ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 17, Issue, 09, pp.34579-34584, September, 2025
DOI: https://doi.org/10.24941/ijcr.49479.09.2025

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

A NOVEL ZERO-DISTORTION FRAGILE SCHEME FOR WATERMARKING TEXTUAL RELATIONAL DATA

Ali Hamadou[1*], Abdoul Aziz Issaka Hassane[1], Lanciné Camara[2] and Harouna Naroua[3]

[1]Department of Mathematics, Dan Dicko Dankoulodo University of Maradi, Maradi, Niger
[2]Social Science and Management University of Bamako, Bamako, Mali
[3]Department of Mathematics and Computer Science, Abdou Moumouni University, Niamey, Niger

ARTICLE INFO

Article History:
Received 20th June, 2025
Received in revised form
24st July, 2025
Accepted 29th August, 2025
Published online 27th September, 2025

Keywords:

Fragile watermarking; Zero-Distortion; Textual Relational Data; Authentication.

***Corresponding author:** *Ali Hamadou*

ABSTRACT

The protection of textual relational data integrity is a great concern in database watermarking. Existing zero-distortion schemes are mostly designed for numeric data. This paper presents a novel distortion-free watermarking method for authentication of non-numeric relational data. For the sake of efficiency in tamper detection and localization, the database relation is horizontally partitioned into groups of average equal size. The watermark embedding process is performed in each group independently by securely modifying the case of some attributes, while preserving data integrity. Through experiments with real-world data, we demonstrate that our scheme achieves 100% tamper detection rates.

Citation: Ali Hamadou, Abdoul Aziz Issaka Hassane, Lanciné Camara and Harouna Naroua. 2025. *"*A Novel Zero-Distortion Fragile Scheme for Watermarking Textual Relational Data". *International Journal of Current Research, 17, (09), 34579-34584.*

INTRODUCTION

With the rapid growth of the Internet and related technologies such as Artificial Intelligence applications, digital data are subject to theft, unauthorized copying, and malicious modification, andillegal redistribution. Digital watermarking has become a promising method to tackle the above-mentioned issues. Initially proposed for multimedia data, watermarking wasextended later on to relational data(1-5).There are two main applications of watermarking: copyright protection and data authentication. Digital watermarking deploys information hiding to conceal a "piece of secret information" (watermark) inside digital data for the purpose of ownership proofing or data integrity control. The soundness of such a method relies on the assumption that altering the digital data in the process of hiding the watermark does not destroy the data usefulness. However, errors introduced to data during the watermark embedding process inevitably affect data usability. In addition, in text-based relational database systems, such permanent distortions are not desirable, as they do not preserve data semantic. Sometimes, even a change of a single bit will change the meaning of data and thus affect query results. Research efforts in distortion-free watermarking for textual databaseshave mainly focused on copyright protection and owner proofing (1,2,6-12). Therefore, these schemes cannot be used for data authentication. A number of fragile distortion-free schemes have been proposed in the literature for integrity control. The main limitations of these approaches can be summarized as follows: (i) they mostly focused on numeric data; (ii) somerely on adding watermark as extra attributes or tuples which not only enlarge the size of the database, but also increase the probability to destroy the watermark by an attacker; (iii)other techniques are based on tuples permutation and abstract interpretation framework which require much effort to verify the watermark.

All these factors lead to the need of authentication schemes which not only retain the semantic value of data but also preserve query results after watermark embedding. In this paper, we extend our work in (6)[1]to deal with the integrity control of text-based relational database. The watermark embedding process is performed by changing the case of some textual attributes, while preserving their semantic and usability. In this approach, uppercase and lowercase are applied to single word attributes. For multiword data, we use sentence case and title case. Sentence case and title case are both mixed-case styles as they involve both upper and lowercase letters. In sentence case, only the first letters of the first word and proper nouns are in uppercase. By contrast, title case has major words in uppercase and minor words in lowercase unless they are the first or last word of a title. The

[1] This work has focused on copyright protection

main features of the proposed method can be summarized as follows:

- **Distortion-free embedding:** It does not introduce any distortion to the original data, thus preserving usability;
- **Semantic preserving:** By only adjusting the case of textual data, it avoids syntactic/semantic distortions, thus preserving query results;
- **Granular tamper localization:** It detects tampering at the group level, narrowing down malicious modifications;
- **Attack characterization:** It provides a mechanism that distinguishes between attack types;
- **Blindness:** The original database relation is not required for watermark detection;
- **Security**: Watermark embedding and detection are performed using one-way hash functions and a secret key known only to the data owner.

