Journal of Theoretical and Applied Information Technology
29th February 2016. Vol.84. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A ZERO-DISTORTION FRAGILE WATERMARKING SCHEME TO DETECT AND LOCALIZE MALICIOUS MODIFICATIONS IN TEXTUAL DATABASE RELATIONS

ABD. S. ALFAGI [1]*, A. ABD. MANAF [1], B. A. HAMIDA [2], R. F. OLANREWAJUB [2]
[1] Advanced Informatics School, Universiti Teknologi Malaysia ,Kuala Lumpur, Malaysia
[2] Department of Electrical and Computer Engineering, International Islamic University Malaysia ,Kuala Lumpur, Malaysia

ABSTRACT

In this paper, we present a new zero-distortion fragile watermarking scheme to detect and localize malicious modifications in textual database relations. Most existing fragile watermarking schemes introduce errors or permanent distortion into the original database content. These distortions violate the integrity of the database consequently the database quality and usability are degraded. Although, some fragile schemes are able to authenticate the database integrity and detect the malicious modifications made on the database but they are either tuples or attributes ordering based and unable to characterize the attack, identify the type of attack, identify the tampered data and locating the tampered tuples. In addition, most existing fragile schemes are based on LSB or MSB in generating the watermark unlike to this scheme, which is based on local characteristics of the relation itself such as frequencies of characters and text length in generating the watermark. This scheme is serviceable for sensitive and insensitive textual relational database since it does not introduce any error into the original contents. In addition, this scheme overcomes the weaknesses of data integrity violate, data usability and data quality degradation. The experimental results show the ability of proposed scheme in authenticating the database relation as well as the ability of characterize the attack, identifying the changed data and locating the text that affected by the malicious modification without depend on tuples and attributes ordering.

Keywords: *Database Watermarking, Fragile Watermarking Scheme, Robust Watermarking Scheme, Tamper Detection, and Authentication.*

1. INTRODUCTION

Digital watermark is a technique of embedding a piece of information into a carrier signal (image, text, audio) without destroy the signal quality. Primarily, digital watermark techniques were designed for protecting copyrights, ownership proof and integrity check of multimedia objects. [1-6]. Some watermark are designed for fingerprinting the data for unique identification of each buyer in order to detect traitors. In recent decades, researchers have extended some of those techniques to protect relational database as well. Most of the existing watermarking schemes for relational databases are techniques that introduce intentional errors or distortions. The introduced distortion can be performed at bit level, or character level, or higher such as attribute or tuple level, over the attribute values [7]. The watermarked relations may suffer of intentional errors or permanent distortion. Thus, a degradation in relational database quality is a result of watermarking process. Some database are sensitive and have strong usability constrain that disallow any errors as well as rejecting any permanent distortion. For example, medical, military, and safety database considered a sensitive data sets that cannot tolerate small errors or permanent distortion. Similarly, educational or business applications have relational databases, which are very sensitive like item-cost, order-quantity, customer-name and so other sensitive attributes.

2. RELATED WORK

Previous watermarking schemes can be categorize into *robust* and *fragile* schemes depend on the watermarking purposes. In this paper, we are focusing on fragile watermarking schemes that aimed for tamper detection and check relational database integrity.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.1 Robust Watermark Schemes

Most robust watermark schemes [8-32] aimed to protect relational database copyright, ownership proof or forgery detection. In robust watermarking, a watermark usually carries owner information in order to validate who the relation database belongs to (e.g., which person, which institute or organization, etc.). Most robust schemes are applicable for numerical data sets [9, 24-28, 30, 31], categorical attributes [15, 33-37], or textual attributes [18, 38-40] of relational database to embed the watermark bits. The embedded watermark could be a meaningless bits pattern [41, 42], meaningful bits pattern such as image [10, 26], speech [19, 43] or owner information [44, 45]. In robust watermarking schemes, the embedded watermark should be robust against attacks, which aim at removing the watermark or making it undetectable. Thus, the design purpose is to make the embedded watermark robust against malicious attacks such as removal of tuples or bit flipping [20]. The main assumption in robust watermarking schemes is that, relational database can tolerate minor change without affecting database usability. However, such change in sensitive database (e.g. medical data) may directly affect the usability of the database contents or may cause database integrity violation. Therefore, most robust schemes are not convenient for sensitive database due to the introduced errors.

