A Zero Text Watermarking Algorithm based on the Probabilistic weights for Content Authentication of Text Documents

Fahd N. Al-Wesabi
PhD Candidate, Faculty of Engineering
SRTM University, Nanded, P.O. Box: 431606, Nanded, Maharashtra, INDIA

Adnan Z. Alshakaf
Professor, Department of IS, Faculty of Computing and IT, UST, P.O. Box: 13064 ,Sana'a, YEMEN

Kulkarni U. Vasantrao
Professor and Head, Department of Computer Science and Engineering, SGGS Institute of Engineering and Technology, P.O. Box: 431606, Nanded, Maharashtra, INDIA

ABSTRACT
In the study of content authentication and tamper detection of digital text documents, a novel English text zero-watermarking approach based on probabilistic patterns proposed in this paper. Hidden Markov model was presented, and the core algorithm of the watermark generation and detection was designed in this paper. In the proposed approach, Hidden Markov model letter-level based of order one was constructed for text analysis and watermark generation based on interrelationship between contents of host text document by using the two-dimensional matrix coordinate of probabilistic weights of states and transitions. However, we can extracted this watermark later using extraction and detection algorithm to identify the status of text document such as authentic, tampered, etc. Furthermore, the effectiveness and feasibility of our proposed algorithm was proved with algorithm experiments and comparative with other recently approach under random insertion and deletion attacks in localized and dispersed form on 6 text datasets of varying length. Results showed that our proposed algorithm is more secure, had better robustness, and always detects tampering attacks occurred randomly on text even when the tampering volume is low or high.

General Terms
Probabilistic weights, information hiding, Content Authentication, Tamper detection, Copyright protection.

Keywords
Digital watermarking, Hidden Markov Model.

1. INTRODUCTION
With the increasing use of internet, e-commerce, and other efficient communication technologies, the copyright protection and authentication of digital contents, has gained great importance. Most of these digital contents are in text form such as email, websites, chats, ecommerce, eBooks, news, and SMS[1].

These text documents may be tempered by malicious attackers, and the modified data can lead to fatal wrong decision and transaction disputes[2].

Content authentication and tamper detection of digital image, audio, and video has been of great interest to the researchers. Recently, copyright protection, content authentication, and tamper detection of text document attracted the interest of researchers. Moreover, during the last decade, the research on text watermarking schemes mainly focused on issues of copyright protection, but gave less attention on content authentication, integrity verification, and tamper detection[4].

Various techniques have been proposed for copyright protection, authentication, and tamper detection for digital text documents. Digital Watermarking (DWM) techniques is considered the most powerful solution to most of these problems. Digital watermarking is a technology in which various information such as image, a plain text, an audio, a video or a combination of all can be embedded as a watermark in digital content for several applications such as copyright protection, owner identification, content authentication, tamper detection, access control, and many other applications [2].

Traditional text watermarking techniques such as format-based, content-based, and image-based requires the use of some transformations or modifications on contents of text document to embed watermark information within text. A new technique have been proposed named as a zero-watermarking for text documents. The main idea of zero-watermarking techniques is that it does not change the contents of original text document, but utilizes the contents of the text itself to generate the watermark information[13].

In this paper, we present a new zero-watermarking technique for digital text documents. Our technique utilizes the probabilistic nature of the natural languages, mainly the first order Markov model.

2. PREVIOUS WORK
Text watermarking techniques have been proposed and classified by many literatures based on several features and embedding modes of text watermarking. We have examined briefly some traditional classifications of digital watermarking as in literatures. These techniques involve text images, content based, format based, features based, synonym substitution based, and syntactic structure based, acronym based, noun-verb based, and many others of text watermarking algorithms that depend on various viewpoints [1][3][4].

2.1 Format-based Techniques
Text watermarking techniques based on format are layout dependent. In [5], proposed three different embedding methods for text documents which are, line shift coding, word shift coding, and feature coding. In line-shift coding technique, each even line is shifted up or down depending on the bit value in the watermark bits. Mostly, The line is shifted up if the bit is one, otherwise, the line is shifted down. The odd lines are considered as control lines and used at decoding.

