International Journal of Database Theory and Application
Vol.10, No.6 (2017), pp.35-46
http://dx.doi.org/10.14257/ijdta.2017.10.6.03

An Effective Approach for Non-Numeric Relational Database Verification

Lancine Camara[1,3], Demba Coulibaly[1], Ali Hamadou[2] and Junyi Li[3*]

[1]*Social Sciences and Management University of Bamako, Bamako, Mali*
[2] *Dan Dicko Dankoulodo University of Maradi, Maradi, Niger*
[3]*College of Information Sciences and Engineering Hunan University Changsha, China*
[1,3]*lcamaralancine@yahoo.fr,* [1]*demcoul@univ-bamako.edu.ml,*
[2,3]*ali.hamadou@uddm.edu.ne,* [3]*junyilee@hnu.edu.cn*

Abstract

With the large distribution of digital data, protecting their integrity becomes necessary and digital watermarking has been proposed as solution for protecting the content of relational database. Previous watermarking techniques mainly focus on the numeric database authentication by inserting watermark bits in digital data which may greatly degrade the data quality. In this paper, we present a distortion free approach to verify the integrity of a combined numeric and non-numeric relational database. The technique first partitions the database in different groups of square matrices, then the ASCII code of non-numeric data of group attributes are computed and used to generate the watermark. Security analyzes and experiments demonstrated that the proposed technique is resilient against malicious attacks and moreover the tampering can be detected up to group level.

Keywords*: Database Watermarking, Integrity Verification*

1. Introduction

Nowadays databases are used in every modern organization to record the state of the organization. With the development of information technology, it is becoming easier to copy or share critical data. Such data may be subject to various attacks with the aim of false data ownership claim or to maliciously alter the database. Attacking data may produce disastrous results. In order to protect the data content against malicious tampering or copyright theft, watermark techniques have been used. Commonly, watermarking techniques can be classified in two groups: the watermarking for integrity verification or tamper proofing which uses fragile watermarking and the watermarking for copyright protection and ownership proofing which uses robust watermarking scheme. In robust scheme, the watermark should resist against attacks with the aim to remove or destroy the watermark, in [1], Xie *et al.* proposed a survey in distortion watermarking technique, they mainly focus on copyright protection approach.

In general, watermarking process should be invisible and should not degrade the data quality (imperceptibility), it should use the secret key during the embedding and the decoding process (security), its detection process should not use original data and watermark (blind), and the data owner should be able to detect the watermark. In practice, watermarked database may be subject to malicious attacks with the aim of modifying the database and leaving the watermark intact. Furthermore, the watermarking can also be used for the recovery of the database information in case of data loss [2].

This paper presents a group-based distortion free watermarking technique for tampering detection in relational database without caring about the data type. Generally, fragile watermark techniques suffer from the quantity of embedded information during the

ISSN: 2005-4270 IJDTA

process because the usability of data should be preserved. To tackle the above issue two approaches have been used: first the distortion free approach which does not modify any data from the database and second, the usability constraint approach which uses embedded information (watermark) from the original database.

In this paper, we have accomplished the following contributions:

(I) We have developed a distortion free partition based approach to verify the content integrity of both numeric and non numeric relational database. As the technique does not modify the database, there is no care about the data usability constraints.

(II) An ASCII code technique is used to encode non numeric data into numeric data and applied our previous approach for data integrity verification.

(III) The algorithm that can detect and localize the tampering made to database up to group level.

(IV) The experiments have been conducted to show the technique effectiveness.

This paper is organized as it follows: the next section reviews the related works. In Section 3, the proposed approach is explained. In Section 4 the performance of our proposed approach is evaluated. The Section 5 analyzes the effectiveness of our approach. Finally, the Section 6 concludes our paper and provides some guidance for future work.

