig. 1b. The count of numbered single spaces appearing before the Double-Space (DS) indicates the decimal equivalent of the embedded short binary string.

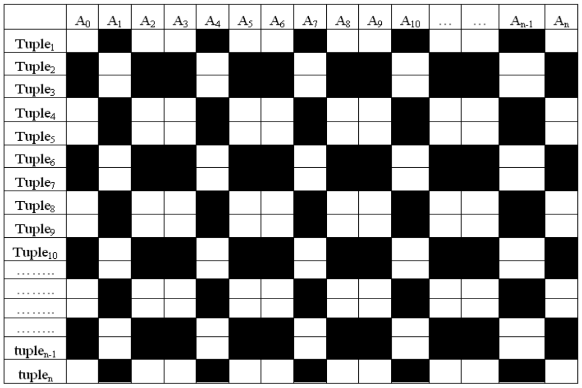
A snapshot of the relational database after embedding the watermark throughout the database is shown in Fig. 2. The tuples in Fig. 2 constitute the database and the A's are the watermarked non-numeric, multi-word attributes for each tuple.

Watermark extraction procedure: The watermark embedding procedure consists of the following operational steps:

Step 1: Arrange the watermark image into m strings each of n bits length

Step 2: Divide the database logically into sub-sets of tuples. A sub-set has m tuples

Step 3: Embed the m short stings of the watermark image into each m-tuple sub-set



	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	…	…	A_{n-1}	A_n
$Tuple_1$		■			■			■			■			■	
$Tuple_2$	■		■	■		■	■		■	■		■	■		■
$Tuple_3$	■		■	■		■	■		■	■		■	■		■
$Tuple_4$		■			■			■			■			■	
$Tuple_5$		■			■			■			■			■	
$Tuple_6$	■		■	■		■	■		■	■		■	■		■
$Tuple_7$	■		■	■		■	■		■	■		■	■		■
$Tuple_8$		■			■			■			■			■	
$Tuple_9$		■			■			■			■			■	
$Tuple_{10}$	■		■	■		■	■		■	■		■	■		■
……..	■		■	■		■	■		■	■		■	■		■
……..		■			■			■			■			■	
……..		■			■			■			■			■	
……..	■		■	■		■	■		■	■		■	■		■
$tuple_{n-1}$	■		■	■		■	■		■	■		■	■		■
$tuple_n$		■			■			■			■			■	

Fig. 2: A snapshot of the watermarked database

Step 4: Embed the n-bit binary string in the corresponding tuple of a sub-set as follows:

- Find the decimal equivalent of the string and give it the symbol d
- Embed the decimal number d in a pre-selected non-numeric, multi-word attribute by creating a double-space after d words of the attribute

Step 5: Repeat step 4 for each tuple in the subset
Step 6: Repeat steps 4 and 5 for each subset of the database under watermarking

RESULTS

The proposed algorithm has been evaluated and tested on an experimental database that we have constructed. The database consists of 1000 tuples and runs under the Oracle platform. We concentrated our performance evaluation on the robustness of the proposed algorithm by virtue of the fact that, database watermarking algorithms must be developed in such a way to make it difficult for an adversary to remove or alter the watermark beyond detection without destroying the value of the object. In particular, the database watermarking algorithm should make the watermarked database robust against the following types of attacks: subset deletion attack, subset addition attack, subset alteration attack and finally subset selection attack. The results are shown in Fig. 3-6.

DISCUSSION

Subset deletion attack: In this type of attack, the attacker may take a subset of the tuples of the watermarked database and hope that the watermark will be removed. The graph shown in Fig. 3 indicates that the watermark will be removed only and only if, all the database was deleted!. That is, removing more than 95% of the database will not result in removing the watermark. This is due to the fact that the proposed algorithm embeds the same watermark everywhere in the database, making this type of attack ineffective.

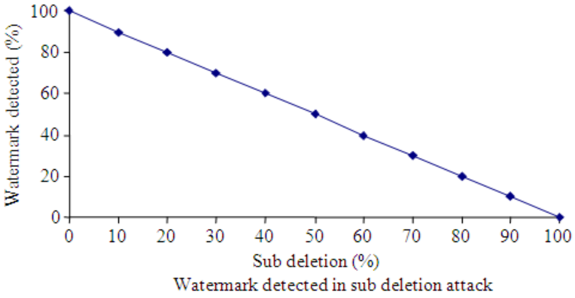


Fig. 3: Robustness results due to the 'subset deletion attack'

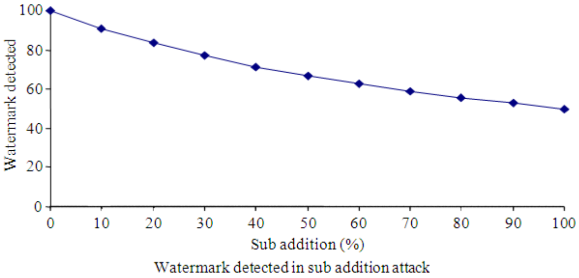


Fig. 4: Robustness results due to the 'subset addition attack'

Subset addition attack: In this type of attack, the attacker adds a set of tuples to the original databse. This is one of the most difficult attacks to defeat. The attacker may add some tuples to the watermarked table. But this form of attack has little impact on the watermark embedded through our algorithm. The graph shown in Fig. 4 indicates that the watermark will never be removed even if the added tuples are as many as the original tuples. That's, only the added tuples will not carry the watermark information.