The rest of this paper is organized as follows. In Section 2, related work is discussed. In Section 3, the watermark embedding and detection procedures are described in detail. Section 4 discusses key features of our scheme. Experimental results are presented in Section 4. The conclusion is given in Section 5.

Related work: Distortion-free watermarking is widely applied in sensitive data applications including financial, medical and military systems. In this section, the most relevant well-known previous works are discussed. In (13), a distortion-free fragile watermarking method is proposed to detect and localize malicious alterations made to a database relation with categorical attributes. The database tuples are first partitioned into groups based on some secure parameters. Then, watermarks are embedded in each group by changing the order of tuples. The distortion-free schemes proposed in (14-19) are also based on tuples grouping and reordering as in (13). In (20), a distortion free watermarking method based on the abstract interpretation frame work is introduced. This watermarking technique is partition based. Instead of inserting the watermark directly to the database partition, the main idea is to generate a binary image of the partition as a watermark that serves for ownership proofing as well as tamper detection. Another approach based on the abstract interpretation framework is proposed in (21).In this method, tuples are first grouped, then a fixed number of most significant bits (MSBs) and least significant bits (LSBs) of a numeric attribute are used to generate the watermark. The verification of integrity is performed by using a public zero-distortion authentication mechanism.

In (22-23), the authors proposed persistent zero-distortion watermarking solutions for copyright protection and integrity verification. Two watermarks are embedded: a private (robust) watermark and a public (fragile) watermark. The persistency of the watermarks is preserved by exploiting some invariants of the database state, namely the stable cells, the abstract database and the semantics-based properties. The fragile distortion-free watermarking technique introduced in (24) is based on an attribute reordering method. First, a secret initial order of attributes is defined by virtually sorting the attributes based on the hash of attribute names. Thereafter, the watermark, computed from data content, is registered with a Certificate Authority (CA). As in (24), the approach proposed in (25-27), the watermark is also generated from the original database and registered in a trusted third party (CA). In (25), the database relation is partitioned into independent square matrix groups and the integrity verification is performed by computing the determinant and the diagonal's minor for each group. The watermarking scheme in (26) generates the watermark based on the local characteristics like frequency distribution of various digits, lengths, and ranges of data values. In (27), the proposed approach ensures that integrity of database can be preserved by generating an image watermark from its content. The distortion-free scheme in (28) has been proposed for content integrity and ownership control of numeric and non-numeric attributes. A hash function and a genetic algorithm are used to generate fake tuples that are stored in a separatefile as watermark.

In (29), authors adapted the Map Reduce paradigm to deal with the issue of watermarking of large relational datasets. The proposed distortion-free algorithm exploits the benefits of parallel and distributed computing environment to decrease the computational cost of watermark generation. In (30), a distortion-free fragile watermarking scheme has been proposed for columnar database architecture. In this method, each attribute is organized into groups and the data elements in each group are re-ordered based on a watermark value. In (31), the authors proposed watermarking as a service with anonymity for database integrity using zero-watermarking paradigm. In this scheme, the data elements are grouped and the group watermark is generated by extracting MSBs of the hash of attribute names. The proposed technique does not add any extra information, and neither any physical change is made to the database content.

Recently, in (32), a semi-fragile watermarking scheme for relational database integrity verification is presented. Besides detection and localization of database tampering, the proposed scheme allows modifications to the data that need periodic updates, without requiring re-watermarking. Although this technique is claimed to be distortion-free, it does not preserve the semantic integrity of acronyms and proper nouns. Moreover, it cannot detect empty-group attacks.

Proposed approach

This section presents the proposed distortion-free watermarking scheme for authentication of textual relational database. Let $R(P_k, A_0, A_1, \dots, A_{\gamma-1})$ be the database relation to watermark where P_k is the primary key and $A_0, A_1, \dots, A_{\gamma-1}$ are other attributes (numeric and non-numeric). Let $\alpha(\alpha \leq \gamma)$ be the number of textual attributes candidate for watermarking. The notations and parameters used in this paper are shown in Table 1. The proposed method consists of three main parts: (i) tuple grouping; (ii) watermark generation and embedding, and (iii) watermark detection and integrity checking.