2. Related Work

The first relational database watermarking well known work was proposed by Agrawal and Kiernan [3]. Their technique was designed to be robust scheme. They identified the least significant bits (LSB) of selected attributes of some selected tuples to embed the watermark. Inserting watermark bits in LSB is inefficient because the watermark can be easily compromised by bits attacks and embedding using LSB may degrade the data quality [4]. Li et Deng [5] have presented a tuples positions exchange approach for proving the ownership publicly using the linear permutation unranking algorithm proposed by Myrvold *et al.* in [6]. In [7] Schachtner *et al.* proposed a determinant criterion approach for finding the exact solution by to constraining the solutions of non-negative matrix factorization problems. Guo *et al.* [8] proposed a fragile approach based on LSB modification which may degrade the data quality. In [9], Bhattacharya *et al.* presented a zero distortion watermarking technique independently to attributes type to verify the integrity of the relational databases based on the Abstract Interpretation framework. The partitioning technique they used can be seen as a virtual grouping operation which generates image of the partition as a watermark of that partition that serve as ownership proof watermark as well as tamper detection. Bedi *et al.* [10] proposed a technique to verify the integrity of non-numeric database based on eigen values concept. In [11], we have presented a distortion free based on numeric data type. The technique first partitions the database into independent groups of matrix square. Then, group-based watermarks are securely generated and registered in a trusted third party. The integrity verification is performed by computing the determinant and the diagonal's minor for each group. As a result, tampering can be localized up to attribute group level. This paper is an extension of our work [11] to non-numeric database furthermore, instead of using the third party, our algorithm generated, embeds, and detects the watermark. I. Kamel [12] work is a distortion free fragile scheme for protecting the numeric database integrity. The technique is based on R-tree data structure that does not change the attribute. Lingyun *et al.* [13] proposed a novel tamper recovery fragile watermarking technique for relational databases. Their scheme is group based approach, the watermark is embedded and verified group-by-group independently and Reed-Solomon coding technique is used to embed the watermark.

3. Our Algorithm

Some notations and parameters used in this paper are showed in the Table 1. In this section, the proposed approach is explained with meaningful understanding. The following parts are covered: The watermark bits generation is first explained afterwards, the database partitioning and the watermark embedding process are covered, and finally the watermark detection is depicted.

Table 1. Notations and Parameters

Symbol	Description
α	Number of tuples in database
γ	Number of attributes in database
$r_i P_i$	Primary Key row
k	Secret key
G_γ^j	j^{th} group
l	Watermark length
$D=(D_i)_{1\le i\le j}$	Set of groups determinant
R_W	Watermarked relation

3.1. Watermark Key K Generation

In this paper, the coordinated universal time (UTC) which represents the primary standard time used for the time over the world synchronization [14] date time can be used as watermark to generate the watermark bits used in the encoding process input. As the UTC is a set of 29 bits, the watermark will consist of a string length of 29 bits.

3.2. Database Partitioning

This section describes the partitioning algorithm. Algorithm 1 shows the parsing process used in the paper. The proposed partitioning algorithm is the partitioning technique used in [15] .It partitions the database R composed of α tuples and γ attributes to v different groups.

Algorithm 1: Database partitioning

Input : Database R, Number of groups $v=\left\lceil \frac{\alpha}{\gamma} \right\rceil$, and Secret key K_S

Output: v groups $(G_1,...,G_v)$ of length γ each

1. **Begin**
2. **for** $i=1$ 재? α **do**
3. $h_i^r = Hash(k_S \| r_i.P \| k_S)$ // i^{th} row primary key hash
4. $j = h_i \bmod v$ // group index
5. insert r_i into G_j
6. Sort all tuples in G_j according to the increasing order of their

primary key hash
7. **end for**
8. **return** $(G_1,..,G_\nu)$
9. **end.**

The partitioning is based on a primary key attribute P of each tuple and a secret key K, a secure hash function is computed for each tuple and γ tuples are inserted logically into individual partition described as the following:

$$G_{j(1\le j\le\gamma)} \text{ and } G_i \neq G_j \quad for \quad i \neq j$$
$$j = \text{Hash}(\text{k}\|\text{r}_i.\text{P}\|\text{k}) \bmod \nu \tag{1}$$

As a result, the database R is partitioned into ν different groups $\{G_1,...,G_\nu\}$.