Subset alteration attack: In this type of attack, the attacker alters the tuples of the database through operations such as linear transformation. The attacker hopes by doing so to erase the watermark from the database. The graph shown in Fig. 5 indicates that the watermark will remain even if 90 % of the tuples of the database were altered. This is due to the fact that the proposed algorithm embeds the same watermark everywhere in the database, making this type of attack ineffective.

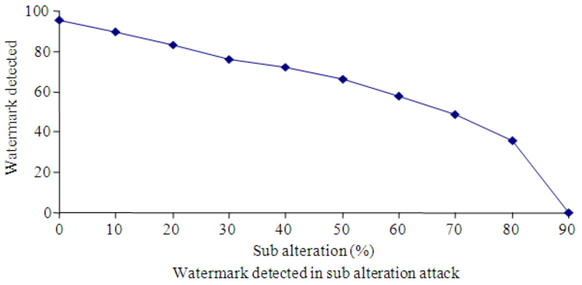


Fig. 5: Robustness results due to the 'subset alteration attack'

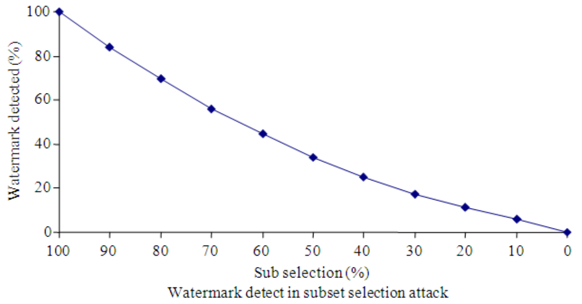


Fig. 6: Robustness results due to the 'subset selection attack'

Subset selection attack: In this type of attack, the attacker randomly selects and uses a subset of the original database that might still provide value for its intended purpose. The attacker hopes by doing so that the selected subset will not contain the watermark. However, since the proposed algorithm embeds the watermark in the whole database, this attack is of little or no threat. The graph shown in Fig. 6 indicates that the watermark will remain even if the attacker selects a subset as small as 10% of the original database. That's no matter how the small subset he selects, the watermark will remain in the selected subset and thus providing the required copyright protection.

CONCLUSION

In this study, we proposed a watermarking algorithm based on hiding watermark bits in spaces of non-numeric, multi-word, attributes of subsets of tuples. A major advantage of using the this approach is the large bit-capacity available to hide large watermarks. This is opposite to the other proposed algorithms where watermark bits have limited potential bit-locations that can be used to hide them effectively without being subjected to removal or destruction. The robustness of the proposed algorithm was verified against a number of database attacks such subset deletion, subset addition, subset alteration and subset selection attacks. Ongoing and future research includes the development of other effective database watermarking algorithms.

REFERENCES

1. Agrawal, R., P. Hass and J. Kiernan, 2003. Watermarking relational data: Framework, algorithms and analysis. Int. J. Very Large Data Bases, 12: 157-169. http://cat.inist.fr/?aModele=afficheN&cpsidt=15040354
2. Agrawal, R. and J. Kiernan, 2002. Watermarking relational databases. Proceeding of the 28th International Conference on Very Large Databases, (ICVLD'02), Hong Kong, China, pp: 1-12. http://www.cse.ust.hk/vldb2002/VLDB2002-proceedings/papers/S05P03.pdf
3. Arnold, M., M. Schumucker and S. Wolthusen, 2003. Techniques and Applications of Digital Watermarking and Content Protection. Artech House, Boston, MA. ISBN: 10: 1580531113, pp: 274.
4. Arnold, M., 2000. Audio watermarking: Features, applications and algorithms. Proceeding of the IEEE International Conference on Multimedia and Expo, July 30-Aug. 2, New York, USA., pp: 1013-1016. DOI: 10.1109/ICME.2000.871531
5. Bassia, P., L. Pitas and N. Nikolaidis, 2001. Robust audio watermarking in the time-domain. IEEE Trans. Multimedia, 3: 232-242. DOI: 10.1109/6046.923822
6. Bertino, E., B. Ooi, Y. Yang and R. Deng, 2005. Privacy and ownership preserving of outsourced medical data. Proceeding of the 4th International Conference on Data Engineering, Apr. 5-8, CA., USA., pp: 521-532. DOI: 10.1109/ICDE.2005.111
7. Gross-Amblard, D., 2003. Query preserving watermarking of relational databases and XML documents. Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, June 9-11, New York, USA., pp: 191-201. DOI: 10.1145/773153.773172
8. Hartung, F. and B. Girod, 1998. Watermarking of uncompressed and compressed video. Signal Process., 66: 238-301. DOI: 10.1016/S0165-1684(98)00011-5
9. Hartung, F. and M. Kutter, 1999. Multimedia watermarking techniques. Proc. IEEE, 87: 1079-1107. DOI: 10.1109/5.771066