Tuple grouping: The database partitioning is a virtual operation that enforces the correlation between tuples. The grouping technique is presented in Algorithm 1. The relation R is securely divided based on the primary key hash values of tuples. As a result, this partitioning method will create groups of roughly equal size $\approx \frac{}{}$. For the hashing purpose, a message authentication code (MAC) is computed on a cryptographic hash function (e.g. SHA-256) concatenated with the secret key in other to prevent an unauthorized person to reproduce valid hash codes from a suspicious relation .Furthermore, it is difficult for an attacker to predict the tuples-

to-group assignment without the knowledge of the secret key K_s and the number g of partitions which are kept secret. Note that the tuples grouping and sorting are virtual operations. The watermark embedding and detection are performed for each group independently.

Table 1. Notations and parameters

Symbol	Description
η	Number of tuples in the relation
γ	Number of attributes in the relation
$t_i.A_j$	j^{th} attribute of the i^{th} tuple
$t_i.P_k$	Primary key of i^{th} tuple
h_i	Hash value of i^{th} tuple's primary key
H_i	Hash value of i^{th} tuple
H	Group hash
K_s	Secret key
g	Number of groups
□	Average size of an original group
□	Current size of a suspicious group
□□	j^{th} original group
□□′	j^{th} watermarked group
□□*	Suspicious group
W	Embedded watermark
W^*	Extracted watermark from a suspicious group
α	Number of textual attributes
HMAC	Hash-based Message Authentication Code

Algorithm 1 : Tuple grouping

Input: Relation R, secret key K_s, total tuples η, number of groups g
Output: Groups $G_0, G_2,\ldots,G_{g-1}$
//Create groups of average size $v = \frac{\eta}{g}$
for each tuple $t_i$in R**do**
h_i = HMAC $(K_s||t_i.P_k||K_s)$ // Hash primary key
group index$j = h_i \ mod \ g$
Add tuple t_i into group G_j
end for
Sort all tuples in increasing order of primary keys hash
return $G_0, G_2,\ldots,G_{g-1}$

Watermark generation and embedding: The proposed method provides a suitable distortion-free and semantic-preserving watermarking solution for textual relational data. The scheme can also be applied to the integrity protection of databases with mixed data types. As shown in Algorithm 3, the watermark embedding process is performed in each group independently by securely modifying the case of some attributes, while preserving data semantic and usability. In line 1, Algorithm 2 is used to generate a watermark from data of all attributes. Afterwards, for each tuple in the group, a cyclic watermark bit b is securely encoded in a modifiable non-numeric attribute as follows. On one hand, if the selected attribute value $t_i.A_j$ is a single word and the watermark bit b is 1, then $t_i.A_j$ is set to lowercase; it is set to uppercase if b is 0. On another hand, if $t_i.A_j$is a multiword, it is set to title case or sentence case depending on the associated watermark bit. Note that acronyms, proper nouns and numeric data are skipped during the embedding process so as to preserve their integrity. Notice also that the number α of textual attributes available for watermarking is a secret parameter known only to the data owner.

Algorithm 2: Watermark generation

Input: GroupG_j, secret key K_s
Output: Watermark W
foreachtuple$t_i$in G_j**do**
H_i = HMAC$(K_s||t_i.A_0||t_i.A_1||\ldots||\ t_i.A\gamma\text{-}_1 \ ||\ K_s)$ //i^{th} tuple hash
end for
H = HMAC$(K_s \ ||\ H_0\ ||H_1\ ||\ldots||\ H_{v\text{-}1}\ ||\ K_s)$ //Group hash
W= v MSBs from H//Assume $|H| \geq v$
return$W$

Algorithm 3: Watermark embedding

Input: GroupG_j, secret key K_s, textual attributes$A_0,A_1,\ldots,A_{\alpha\text{-}1}$
Output: Watermarked group G_j'
$W = GenWM(G_j,K_s)$//Algorithm 2 (watermark generation)
foreachtuple$t_i$in G_j**do**
$j = h_i$mod α// Select attribute
$b = W((i+j))$mod v // Select cyclic watermark bit
if ($t_i.A_j$contains acronym or proper noun) **then**
skip // Preserve semantic of acronyms and proper nouns
if ($t_i.A_j$is single word) **then**
if (b == 1) **then**
set $t_i.A_j$to *lowercase*
else
set $t_i.A_j$ to *uppercase*
end if
end if
if ($t_i.A_j$is multiword) **then**
if (b == 1) **then**
set $t_i.A_j$ to *title case*
else
set $t_i.A_j$to *sentence case*
end if
end if
end for
return G_j'