In the case where $\alpha \bmod \gamma \neq 0$, we simply securely insert records to complete the last group in order to get a square matrix and we make sure there is no identical tuple in the same group. Note that the added records will be deleted after the watermark insertion

3.3. Watermark Embedding

The watermark embedding algorithm embeds single watermark bit in each database partition to obtain the database group determinant value. It takes as input the database to be protected, the watermark bits to be inserted, and the secret key is known only by the database owner to compute the watermarked database partitions determinant.

Algorithm 2: Watermark embedding

Input: Database R, Watermark $W=\{b_1,...,b_{l-1}\}$, Secret key k

Output: Watermarked database R_W, Marked group determinant D

1. Begin
2. initialization: $R_W \leftarrow \{\ \}$, $D \leftarrow \{\ \}$
3. database partitioning into groups G_j // see algorithm 1
4. for $j=1$ *to* ν
5. embed single bit b_i to G_j // see algorithm 3
6. Compute and set to ASCII Sum the value of all non numeric attribute in G_j
7. append D_j *to* D
8. append $G_\gamma^{j'}$ *to* R_W
9. end for
10. Return R_W, D
11. end.

In line 3, the embedding algorithm used algorithm 1 to partition the database R into ν different groups afterwards, the algorithm 3 is used to embed a single b_i to each group. In line 5 the algorithm looks into group, finds non-numeric data values, converts and sets them in numeric data using their corresponding ASCII value. The group determinant D_j is computed and logged to file. In line 7, the computed $(D_j)1\le j\le\nu$ are collected to form the marked group determinant D used in the watermark decoding process. In line 8, the watermarked group is collected to form the watermarked database. Finally the

watermark embedding process ends at line 10 and returns the set of marked group determinant D.

3.4. Group Watermark Embedding Process

Algorithm 3: Group Watermark embedding
Input: G_γ^j, $W = b_1, ..., b_{l-1}$
Output: $G_\gamma^{j'}$, D_j
1. Begin
2. **for** each partition G_γ^j // j^{th} group
3. compute b_i with $i = j \bmod l$ // watermark index to be embedded
4 . **if** $(b_i == 0)$
5. $G_\gamma^{j'} = unrank(\gamma, j, \pi)$ // unranking function for a permutation π
6. Compute D_j
7. else
8. $G_\gamma^{j'} = unrank(\gamma, j+1, \pi)$ // unranking function for a permutation π
9. Compute D_j
10. end if
11. return $G_\gamma^{j'}$, D_j
12. Function $unrank(k, r, \pi)$
13. if $(k \succ 0)$ then
14. $swap(\pi[k-1], \pi[r \bmod k])$
15. $unrank(k-1, \lfloor r/k \rfloor, \pi)$
16. end if

This step shows the watermark embedding process at an individual group level. The embedding algorithm takes as input the database groups and the watermark bits to be inserted. The watermark insertion algorithm is processed for each group and processed as the following. In line 3, the watermark bit i to be inserted in the j^{th} partition is computed. Note that the embedding function does not modify the database, it just virtually exchanges tuples position. From line 4 to 5 if the bit b_i to be inserted is 0, the $unrank$ function for a permutation π takes as input the database attribute γ and the group index j to sort the group tuples in a certain order then, the group determinant is computed and logged. If the bit b_i is not 0 the line 7 is processed, the $unrank$ function for a permutation π takes as input the database attribute γ and the group index j increase by 1 ($j+1$) to sort the group tuples in a certain order. Finally, the group processing function ends at line 10 and it returns the watermarked group $G_\gamma^{j'}$. The $unrank$ function for a permutation π behavior is described from line 11 to 14.