10. Hildebrandt, E. and G. Saake, 1998. User authentication in multi-database systems. Proceeding of the 9th International Workshop on Database and Expert Systems Applications, Aug. 25-28, Vienna, Austria, pp: 281-286. DOI: 10.1109/DEXA.1998.707414
11. Huang, M., J. Cao, Z. Peng and Y. Fang, 2004. A new watermark mechanism for relational data. Proceeding of the 4th International Conference on Computer and Technology, Sep. 14-16, IEEE Xplore Press, USA., pp: 946-950. DOI: 10.1109/CIT.2004.1357318
12. Katzenbeisser, S. and F. Petitcolas, 2000. Information Hiding: Techniques for Steganography and Digital Watermarking. Artech House, Boston, MA., ISBN: 10: 1580530354, pp: 220.
13. Langelaar, G., I. Setyawan and R. Lagendijk, 2000. Watermarking digital image and video data: A state-of-art overview. IEEE Signal Process. Mag., 17: 20-46.
14. Lee, Y., V. Swarup and S. Jajodia, 2005. Fingerprinting relational databases: Schemes and specialties. IEEE Trans. Depend. Secure Comput., 2: 34-45. DOI: 10.1109/TDSC.2005.12
15. Lemma, A., J. Aprea and L. Kherkhof, 2003. A temporal-domain audio watermarking technique. IEEE Trans. Signal Process., 51: 1088-1097. DOI: 10.1109/TSP.2003.809372
16. Li, Y., V. Swarup and S. Jajodia, 2003. Constructing a virtual primary key for fingerprinting relational data. Proceeding of the 3rd ACM Workshop on Digital Rights Management, Oct. 27-27, Washington, DC., USA., pp: 133-141. http://portal.acm.org/citation.cfm?id=947380.947398
17. Li, Y. and R. Deng, 2006. Publicly verifiable ownership protection of relational database. Proceedings of the 2006 ACM Symposium on Information, computer and communications security, Mar. 21-24, Taipei, Taiwan, pp: 78-89. http://portal.acm.org/citation.cfm?id=1128817.1128832
18. Potdar, V., S. Han and E. Chang, 2005. A survey of digital image watermarking techniques. Proceeding of the 3rd International IEEE Conference on Industrial Informatics, Aug. 10-12, IEEE Xplore Press, USA., pp: 709-716. DOI: 10.1109/INDIN.2005.1560462
19. SIIA, 2000. Database Protection: Making the case for a new federal database protection law. http://www.siia.net/sharedcontent/gove/issues/ip/dbbrief.html
20. Sion, S., M. Atallah and S. Prabhakar, 2005. Rights protection for relational data. IEEE Trans. Knowl. Data Eng., 16: 912-926. http://direct.bl.uk/bld/PlaceOrder.do?UIN=172151710&ETOC=RN&from=searchengine
21. Sion, S., M. Atallah and S. Prabhakar, 2003. Rights protection for relational data. Proceeding of the ACM International Conference on Management of Data, June 9-12, San Diego, California, pp: 98-109. http://portal.acm.org/citation.cfm?doid=872757.872772
22. Sion, S., M. Atallah and S. Prabhakar, 2004. Relational data rights protection through watermarking. IEEE Trans. Knowl. Data Eng., 16: 912-926.
23. Sion, S., M. Atallah and S. Prabhakar, 2004. Wmdb.*: Rights protection for numeric relational data. Proceeding of the 20th International Conference on Data Engineering, Mar. 30-Apr. 2, USA., pp: 863. DOI: 10.1109/ICDE.2004.1320091
24. Sion, S., M. Atallah and S. Prabhakar, 2004. Rights protection for categorial data. IEEE Trans. Knowl. Data Eng., 17: 1509-1525. DOI: 10.1109/TKDE.2004.94
25. Sion, S., M. Atallah and S. Prabhakar, 2004. Proving ownership over categorical data. Proceeding of the 20th International Conference on Data Engineering, Mar. 30-Apr. 2, IEEE Xplore Press, USA., pp: 584-595. DOI: 10.1109/ICDE.2004.1320029
26. Wu, M., E. Tang and B. Liu, 2000. Data hiding in digital binary image. Proceeding of the IEEE International Conference on Multimedia and Expo, July 30-Aug. 02, New York, USA., pp: 393-396. DOI: 10.1109/ICME.2000.869623
27. Zhang, Z., X. Jin, J. Wang and D. Li, 2003. A robust watermarking scheme for relational data. Proceeding of the 13th Workshop on Information Technology and Engineering, USA.
28. Zhang, Z., X. Jin, J. Wang and D. Li, 2004. Watermarking relational database using image. Proceeding of the International Conference on Machine Learning and Cybernetics, Aug. 26-29, IEEE Xplore Press, USA., pp: 1739-1744. DOI: 10.1109/ICMLC.2004.1382056