2.2 Fragile watermark schemes

Fragile watermarking schemes are designed for integrity check and detecting manipulated data, since in many scenarios an attacker aims at altering the watermarked data while keeping the embedded watermark untouched [30]. Fragile watermarks should be sensitive to alterations, which means that the embedded watermark should not be detectable if any modifications have been made [46]. Many of fragile watermarking schemes are based on the content characteristics of database relations itself to generate a watermark which then used to check the relational database integrity or detecting tampered places. Guo, Li [20] presented a fragile watermarking scheme to detect tampering and any modification made on relational database R. In this scheme, the generated watermark is based on the content characteristics of numerical attributes of the relation R. Then the generated watermark securely embedded into most two LSBs of all numerical attributes in the relational database, which introduce a considerable distortion into the original database contents thus the usability of the database is effect. Similarly, khataeimaragheh and Rashidi [21] embedded the generated watermarks into the most two LSBs of all numerical attributes in the relation that forms a two-bit watermark grid. In both schemes, all tuples from candidate attributes are grouped to embed the generated watermarks. Almost in the same way Iqbal, Rauf [22] presented a fragile scheme for relational database integrity check but Iqbal, Rauf partitioned the R logically into three groups then generates self-constructing fragile watermark information from each group. The generated watermark of each groups is embedded at the LSBs of numerical attributes of that group. Conversely, Prasannakumari [23] presented a fragile watermarking scehem for detecting tamper in relational databse based on inserting a fake attribute into the relation to act as a watermark. The values of fake attribute are determined by aggregate function on original database content. These scheme considered a distortion-based schemes that introduce errors to the original content of the database. They either introduce a distortion into original contents of the database, affect the usability of the database as in [20] [21] [22] or change the relaitonal database structuer as in [23]. However, a permanent distortion or high level of distortion in sensitive data (e.g. medical data) are the main concern of owners since it may lead to wrong decision or might lead to undesired result or cause a significant cost. Therefore, most fragile schemes that introduce a distortion are not appropriate for sensitive database.

On the other hand, there are some distortion-free or a zero-distortion schemes [33, 34, 46, 47] proposed for integrity check and tamper proof. These schemes also generate a watermark based on content characteristics of relational database. In such schemes for check the integrity, the order of tuples and attributes is very important factor. Li, Guo [33] presented a distortion free watermarking scheme. Similar to, [34] and [46] who presented a fragile distortion-free watermarking scheme for tampering detection and integrity check. They generate a fragile watermark from categorical attributes of the relation R then partitioning R into disjoint groups using a message authentication code MAC with secret key. After that, the generated watermark embedded into each group separately. For verification process, same key is needed to regenerate the watermark using same partitioning approach then each partition can be verified independently. In contrast to tuples grouping and orders based schemes, Hamadou, Sun [47] presented a fragile watermarking scheme for integrity check based on attributes grouping. This scheme virtually sorted all attributes based the hash

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

values of each attribute's name in order to define a secret initial order. Then generates a watermark for each attributes based on the Most Significant Bits (MSBs). The generated watermark is registered with the Certification Authority (CA) for certification purpose. However, these schemes are tuples grouping orders based. Consequently, they are primary key (PK) based schemes that need unique PK for secure tuples grouping, thus any change in PK mislead regenerating same watermark even if there is no attacks on the R. In addition, schemes that are attributes grouping orders based need unique names for attributes, thus any change in attributes' name by attackers cause failure in the tamper detection process. Furthermore, those schemes are only workable for either relational database that contains categorical or numerical attributes inside the relation R.

From previously related fragile watermarking schemes, three important issues are addressed in this paper. Firstly, a fragile distortion-based watermarking schemes [20] [21] [22] [23] [30] [46] are applicable for certain type of relational database that tolerate change and accept errors without affecting database usability; thus, those schemes are not workable for sensitive relational database like medical, military, safety, and so forth. These relational databases are a non-error tolerant datasets. Secondly, even though there are a fragile zero distortion watermarking schemes presented by [33, 34, 46, 47] that do not introduce any change to the original contents of R, but these watermarking schemes are either tuples or attributes ordering based. Therefore, these watermarking schemes are susceptible to sorting attacks that make the system fauiler in the tamper detection process. Thirdly, most of previously mentioned fragile distortion-based and zero distortion watermarking schemes are applicable for numerical or categorical relational databases since they generate the watermark based on LSB or MSB of numerical attributes or categorical attributes. Therefore, they are not serviceable for textual database.

To address these issues, the authors proposed a novel fragile zero-distortion watermarking scheme for textual relational database for tamper detection and characterization. Dissimilar to [20, 21] [22, 23, 30, 33, 34, 46, 47], the proposed scheme is not based on LSB or MSB for generating the watermark, instead it is based on frequencies of characters and text length. In addition, this scheme is serviceable for sensitive and insensitive textual relational database since it does not introduce any modification into the original contents. Furthermore, this scheme overcomes the weaknesses like data integrity and data usability in existing fragile watermarking schemes. The proposed scheme, algorithmically evaluating the local characteristics of the relational database R and generate a watermark based on frequency distribution of characters (a~z) and text length of textual attributes. Doing so, enables this scheme to check the relational database integrity and characterize the malicious modifications (insertion, deletion, or update.