3.5. Watermark Detection Process

Algorithm 4: Watermark detection

Input: R_W, D, and watermark length l

Output: detected watermark W_i

1. Begin
2. set $W_i[0,...,l-1] \leftarrow 0$
3. R_W partitioning into groups $G_\gamma^{j''}$ // see algorithm 1
4. for each partition $G_\gamma^{j''}$
5. Compute and set to ASCII Sum value of all non numeric attribute in $G_\gamma^{j''}$
5. for $k = 0,..., l-1$
6. $unrank(\gamma, j, \pi)$ // unranking function for a permutation π
7. Compute $D_j^{'}$ // j^{th} determinant group
8. if $D_j = D_j^{'}$
9. $W_i(temp) \leftarrow 0$
10. else if
11. $unrank(\gamma, j+1, \pi)$ // unranking function for a permutation π
12. Compute $D_j^{'}$ // j^{th} determinant group
13. if $D_j = D_j^{'}$
14. $W_i(temp) \leftarrow 1$
15. else
16. $W_i(temp) \leftarrow F$
17. end if
18. end for
19. $W_i \leftarrow \{b_1^{'},...,b_{l-1}^{'}\}$
20. end for
21. return W_i.

The watermark decoding is the process of extracting the embedded watermark bits from the suspicious database $R^{'}$ by having as input the suspicious database $R^{'}$, the set of marked group determinant D, the number of group (the knowing of number of group is not a requirement), and watermark length l. The different steps of the watermarking decoding are displayed in algorithm 4. The decoding algorithm first uses the algorithm 1 to partition R_W into ν different partitions. The algorithm used the partition index (j) for partition tuples sorting and executing the *unrank* function for a permutation π afterwards, all non-numeric data from the group are converted and set to their corresponding ASCII value.

The group determinant value is then computed and compared with an expected value. If the two determinant values match, the extracted bit b_i is set as 0 but if they didn't match, the decoding algorithm increment the partition index (j +1) for tuples sorting and execute the *unrank* function for a permutation π.

Afterwards, all non-numeric data from the group are converted and set to their corresponding ASCII value. The group determinant is computed and compared with an

expected value. If the two values match the extracted bit b_i is set as 1 but if they didn't match, from line 21 the watermark is extracted as F an erasure. If the watermark bit is extracted as an erasure F means that, the watermarked group has been maliciously modified. Note that our decoding algorithm is blind because it does neither use the original database nor the embedded watermark bits during the watermark decoding process.

4. Evaluation

In this section, the experimental result of the resilience of our approach is reported. We have performed our algorithms in 2.2GHz Intel Dual CPU with 2 GB of RAM computer running Microsoft SQL Server 2008 and Myeclipse 6.6. We have used the 2013 release of Lahman Baseball dataset [16] in our experiment. The database contains pitching, hitting, and fielding statistics for Major League Baseball from 1871 through 2012. It includes data from the two current leagues (American and National), the four other "major" leagues (American Association, Union Association, Players League, and Federal League), and the National Association of 1871-1875. To verify efficiency of our technique to detect malicious modification made to the database, we consider a Managers dataset from Lahman. It consists of various data types (numeric and non-numeric data) and has 10 attributes and 3337 tuples. In this experiment, the database primary key value (playerID) is not modified, the reason is the fact that if the primary key is modified the partitions will fail to be generated and the tampering is easily proved. We first test our algorithm against tuples insertion attack, tuples deletion attack, attributes modification attack, and multifaceted attacks. Massive insertion, deletion, and even alteration are easily detected in our approach. They considerably disturb the database partitions, and then the partition determinant value is unconditionally modified. Since our technique is based on determinant value computation, the watermark cannot be detected after any modification made to the database. Therefore, we mainly focus on multifaceted attacks [17], a sophisticated attacker that generates any kind of permutation of insertion, deletion, and alteration with the aim of modifying the dataset and make the watermark untouched and detectable in the detection process.

4.1. Insertion Attack

In watermark embedding process, the watermark bit is inserted in individual partition. We have progressively inserted randomly fake tuples in the database relation. Accordingly, the same database partitions cannot be generated and the expected watermark bit will fail to be extracted, consequently the modification is detected. For massive insertion, obviously the number of partition is not a requirement in our approach but the tampering is easily detected by simply comparing the number of database partition. The Figure 6.1 shows the resilience of the proposed approach against insertion attack.