Watermark detection and integrity checking: The watermark detection is the process of decoding the embedded watermark bits from the suspicious database. Our detection approach can detect and characterize various types of attacks at the group level. To handle edge cases, Algorithm 4 first checks for empty groups. If a group is empty, it is flagged as tampered as shown in lines 1 to 7. Tampering caused by insertion and deletion attack scan be easily detected by simply comparing the current size of the suspicious group with its relevant expected size. To detect the content tampering (case/numeric/acronymchanges), we first exytract a watermark from the suspicious group as shown in Algorithm 5. Next, the corresponding original watermark is computed as in the embedding phase. Ifthe two watermarks match, the group is authentic, otherwise it has been tampered.

Algorithm 4: Tamper detection and characterization

Input: Suspicious groupG_j^*, secret key K_s, total tuples η, number of groups g, α
Output: "Tampered: attack type" or "Authentic"
1. $v = \left\lceil \frac{\eta}{g} \right\rceil$ //Expected average group size as in embedding
2. $s = \left| G_j^* \right|$ //Current size of G_j^*
3. if $(s == 0)$ then
4. if $(v > 0)$ then
5. return" Tampered: Empty-group attack" //Mass deletion
6. endif
7. endif
8. if $(s > v)$ then
9. return "Tampered: Tuple insertion attack"
10. else
11. if $(s < v)$ then
12. return "Tampered: Tuple deletion attack"
13. endif
14. endif
15. //Content tampering control (case/numeric/acronym changes)
16. $W^* = ExtractWM(G_j^*, \alpha)$ //Extracted watermark (Algorithm 5)
17. $W = GenWM(G_j^*, K_s)$ // Original watermark (Algorithm 2)
18. if$W \neq W^*$then
19. return "Tampered: Content Altered"
20. else
21. return "Authentic"
22. end if

Algorithm 5: Watermark extraction

Input: Suspicious group G_j^*, total textual attributes α
Output: W^*
1.foreachtuple t_i inG_j^* do
2.$j = h_i$mod α //Select textual attribute
3.if (t_i. A_j contains acronym or proper noun) then
4.skip // Value ignored during embedding
5.if ($t_i.A_j$is single word) then
6. if (case of $t_i.A_j$is *lowercase*) then
7.$W^*[i] = 1$
8.else
9.if (case of $t_i.A_j$is *uppercase*) then
10.$W^*[i] = 0$
11.end if
12. end if
13.if ($t_i.A_j$is multiword) then
14. if (case of $t_i.A_j$is *title case*) then
15.$W^*(i) = 1$
16.else
17.$W^*(i) = 0$
18.end if
19. end if
20.end for
21.returnW^*

DISCUSSION

In this section, we discuss some important features of our scheme.

Security analysis: The watermarked database may be subject to various attacks. The goal of an attacker is to make malicious modifications without disturbing the embedded watermark. The grouping mechanism used in our scheme relies on the secret key K_s and the number of groups g which are only known to the data owner. Without these two parameters, an attacker cannot predict group boundaries or target specific watermarked tuples and attributes.

The embedded watermark W is derived from a cryptographic hash function concatenated withK_s.Without knowledge of K_s, an attacker cannot forge a valid watermark, even with unlimited access to the watermarked relation. Moreover, the use of cyclic watermarking provides additional security benefits: (i) it makes the embedding bits unpredictable,(ii) it ensures that all bits of W are used evenly across tuples, thus preventing bias.

Due to the perfect fragility of our scheme, any attack (tuple insertion/deletion, attribute modification) affects group hashes, thus invalidating$W$and causing mismatch withthe extracted watermarkW^*. For $W^*$to match W, an attacker must:(i) guess the correctmarked attributes indices (success probability $\frac{1}{\alpha}$ per tuple), and (ii)flip attribute cases to match$W$ exactly (success probability $\frac{1}{2^v}, v = |W|$).