3. PROPOSED SCHEME AND FRAMEWORK

In this section, the fragile zero-distortion watermarking scheme for textual relational database for tamper detection and characterization is presented. Besides that, some significant and desired properties of a fragile watermarking system that are assured in this scheme are illustrated.

3.1 Desired Properties Of Proposed Scheme

In fragile watermarking attackers attempt to make malicious modification in watermarked relational database without affecting the watermark, whereas in robust watermark the attackers attempt to destroy the watermark without affecting the relational database usability. Therefore, the proposed scheme is intended to assure the fragility, usability, imperceptibility, blindness, and characterization.

i. ***Fragile***: that is if there are any data modifications, the generated watermark is undetectable.
ii. ***Imperceptible***: The proposed scheme is based on zero distortion; it does not introduce any distortion in the underlying data. Therefore, the embedded watermark is invisible or imperceptible.
iii. ***Blind***: The original relational database should not be required to detect and characterize malicious data modifications.
iv. ***Key-based system***: A secret key is requires in generation and verification process in this scheme.
v. ***Characterization***: The proposed scheme characterizes the malicious data modifications in database relation.

Table 1: Notations

Symbol	Description
R	Database relation
T_i	The i^{th} tuple
A_j	The j^{th} attribute
$Char_i$	The i^{th} character
$TxtLen_i$	The i^{th} Length of Text

Journal of Theoretical and Applied Information Technology
29th February 2016. Vol.84. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

W_{char}	The characters watermark
W_{txtlen}	The text length watermark
$fchar_i$	Frequency of character i^{th}
$fTxtLen_i$	Frequency of Text length i^{th}
$rfchar_i$	Relative frequency for i^{th} character
$rfTxtLen_i$	Relative frequency for i^{th} text length
$\Delta fchar_i$	Change in frequency of i^{th} character
$\Delta fTxtLen_i$	Change in frequency of i^{th} text length
$char_{totalcount}$	The total of Characters
W_{RDB}	Watermark of relational database
EW_{RDB}	Encrypted Watermark of relational database
$R_{egistred}W_{RDB}$	Registered Watermark of relational database
WAR	Watermark Accuracy Rate
WDR	Watermark Distortion Rate

2.3 Watermark Generation.

The watermark generation is totally based on the content characteristics of textual attributes inside the R. Therefore, we assume that the candidate relaiton for watermarking has some textual attributes. The proposed framework is shown in Figure 1. It mainly consist on two components: Watermark Generating System (WGS) and Watermark Encrypting System (WES). The specified textual attributes from Relation R with Primary Key PK and v attributes denoted by R(PK, A_1, A_2, . . . , A_v) is the main input that send to WGS. This system generates two watermark; characters' watermark (W_{char}) and text length watermark (W_{txtlen}). These watermarks are concantenated to form relational database watermark (W_{RDB}). The WES recieves W_{RDB}, SK and any related security parameters (e.g. owner's ID). WES encrypts the W_{RDB} with SK to obtain an encrypted relational database watermark (EW_{RDB}) which is then concatenated with owner's ID to form certificated watermark (W_{Cer}). This watermark registered at Certification Authority (CA) for security and purposes.

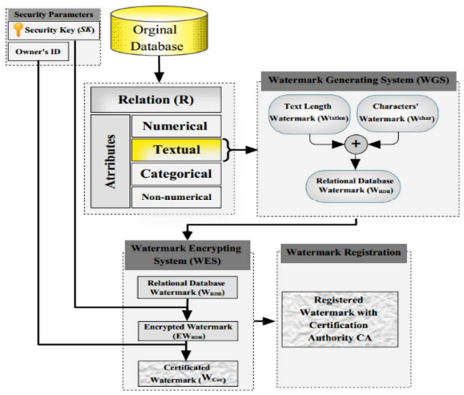

Figure 1: Proposed Watermark Generation And Registration Framework For Textual Relational Database

Algorithm 1 illustrates the main process of watermark generation that generate W_{RDB}. At line 1, the frequency of each character (a~z) is computed as in Algorithm 2 to W_{char}. Line 2, computes the frequency of text length as in Algorithm 3 in order to generate W_{txtlen}. These watermarks are used to generate W_{RDB} as shown at line 3. The W_{RDB} is encrypted with a secret key which known to the database owner only to obtain encrypted EW_{RDB} as in line 4. At line 5 a certified watermark W_{cer} is computed by concatenating the EW_{RDB} with the owners information (e.g owner's ID), the date and time to serve as stamp. The final step at line 6 for obtain the registered watermark of relational database $R_{egistred}W_{RDB}$ with certification Authority.