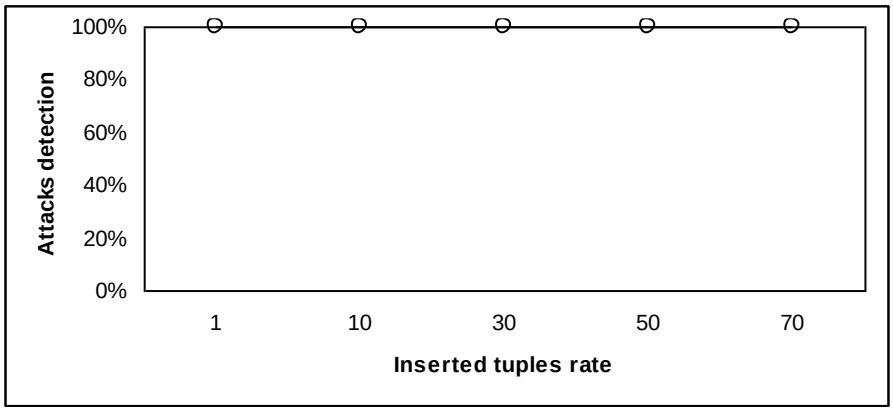

Figure 4.1. Resilience to Malicious Insertion Attacks

4.2. Deletion Attack

To test the resilience of our approach against deletion attack, the database tuples are deleted randomly and gradually from the database relation. In the proposed approach after deletion of tuples, it will affect the partition data and expected watermark bit will fail to be extracted from the suspicious database. From the Figure 4.2, the tampering is detected at 100% after database tuples deletion.

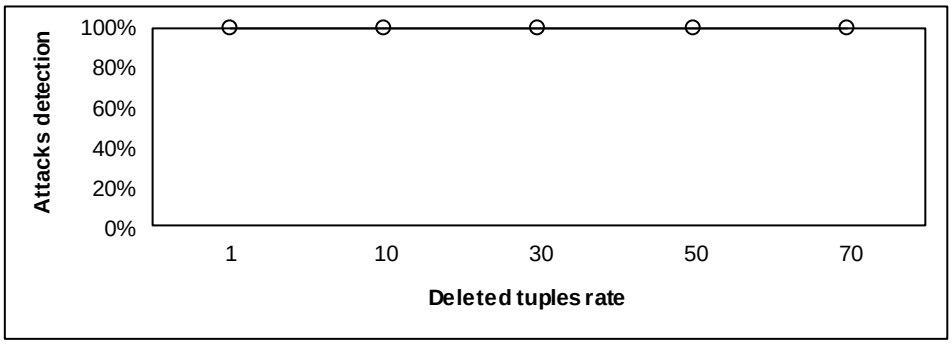

Figure 4.2. Resilience to Malicious Deletion Attacks

4.3. Alteration Attack

In alteration attacks, the database attribute values are maliciously modified.

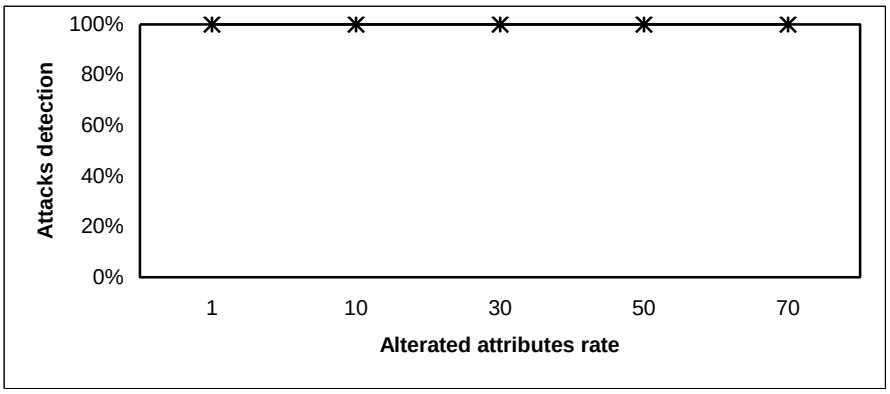

Figure 4.3. Resilience to Malicious Alteration Attacks

The attackers want to modify the database while preserving the watermark by making it detectable. In alteration attack, some tuples are progressively selected and modified. Our approach can detect even minor modification made to the database. The reason is the fact that partition determinant value and watermark bit that should be extracted from the partition are correlated. Our pilot study showed from Figure 4.3 reveals that our technique is highly resilient against alteration attacks.