For instance, even with $\alpha = 1$ and $v = 100$, the probability of success is $\frac{1}{100}$.Our scheme also ensures the integrity of non-modifiable attributes (acronyms, proper nouns, numeric) without additional overhead as they are covered in the group hashing. Any case or value modification to these data will change the group hash and randomize the watermark. Furthermore, with a simple group size check, our scheme can efficiently detect, localize and characterize tuple insertion and deletion attacks.

Semantic preserving: By changing only letter casesof modifiablenon-numeric attributes, our method avoids syntactic and semantic distortions and preserves data semantic and usability. The case of non-modifiable textual attributes such as acronyms and proper nounsis not altered.

Blind detection: The proposed scheme is fully blind as it requires neither the original database nor stored hashes.For watermark verification only secret parameters K_s, η, g, and αare necessary. Furthermore, there is no need to keep a registry for non-modifiable attributes.

Experimental results: In this section, we evaluate the effectiveness of the proposed distortion-free watermarking techniquein detecting three common database attacks: tuple insertion, tuple deletion, and attribute alteration. The experiments were conducted on an Intel core i7-10510U CPU, 2.3 GHz with 16 GB RAM, using Python andpandas. We used Lahman Baseball Database, a real-world dataset with textual and numeric attributes (33). It containspitching, hitting, and fielding statistics for Major League Baseball from 1871 through 2024. The datasetis composed of 27 tables. For testing purpose, we used the People table which consists of 12numeric attributes, 12 non-numeric attributes and 21271 tuples. For watermark embedding, we set g (i.e. the number of groups) to 100to create groups of sufficient size$(v \approx 212)$to ensure a large watermark length while maintaining fine-grained localization. Since our scheme is strictly fragile, any modification (even a single bit change) made to the watermarked database should be detected as tampering. For each kind of attack,we randomly changed the group size and varied the attack ratio up to 90%. We measured the efficiency of the scheme in terms oftamper detection rate. Experiments were performed repeatedly and the results were averaged over multiple runs.

Tuple insertion attack: For testing this attack, we progressively and randomly inserted fake tuples in the database table. This operation increased the size *v* of all affected groups. Consequently, we noticed that the relevant watermarks were not correctly extracted and the tampering was fully detected as shown in Figure 1.

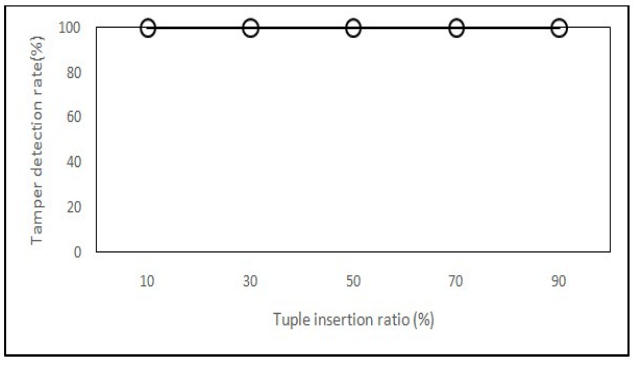

Figure 1.Tamper detection for tuple insertion attack

Tuple deletion attack: In this simulation, several tuples were randomly and gradually deleted from the database. Figure 2 shows the resilience of our scheme against this attack. We can see that the detection rate is 100% even when90% of tuples are deleted.

Attribute alteration attack: Two different experiments were performed for this attack: (i) value alteration, and (ii) case-flipping in textual data.In the first case, we randomly and

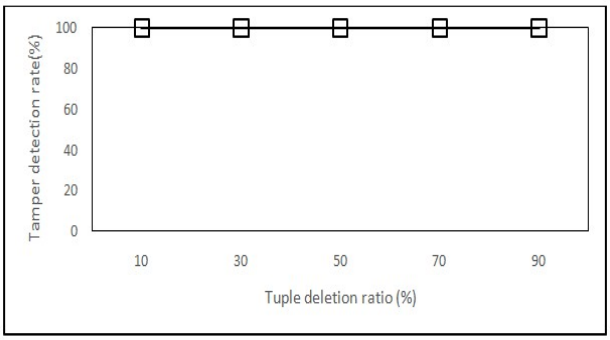

Figure 2. Tamper detection for tuple deletion attack

repeatedly modified the values of some attributes. In the second experiment, the case of several attributes was arbitrarily flipped. We observed that all modifications were perfectly detected and localized at group level. This result confirms the strict fragility of our scheme. Figure 3 shows the outcome for attribute alteration attack.