(1) W_{char} = Characters watermark generation () // *see Algorithm 2*
(2) W_{txtlen} = Text length watermark generation () // *see Algorithm 3*
(3) W_{RDB} = W_{char} || W_{txtlen}
(4) EW_{RDB} = Encrypt (W_{RDB} , SK)
(5) $W_{certified}$ = EW_{RDB} || ownerID || date || time
(6) $R_{egistred}W_{RDB}$ = $W_{certified}$ register to CA

Algorithm 1: Relational database watermark generation

Algorithm 2 generates a W_{char}, which is based on characters frequency in all selected textual attribute A_j. At line 1-2, the length of tuples and the attributes are determined which are then used to check each character individually as shown in line 6. At line 7, the frequency of each character is computed in order to calculate the $rfchar_i$, which is then used to W_{char} as shown in line 13-14. The final generated watermark of characters is concatenated with the total Characters of the specified length of i^{th} tuples and j^{th} attributes. The W_{char} aimed to characterize the malicious modification if any.

(1) **For each** $T_i \in$ R **Do**
(2) For each $A_j \in$ R **Do**
(3) String str = get value of(T_i, A_j)
(4) int iLength=Len(str)
(5) **For** i=0 to iLength-1 **Do**
(6) $Char_{(i)}$=Mid(str,i,1)
(7) *fchar* of [$Char_{(i)}$] ++
(8) $char_{totalcount}$++
(9) **End for**
(10) **End for**
(11) **End for**
(12) **For each** character i $\in$ (a~z) **Do**
(13) $rfchar_i$ = (*fchar* of char[i] / $char_{totalcount}$) * 100
(14) W_{char} = W_{char} || $rfchar_i$
(15) **End for**
(16) W_{char} = W_{char} || $char_{totalcount}$

Algorithm 2: Characters watermark generation

Algorithm 3 is used to generate W_{txtlen}. At line 1-2, the length of text at i^{th} tuple and j^{th} attribute value is determined. Line 4-6 compute the frequency of each length and the total count of text length in

Journal of Theoretical and Applied Information Technology
29th February 2016. Vol.84. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

order to generate text length watermark W_{txtlen}. The final watermark is concatenated with the text length total count. The W_{txtlen} aimed to characterize the malicious modification on the length of text at i^{th} and j^{th} value and to identify length that has been tampered with.

(1) **For each** $T_i \in$ R **Do**
(2) **For each** $A_j \in$ R **Do**
(3) String str = get value of(T_i, A_j)
(4) $TxtLen_{(i)}$=Len(str)
(5) $ftxtlen_i$ [$TxtLen_{(i)}$] ++
(6) $TxtLen_{totalcount}$++
(7) **End for**
(8) **End for**
(9) **For each** i $\in$ length **Do**
(10) $rfTxtLen_i = (ftxtlen_i / txtLen_{totalcount}) * 100$
(11) $W_{txtlen} = W_{txtlen} \| rfTxtLen_i$
(12) **End for**
(13) $W_{txtlen} = W_{txtlen} \| TxtLen_{totalcount}$

Algorithm 3: Text length watermark generation

2.4 Watermark Verification.

The verification process are important at receiving side to detect whether the relational database has been tampered with or no. The verification framework of proposed scheme is shown in *Figure 2*.

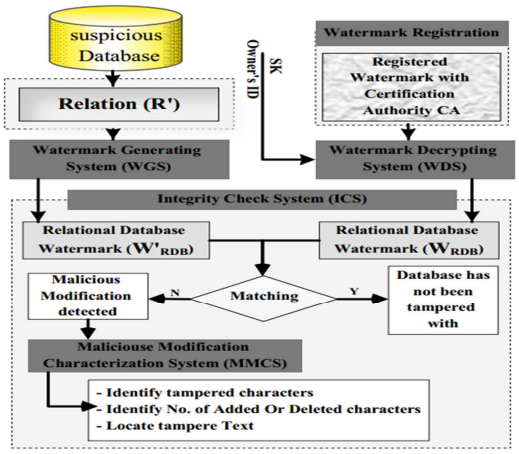

Figure 2: Proposed Framework For Detection Of Malicious Tempering

This framework performs relational database integrity check on the suspicious R' which is inputted to WGS that generates W'_{char} and W'_{txtlen} in order to obtain W'_{RDB} (Algorithm 4 line 5). The Watermark Decrypting System (WDS) works to decrypt the registered watermark (Algorithm 4 lines 1-4). It is to be noted that, same *SK* is needed to obtain W_{RDB}. Then both W'_{RDB} and W_{RDB} send to Integrity Check System (ICS). If ICS found zero difference then the received R' is not modified (Algorithm 4 lines 14) otherwise the ICS runs Algorithm 5 and 6 that is a Malicious Modification Characterization System (MMCS). This system identifies the tampered characters, number of added or deleted characters, and locating the tampered text based on the text length. It is to be noted that text length can be defined using adjustable range. Defining length range help to identify in which length the malicious modifications made. Lines 12-17 in Algorithm 4, compute the Watermark Accuracy Rate (WAR) and Watermark Distortion Rate (WDR) in order to evaluate the accuracy and the distortion rate of the watermark. These values are important factors that help database owners' to accept or reject the received database. Since some database may accept minor change without affecting database usability.