4.4. Multifaceted Attack

In multifaceted attack, simultaneous deletion, insertion, and alteration attacks are performed with the aim of modifying the attribute values and make the expected watermark extracted from the suspicious database. In this attack, if the number of tuple deletion is more than tuple insertion or vice versa, the tampering is easily detected. An inform attacker may delete and insert the same number of tuples. It will furthermore try to generate the same partition and play with attribute value so that to preserve the same value of partition determinant. This attack is more challenging and our proposed technique is resilient to multifaceted attack. Since the proposed technique is based on fragile watermarking, a minor malicious modification made to the original database should be detected. An attacker to succeed the attack must first generate the partition which may be difficult without the knowledge of secret key and should modify the partition by preserving the partition determinant value which is tedious task. Our approach is resilience against multifaceted attacks.

5. Security and Analysis

The use of secret key is primordial to any good database watermarking technique. Our approach follows the above rule and the secret key is essential in database partitioning process. Moreover, in our approach two different concepts are used and both are secured. At first an individual partition determinant value is used for watermark embedding and detection process. Since our technique is fragile and malicious modifications in database is not allowed but if a modification occurs in the database, the partition determinant value will inevitably change. In our previous work [11, 15], we analyzed the success of the use of probability to change the database while keeping the watermark intact is hard to process may be impossible. By using theory of probability, the probability of successfully modifying the database by preserving its group's determinant values becomes:

$$P_{success}(S) = \left(\frac{4 * \gamma}{(\gamma^2)!} \right)_{\gamma \geq 3}^{\left(\frac{\alpha}{\gamma}\right)^n} \tag{2}$$

where γ represents the number of database attribute and α the number of database tuples, having a large database the probability of success is very small and approaches zero.

$$P_{success}(S) = \lim_{n \to \infty} \left(\frac{4 * \gamma}{(\gamma^2)!} \right)^{\left(\frac{\alpha}{\gamma}\right)^n} \rightarrow 0 \tag{3}$$

Second, to get non-numeric attributes information, we computed their ASCII code value which is unique for each character. To break that security level one solution can be the use of some characters from the attributes and modify their order to constitute a desire and understanding string which may be tedious task.

5.2. Blind Detection

Our technique is blind in the fact neither the database nor the watermark is used in the approach detection process. We only used the watermark length and the group determinant value in the watermark detection process which is a piece of information retrieved from the database groups.

5.3. Tampering Localization

The number of partitions can be considered as a secret parameter in the fact that it will make the partition generation difficult for an attacker and massive deletion and insertion are easily detected. For massive tuples deletion and insertion, the tampering is easily proved by simply computing the group number. For any slight modifications made to database, the primary key is not modified, the database group can be generated and the extracted watermark bit will indicate the index of tampered group.

6. Conclusion

In this paper, we proposed an effective distortion free approach for the detection of malicious modification made to non-numeric database. The technique is partition based and designed to be fragile, it embeds watermark bit in database partition by simply sorting the partition tuples and computing the partition determinant value. Our evaluation showed that the presented technique is highly resilient to malicious attacks made to the database. Our technique is fragile scheme. It is designed for tamper detection and cannot protect the right and the ownership of relational database.