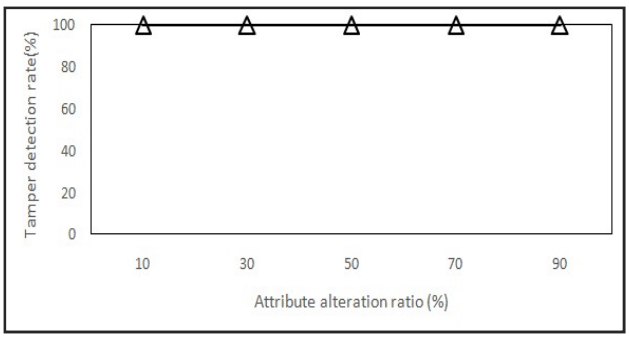

Figure 3. Tamper detection for attribute alteration attack

CONCLUSION

In this paper, we introduced a novel zero-distortion fragile watermarking method for tamper detection and authentication of relational database. Although, the proposed scheme is primarily designed for textual data, it can also be applied to databases with mixed data types. The watermark is constructed from data (numeric and non-numeric) content and encoded by securely changing the case of some non-numeric values. Through simulations using real-world data, we demonstrated that the propsed scheme is efficient in detecting usual attacks. In the future, we intend to explore a multi-purpose solution that will incorporate both copyright and integrity protection of text-based data.

REFERENCES

1. Rani, S., and Raju, H. Comparative analysis of relational database watermarking techniques: An empirical study. IEEE Access, 10, 27970-27989, 2022
2. Kamran, Muhammad & Farooq, Muddassar, ”A Comprehensive Survey of Watermarking Relational Databases Research”, arXiv:1801.08271v1 (cs.CR) 25 Jan 2018
3. Zhenzhe Gao, Yu Cheng, Zhaoxia Yin, A survey of fragile model watermarking, Signal Processing, Volume 238, 2026, 110088, ISSN 0165-1684
4. Hamadou, A. , Hassane, A. , Camara, L. and Naroua, H. Reversible Semi-Fragile Watermarking Technique for Integrity Control of Relational Database. *Engineering*, **16**, 2024, 309-323.
5. Hamadou, A., Camara, L., Issaka Hassane, A. A., & Naroua, H. Reversible fragile watermarking scheme for relational database based on prediction‐error expansion. *Mathematical Problems in Engineering*, 2020(1), 1740205.
6. Shah, S. A., Sun Xingming, Hamadou Ali, and Majid Abdul. "Query preserving relational database watermarking." Informatica 35, no. 3 (2011).
7. Zhang, Y. Z. Wang, Z. Wang, and C. Liu, ``A robust and adaptive watermarking technique for relational database," in *Proc. 18th China Annu.Conf.* Singapore: Springer, Jul. 2021, pp. 3-26.
8. Melkundi S. and C. Chandankhede, ``A robust technique for relationaldatabase watermarking and veri_cation," in *Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT)*, Jan. 2015, pp. 1-7.
9. Chang, C.-C. T.-S. Nguyen, and C.-C. Lin, ``A blind robust reversible watermark scheme for textual relational databases with virtual primarykey," in *Proc. Int. Workshop Digit. Watermarking*. Cham, Switzerland: Springer, 2014, pp. 75-89.
10. Zhang, L. W. Gao, N. Jiang, L. Zhang, and Y. Zhang, ``Relational databases watermarking for textual and numerical data," in *Proc.Int. Conf. Mech. Sci., Electric Eng. Comput. (MEC)*, Aug. 2011,pp. 1633-1636.
11. Hanyurwimfura, D., Liu, Y., Liu, Z.: Text format based relational database watermarking for non-numeric data. In: 2010 International Conference on Computer Design and Applications, vol. 4, pp. V4-312–V4-316. IEEE (2010)
12. Melkundi, S., Chandankhede, C.: A robust technique for relational database watermarking and verification. In: 2015 International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1–7. IEEE (2015)
13. Li, Y. H. Guo, and S. Jajodia, ``Tamper detection and localization for categorical data using fragile watermarks," in *Proc. 4th ACM Workshop Digit. Rights Manage.*, 2004, pp. 73-82.
14. Bhattacharya S. and A. Cortesi, ``A distortion free watermark framework for relational databases," in *Proc. ICSOFT*, 2009, pp. 229-234.
15. Kamel, I. ``A schema for protecting the integrity of databases," Comput. Secur., vol. 28, no. 7, pp. 698709, Oct. 2009.
16. Li, M. W. Zhao, and J. Guo, ``An asymmetric watermarking scheme forrelational database," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, May 2011, pp. 180-184.
17. Arun, R. K. Praveen, D. C. Bose, and H. V. Nath, ``A distortion free relational database watermarking using patch work method," in *Proc. Int. Conf. Inf. Syst. Design Intell. Appl.*, Visakhapatnam, India. Berlin, Germany: Springer, Jan. 2012, pp. 531-538.
18. Kamel, I. M. Alaa Eddin, W. Yaqub, and K. Kamel, ``Distortion-freefragile watermark for relational databases," *Int. J. Big Data Intell.*, vol. 3,no. 3, pp. 190-201, 2016.
19. Lancine Camara, Demba Coulibaly, Ali Hamadou and Junyi Li, An Effective Approach for Non-Numeric Relational Database Verification. International Journal of