(1) **Obtain** $Registered W_{RDB}$ from CA
(2) **Extract** W_{cer} from $Registered W_{RDB}$
(3) **Extract** EW_{RDB} from W_{cer}
(4) W_{RDB} =Decrypt (EW_{RDB},*SK*)
(5) **Compute** W'_{RDB} for R' **using Algorithm 1**
(6) **Send** W_{RDB} and W'_{RDB} ICS
(7) **For** *i*=0 to Length(W_{RDB})-1 **DO**
(8) If $W_{RDB}[i]$= W'_{RDB}[i] then
(9) MatchCount++
(10) **Endif**
(11) TotalMatchCount++
(12) WAR= MatchCount/TotalMatchCount *100
(13) WDR=1-WAR
(14) If WDR <> 0 then
(15) R' is tampered with
(16) Use ($\Delta fchar_i$) to determine tampered characters *//see* ***Algorithm5***
(17) Use ($\Delta ftxtlen_{ij}$) to locate tampered Length *//see* ***Algorithm6***
(18) Endif

Algorithm 4: Watermark verification

(1) **Extract** W_{char} **from** W_{RDB}
(2) **Extract** $rfchar_i$ from W_{char}
(3) Compute $fchar_i$ =($rfchar\ of\ char[i] * char_{totalcount}$)/100
(4) Generate $f'char_i$ for R' using **Algorithm 2**
(5) $\Delta fchar_i = fchar_i - f'char_i$
(6) $\Delta frafchar_i = \Delta fchar_i / fchar_i * 100$

Algorithm 5: Compute change in character frequency

(1) **Extract** W_{TxtLen} **from** W_{RDB}
(2) **Extract** $rfTxtLen_i$ from W_{TxtLen}
(3) Compute $ftxtlen_i$ =($rfTxtLeni / txtLen_{totalcount}$)*100
(4) Generate $f'txtleni$ for R' using **Algorithm 3**
(5) $\Delta fixtleni = fixtlen_i - f'txtlen_i$
(6) $\Delta frafixtlen_i = \Delta fixtlen_i / fixtlen_i * 100$

Algorithm 6: Compute change in text length frequency

4. RESULT AND DISCUSSION

The proposed scheme is tested and evaluated on real-life dataset namely Doctors by grad and specialty. The data set collected from Hospital and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Community Health Service (HCHS) database, which is provided by Health and Social Care Information Center (HSCIC) and available at [48]. A Visual studio 2015 is installed on 3.2GHz Intel Core i5 CPU with 8 GB of RAM and windows 10 to conduct the experiment. The data set has 37321 tuples, each with numerical, categorical and textual attributes. In the experiment we focused on a textual attribute.

4.1 Authenticating Relational Database

For authenticating relational database, we have conducted random (insertion, deletion and update) attacks on relation R. Figure 3 shows the effect of attacks on the watermark.

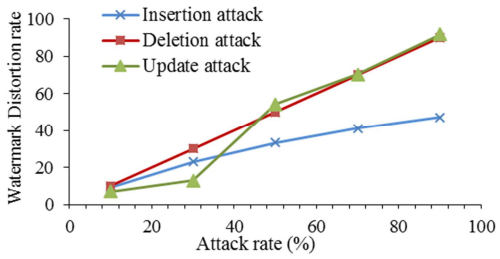

Figure 3: WDR For Malicious Modification Of Deletion, Insertion And Update Attack With

The three colored trends represent the WDR for malicious modification of insertion deletion and update attacks. The X-axis indicates the percentage of attack that started from zero to 100% percent of attack. The Y-axis indicates the distortion rate on the regenerated watermark from the relation R after 10%, 30%, 50%, 70% and 90% of insertion, deletion, and update attacks respectively. The zero rate of distortion of WDR indicates that R is authenticated and no tamper has been made on the R whereas the higher rate of distortion indicates that the relation has been maliciously tampered with and when WDR goes higher indicates that the percentage of tampering is also high. The obtained result shows that, the trends of WDR is always goes high whenever the percentage of deletion, insertion or update attack increased. The fragility of the relational database watermark W_{RDB} is observed for even simple attack and the proposed scheme was able to detect any malicious modifications made on the relation.