References

[1] M. R. Xie, C. C. Wu, J. J. Shen and M. S. Hwang, "A Survey of Data Distortion Watermarking Relational Databases", International Journal of Network Security, vol. 18, no. 6, **(2016)**, pp. 1022-1033.
[2] S. Rani, P. Kachhap and R. Halder, "Data-Flow Analysis-Based Approach of Database Watermarking", Advanced Computing and Systems for Security, of the series Advances in Intelligent Systems and Computing, vol. 396, **(2015)**, pp. 153-171.
[3] J. Waleed, H. D. Jun and S. Hameed, "A robust Optimal Zero-Watermarking Technique for Secret Watermark Sharing", International Journal of Security and Its Applications, vol. 8, no. 5, **(2014)**, pp.349-360.
[4] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on Copyright Marking Systems", LNCS, vol. 1525, **(1998)**, pp. 218-238.
[5] Y. Li and R. H. Deng, "Publicly verifiable ownership protection for relational databases", in ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security. Taipei, Taiwan: ACM, **(2006)**, pp. 78–89
[6] W. J. Myrvold and F. Ruskey, "Ranking and unranking permutations in linear time", Information Processing Letters, vol. 79, no. 6, **(2001)**, pp. 281-284.
[7] R. Schachtner, G. Pöppel, A. M. Tomé and E. W. Lang, "Minimum Determinant Constraint for Non-negative Matrix Factorization", Independent Component Analysis, Springer verlag, **(2009)**, pp. 106-113.
[8] J. Guo, "Fragile Watermarking Scheme for Tamper Detection of Relational Database", In International Conference on Computer and Management (CAMAN). **(2011)**, pp. 1-4.
[9] S. Bhattacharya and A. Cortesi, "Distortion-free Authentication Watermarking", Software and Data Technologies Communications in Computer and Information Science, Springer Berlin Heidelberg, vol. 170, **(2013)**, pp. 205-219.
[10] R. Bedi, A. Thengade, and V. M. Wadhai, "A New Watermarking Approach for Non-numeric Relational Database", International Journal of Computer Applications, vol. 13, no. 7, **(2011)**.
[11] L. Camara, J. Li, R. Li, and W. Xie, "Distortion-Free Watermarking Approach for Relational Database Integrity checking", Mathematical Problems in Engineering, Article ID 697165, 10 pages, 2014. doi:10.1155/2014, vol. 2014, **(2011)**.
[12] I. Kamel, "A schema for protecting the integrity of databases", Computers & Security, vol.28, no. 7 **(2009)**, pp. 698-709.
[13] G. Lingyun, W. Dong and H. Ali, "New Fragile Database Watermarking Scheme with Restoration Using Reed-Solomon Codes", Journal of Computational and Theoretical Nanoscience, vol. 10, no. 1, **(2013)**, pp. 147-153.

[14] D. Allan, N. Ashby, C. Hodge, and H.-P. Company, "The science of Time keeping Application note 1289", Hewlett-Packard, **(1997)**.
[15] L. Camara, J. Li, R. Li, F. Kagorora, and D. Hanyurwimfura, "Block-Based Scheme for Database Integrity verification", International Journal of Security and Its Applications, vol. 8, no. 6, **(2014)**, pp. 25-40.
[16] D. Allan, N. Ashby, C. Hodge, and H.-P. Company, "The science of Time keeping Application note 1289", Hewlett-Packard, **(1997)**.
[17] M. Kamran, A. Suhail, and M. Farooq, "A Robust, Distortion Minimizing Technique for Watermarking Relational Databases Using Once-for-All Usability Constraints", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 12, **(2013)**, pp. 2294-2707.

Authors

Camara Lancine, received his PhD degree of Computer Science and Technology in 2015 from Hunan University. He is currently teaching at the UNIVERSITY INSTUTIT OF MANAGEMENT of Bamako, Mali. His main research interests include information Security, and database application testing.

Demba Coulibaly, received his PhD degree of Computer Science and Technology in 2009 from University PARIS - Dauphine (France). He is teaching at the UNIVERSITY INSTUTIT OF MANAGEMENT of Bamako, Mali. His main research interests include Software Engineering, web services and database application.

Ali Hamadou, received his PhD degree of Computer Science and Technology in 2012 from Hunan University. He is currently teaching at MARADI DAN DICKO DANKOULODO UNIVERSITY, MARADI, NIGER. His main research interests include Information Security and Software Engineering.

Junyi Li, born in 1970, PhD and associate professor at the College of Information Science and Engineering, Hunan University. His main research interests include software engineering and database security.