Database Theory and Application Vol.10, No.6 (2017), pp.35-46
20. Bhattacharya S. and A. Cortesi, ``A generic distortion free watermarking technique for relational databases," in *Proc. Int. Conf. Inf. Syst. Secur.* Berlin, Germany: Springer, 2009, pp. 252-264.
21. Bhattacharya S. and A. Cortesi, ``Distortion-free authentication watermarking," in *Proc. Int. Conf. Softw. Data Technol.* Berlin, Germany: Springer, 2010, pp. 205-219.
22. Halder R. and A. Cortesi, ``A persistent public watermarking of relational databases," in Proc. Int. Conf. Inf. Syst. Secur. Berlin, Germany: Springer,2010, pp. 216-230.
23. Halder R. and A. Cortesi, ``Persistent watermarking of relational databases," in *Proc. IEEE Int. Conf. Adv. Commun., Netw., Comput.(CNC)*, 2010, pp. 46-52.
24. Hamadou, A. X. Sun, and L. Gao, ``A fragile zero-watermarking technique for authentication of relational databases," *Int. J. Digit. Content Technol. Appl.*, vol. 5, no. 5, pp. 189-200, May 2011.
25. Camara, L. J. Li, R. Li, and W. Xie, ``Distortion-free watermarking approach for relational database integrity checking," Math. ProblemsEng., vol. 2014, Dec. 2014, Art. no. 697165.
26. Khan A. and S. A. Husain, ``A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations, "*Sci. World J.*, vol. 2013, pp. 1-16, Oct. 2013.
27. Siledar S. and S. Tamane, ``A distortion-free watermarking approach for verifying integrity of relational databases," in Proc. Int. Conf. Smart Innov. Design, Environ., Manage., Planning Comput. (ICSIDEMPC), Oct. 2020, pp. 192195.
28. Darwish, S. M. ``Distortion free database watermarking system basedon intelligent mechanism for content integrity and ownership control,"*J. Comput.*, vol. 4, pp. 1053-1066, Dec. 2018.
29. Rani, S. D. K. Koshley, and R. Halder, ``Adapting mapreduce for efficient watermarking of large relational dataset," in *Proc. Trustcom/Big Data SE/ICESS*,2017, pp. 729-736.
30. Waheeb Yaqub, Ibrahim Kamel, and Zeyar Aung. "Distortion-free watermarking scheme for compressed data in columnar database." Proceedings of the 15th ICETE. porto, portugal (2018): 343-353.
31. Naz, F. A. Khan, M. Ahmed, M. I. Khan, S. Din, A. Ahmad, and G. Jeon, ``Watermarking as a service (WAAS) with anonymity," *Multimedia Tools Appl.*, vol. 79 no. 23, pp. 16051-16075, 2020.
32. Shah, S. A. I. A. Khan, S. Z. H. Kazmi, and F. H. B. M. Nasaruddin, "Semi-fragile watermarking scheme for relational database tamper detection, "*Malaysian J. Comput. Sci.*, vol. 34, no. 1, pp. 1-12, Oct. 2021.
33. Lahman Baseball Database. https://sabr.org/lahman-database/
34. Ferguson N. and B. Schneider, Practical Cryptography. Wiley & Sons, 2003.

****