4.2 Identifying Malicious Modifications

In this test, we have conducted a random insertion, deletion, and update attacks on different characters inside the textual tuples of R. The line graph presented in Figure 4 shows the result of change in the related frequency of each characters. The positive trend indicates that there is a malicious insertion in the i^{th} character, whereas the negative trend indicates that malicious deletion have been made on i^{th} characters. The trend that has both positive and negative values indicate to malicious update on i^{th} characters. The X-axis represents the English alphabets (when i=1 indicates to alphabet a, and so on). From the figure, it is noticeably that most characters have been tampered with insertion attack that increases the frequencies of some i^{th} characters by 10% to 15 %. While the negative trend indicate, some characters have been tampered with deletion attack with almost same percent of 10% to 15 %. In addition, the figure shows that there is also update attack on the relations since there is trend has positive and negative values. The update trend indicates that few characters were inserted. For example, the $char_{(3,4,5)}$ have been updated by 10% of increase meanwhile some other characters have been decreased such as $char_{(12,\ 15,\ 18,21)}$. The figure also shows that only few characters (e.g. $char_{(10,11,17,23)}$) have not been tampered with as they recorded zero change in their frequencies and they were not affected by insertion, deletion or update attack. The result illustrates the proposed scheme having the ability on identifying modification such as (insertion, deletion, or update) as well as the ability of identifying the tampered characters.

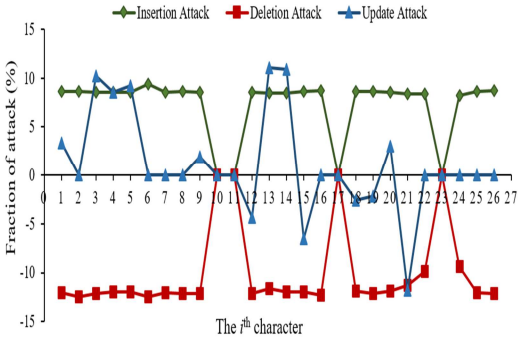

Figure 4: Identification Of Malicious Modification On Random Characters With <20% Of (Insertion, Deletion And Update Attack) On Character Frequency

4.3 Locating Malicious Modifications

The line graph presented in Figure 5 describes the location of malicious insertion, deletion and update attacks on random characters made in R. The fractional change in relative frequency of i^{th} length is used to locate the tampered text. The trends represent the fractional change of relative

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

frequency of i^{th} length after simulating insertion, deletion, and update attack. The positive values of the trend indicate the increase of the i^{th} length, whereas the negative values indicate the decrease of the i^{th} length. From the figure it is clearly seen that most tampered characters located in the i^{th} length when *i=(1,3,4,5,6,7,8,9,10,11)* which mean the majority of data tampering (insertion and deletion) made on text length that ranged from 0~3, 4~7, 8~11, 12~15, 16~19, 20~23, 24~27, 28~31, 32~35, 36~39 and 40~43. Whereas there is no change in the i^{th} length when *i=(12-20)*. In update attack, the most tampered characters located in the i^{th} length when *i=(2,4,7)* while the other length has not changed. The result illustrates the ability of proposed scheme in locating the tampered text.

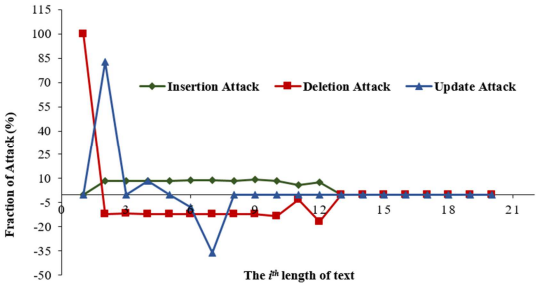

Figure 5: Location Of Malicious Modification On Random Characters With <20% Of (Insertion, Deletion And Update Attack) On Character Frequency

5. DISCUSSION

The result presented in *Figure 3*, Figure 4 and Figure 5 show the ability of proposed scheme in not only authenticating the relation R but also the ability of characterize the attack and identifying the changed characters as well as locating the text length range that affected by the malicious modification. It is to be noted that the data characteristics used for our experiments like characters and text length frequency are cohesive to each another. However, we evaluated the effect of malicious data alterations on characters frequency only. For example, if attackers maliciously delete or insert any character (a~z) into any tuple the length of text may also decrease or increase. We conclude our finding and observations as following:

- If the WDR is not equal to zero then the suspected relation R' is tampered with and the data inside R has been modified (Figure 3).
- If there is a positive trend in frequency of i^{th} character, means malicious insertion made to the relation R' (Figure 4).
- If there is a negative trend in frequency of i^{th} character, means malicious deletion made to the relation R' (Figure 4).
- If there is positive and negative values in frequency of i^{th} character, means that malicious update made to the relation R'(Figure 4).
- If there is a positive trend in the i^{th} length of text, means malicious insertion made to the i^{th} length inside the relation R' (Figure 5).
- If there is a negative trend in the i^{th} length of text, means malicious insertion made to the i^{th} length inside the relation R' (Figure 5).

6. CONCLUSION

In this paper, a zero-distortion fragile watermarking scheme to detect and localize malicious modifications in textual database relations, the watermark generation and the malicious modification detection frameworks are presented. The scheme is based on the approach that introduce a zero distortion into the original database content. Therefore, the scheme overcomes the limitation of data integrity and usability in some existing watermarking scheme which make this scheme workable for sensitive relational database and non-error tolerant datasets like medical, military, safety, and so forth. In contrast to many existing schemes, the proposed scheme is not LSB or MSB based for generating the watermark, instead it is based on local characteristics of the relarion itself such as frequencies of characters and text length. Thus, this scheme is serviceable for textual database. The experimental result shows that the proposed scheme can check the database integrity, characterize the malicious modification, and identify the modified characters as well as locating the tampered tuples without depend on tuples or attributes ordering. In the future, we intend to work on some additional local characteristics of the database relations for generating the watermark and to extend the proposed scheme to serve textual and numerical database relations.

ACKNOWLEDGMENTS

The authors would like to express greatest appreciation to Advanced Informatics School (AIS), Universiti Teknologi Malaysia (UTM) for financial support. Furthermore, thanks to a financial

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

support by Ministry of Defense, Libya under their scholarship program for PhD studies of the first author.

REFRENCES

[1] Halder, R., S. Pal, and A. Cortesi, *Watermarking Techniques for Relational Databases: Survey, Classification and Comparison*. J. UCS, 2010. **16**(21): p. 3164-3190.

[2] Arathi, C., *Literature Survey on Distortion based Watermarking Techniques for Databases.* International Journal of Computer Science & Communication Networks, 2012. **2**(4).

[3] Abdullah, S.M., A.A. Manaf, and M. Zamani, *Capacity and quality improvement in reversible image watermarking approach*, in *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*. 2010: Seoul. p. 81 - 85.

[4] Tsai, M.-H., H.-Y. Tseng, and C.-Y. Lai. *A Database Watermarking Technique for Temper Detection*. in *JCIS*. 2006.

[5] Dadkhah, S., et al., *An effective SVD-based image tampering detection and self-recovery using active watermarking.* Signal Processing: Image Communication, 2014. **29**(10): p. 1197-1210.

[6] Olanrewaju, R.F. and O. Khalifa. *Digital audio watermarking; techniques and applications*. in *Computer and Communication Engineering (ICCCE), 2012 International Conference on*. 2012.

[7] Sagar, R., *Watermark based Copyright Protection for Relational database.* International Journal of Computer Applications, 2013. **78**(2).

[8] Agrawal, R., P.J. Haas, and J. Kiernan. *A system for watermarking relational databases*. in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 2003. ACM.

[9] Gupta, G. and J. Pieprzyk, *Database relation watermarking resilient against secondary watermarking attacks*, in *Information Systems Security*. 2009, Springer. p. 222-236.

[10] Zhang, Y., et al. *A Method of Verifying Relational Databases Ownership with Image Watermark.* in *The 6th International Symposium on Test and Measurement, Dalian, PR China*. 2005.

[11] Guo, F., et al., *An improved algorithm to watermark numeric relational data*, in *Information Security Applications*. 2006, Springer. p. 138-149.

[12] Huang, M., et al. *A new watermark mechanism for relational data*. in *Computer and Information Technology, International Conference on*. 2004. IEEE Computer Society.

[13] Hu, T.-L., et al., *Garwm: Towards a generalized and adaptive watermark scheme for relational data*, in *Advances in Web-Age Information Management*. 2005, Springer. p. 380-391.

[14] Sion, R. *Proving ownership over categorical data*. in *Data Engineering, 2004. Proceedings. 20th International Conference on*. 2004. IEEE.

[15] Sion, R., M.J. Atallah, and S. Prabhakar, *Rights protection for categorical data.* Knowledge and Data Engineering, IEEE Transactions on, 2005. **17**(7): p. 912-926.

[16] Zhou, C.W.J.W.M. and G.C.D. Li. *Atbam: An arnold transform based method on watermarking relational data*. 2008. Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on.

[17] Zhou, X., M. Huang, and Z. Peng. *An additive-attack-proof watermarking mechanism for databases' copyrights protection using image*. in *Proceedings of the 2007 ACM symposium on Applied computing*. 2007. ACM.

[18] Al-Haj, A. and A. Odeh, *Robust and blind watermarking of relational database systems.* Journal of Computer Science, 2008. **4**(12): p. 1024.

[19] Wang, H., X. Cui, and Z. Cao, *A Speech Based Algorithm for Watermarking Relational Databases.* 2008: p. 603-606.

[20] Guo, H., et al., *A fragile watermarking scheme for detecting malicious modifications of database relations.* Information Sciences, 2006. **176**(10): p. 1350-1378.

[21] khataeimaragheh, H. and H. Rashidi, *A Novel Watermarking Scheme for Detecting and Recovering Distotions in Database Tables.* International Journal of Database Management Systems, 2010. **2**(3): p. 1-11.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] Iqbal, S., et al., *Self-constructing fragile watermark algorithm for. relational database integrity proof.* World Applied Sciences Journal, 2012. **19**(9): p. 1273-1277.

[23] Prasannakumari, V., *A robust tamperproof watermarking for data integrity in relational databases.* Research Journal of Information Technology, 2009. **1**(3): p. 115-121.

[24] Khanduja, V., et al., *A robust multiple watermarking technique for information recovery*, in *Advance coputing conference*. 2014.

[25] Farfoura, M.E., S.-J. Horng, and X. Wang, *A novel blind reversible method for watermarking relational databases.* Journal of the Chinese Institute of Engineers, 2014. **36**(1): p. 87-97.

[26] Farfoura, M.E. and S.-J. Horng, *A Novel Blind Reversible Method for Watermarking Relational Databases.* International Symposium on Parallel and Distributed Processing with Applications, 2010: p. 563-569.

[27] Thodi, D.M. and J.J. Rodriguez. *Prediction-error based reversible watermarking*. in *Image Processing, 2004. ICIP'04. 2004 International Conference on*. 2004. IEEE.

[28] Thodi, D.M. and J.J. Rodríguez. *Reversible watermarking by prediction-error expansion*. in *Image Analysis and Interpretation, 2004. 6th IEEE Southwest Symposium on*. 2004. IEEE.

[29] Jawad, K. and A. Khan, *Genetic algorithm and difference expansion based reversible watermarking for relational databases.* Journal of Systems and Software, 2013. **86**(11): p. 2742-2753.

[30] Farfoura, M.E., et al., *A blind reversible method for watermarking relational databases based on a time-stamping protocol.* Expert Systems with Applications, 2012. **39**(3): p. 3185-3196.

[31] Mehta, B.B. and U.P. Rao, *A Novel approach as Multi-place Watermarking for Security in Database.* arXiv preprint arXiv:1402.7341, 2014.

[32] Olanrewaju, R., et al. *Detection of alterations in watermarked medical images using Fast Fourier Transform and Complex-Valued Neural Network*. in *Mechatronics (ICOM), 2011 4th International Conference On*. 2011. IEEE.

[33] Li, Y., H. Guo, and S. Jajodia. *Tamper detection and localization for categorical data using fragile watermarks*. in *Proceedings of the 4th ACM workshop on Digital rights management*. 2004. ACM.

[34] Bhattacharya, S. and A. Cortesi. *A Distortion Free Watermark Framework for Relational Databases*. in *ICSOFT (2)*. 2009.

[35] Bhattacharya, S. and A. Cortesi. *Database Authentication by Distortion Free Watermarking*. in *ICSOFT (1)*. 2010. Citeseer.

[36] Bhattacharya, S. and A. Cortesi, *A generic distortion free watermarking technique for relational databases*, in *Information Systems Security*. 2009, Springer. p. 252-264.

[37] Sion, R., M. Atallah, and S. Prabhakar, *Rights protection for relational data.* IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,, 2004. **16**(2).

[38] Shah, S.A., S. Xingming, and H. Ali, *Query Preserving Relational Database Watermarking.* An international Journal of computing and informatics, 2011.

[39] Bedi, R., A. Thengade, and V.M. Wadhai, *A New Watermarking Approach for Non-numaric Relational Database.* International Journal of Computer Applications, 2011. **13**(7).

[40] Sonnleitner, E. *A robust watermarking approach for large databases*. in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*. 2012. IEEE.

[41] Agrawal, R. and J. Kiernan. *Watermarking Relational Databases*. in *VLDB Conference*. 2002. Hong Kong,China,.

[42] Qin, Z., et al. *Watermark based copyright protection of outsourced database*. in *Database Engineering and Applications Symposium, 2006. IDEAS'06. 10th International*. 2006. IEEE.

[43] Zhang, Y.H., Z.X. Gao, and D.X. Yu, *Speech Algorithm for Watermarking Relational Databases Based on Weighted.* Advanced Materials Research, 2010. **121-122**: p. 399-404.

[44] Huang, K., et al., *A Cluster-Based Watermarking Technique for Relational Database.* First International Workshop on Database Technology and Applications, 2009.

[45] Zhao, X., L. Li, and Q. Wu, *A Novel Multiple Watch marking for Relational Databases using Multi-Media.* Physics Procedia, 2012. **25**: p. 687-692.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[46] Kamel, I., *A schema for protecting the integrity of databases.* Computers & Security, 2009. **28**(7): p. 698-709.
[47] Hamadou, A., et al., *A fragile zero-watermarking technique for authentication of relational databases.* International Journal of Digital Content Technology and its Applications, 2011. **5**(5): p. 189-200.
[48] *Health and Social Care Information Centre (hscic)*. 2015, http://www.hscic.gov.uk.