Similarly, In word-shift coding technique, words are shifted and modifies the inter-word spaces to embed the watermark bits. Finally, In the feature coding technique, certain text features such as the pixel of characters, the length of the end lines in characters are altered in a specific way to encode the zeros and ones of watermark bits. Watermark detection process is performed by comparing the original and watermarked document.

2.2 Content-based Techniques

Text watermarking techniques based on content are structure-based natural language dependent[4]. In [6][14], a syntactic approach has been proposed which use syntactic structure of cover text for embedding watermark bits by performed syntactic transformations to syntactic tree diagram taking into account conserving of natural properties of text during watermark embedding process. In [18], a synonym substitution has been proposed to embed watermark by replacing certain words with their synonyms without changing the sense and context of text.

2.3 Binary Image-based Techniques

Watermarking techniques of binary image documents can be depends on traditional image watermarking techniques that based on space domain and transform domain, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Least Significant Bit (LSB) [5]. Several formal text watermarking methods have been proposed based on embedding watermark in text image by shifting the words and sentences right or left, or shifting the lines up or down to embed watermark bits as we mentioned above in section format-based watermarking [5][7].

2.4 Zero-based Techniques

Text watermarking techniques based on Zero-based watermarking are content features dependent. There are several approaches that designed for text documents have been proposed in the literatures which are reviewed in this paper [1][19] [20] and [21].

The first algorithm has been proposed by [19] for tamper detection in plain text documents based on length of words and using digital watermarking and certifying authority techniques. The second algorithm has been proposed by [20] for improve of text authenticity in which utilizes the contents of text to generate a watermark and this watermark is later extracted to prove the authenticity of text document. The third algorithm has been proposed by [1] for copyright protection of text contents based on occurrence frequency of non-vowel ASCII characters and words. The last algorithm has been proposed by [21] for to protect all open textual digital contents from counterfeit in which insert the watermark image logically in text and extracted it later to prove ownership.

2.5 Combined-based Techniques

we can say the text is dissimilar image. Thus, language has a distinct and syntactical nature that makes such techniques more difficult to apply. Thus, text should be treated as text instead of an image, and the watermarking process should be performed differently. In [22] A combined method has been proposed for copyright protection that combines the best of both image based text watermarking and language based watermarking techniques.

3. THE PROPOSED APPROACH

Traditional digital text watermarking approaches are based on making some modifications on original text document to embed added external information in text document and this information can be used later for various purposes such as content authentication, integrity verification, tamper detection, or copyright protection. In this paper we propose a zero-watermarking approach based on the Markov model for the natural language in which the original text document is not altered to embed watermark, that's mean the watermark embedding process is performs logically. The probabilistic features of the text are utilized to generate a watermark key which is stored in a watermark database. This watermark key can be used later and matched with watermark generated from attacked document for identifying any tampering that may happen to the document and authenticating its content.

The watermark generation and extraction process is illustrated in figure 1. Watermark information is generated from text analysis process using Hidden Markov model and stored in watermark database with other related information of text document such as document ID, current date and time. On the other hand, It is used in the extraction algorithm later for content authentication and tampering detection in the text document.

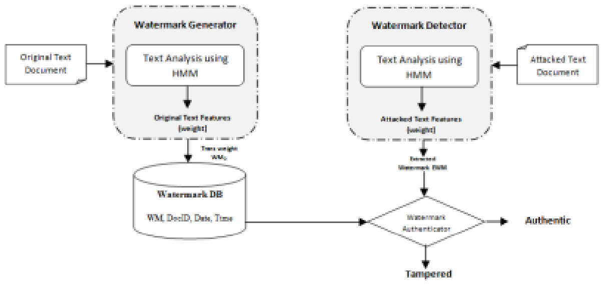

Fig. 1: Watermark Generation and Extraction Processes

Generally, an attacker tries to perform any tampering processes on the text such as insertion of new text, deletion of existing text or any other kind of attacks. Typical stages of watermarking system involves two process: (1) embedding process and (2) detection process. Watermark embedding is done by the original author during document uploading and detection done later automatically during document downloading for content authentication and tampering detection.

3.1 DWM Embedding Algorithm

The watermark embedding algorithm requires the original text file as input. A watermark is generated as the output of this algorithm. This watermark is then stored in watermark database along with the original text document, document id, author name, current date and time.

The original text document (TDO) is to be provided by the author. Then we compute the number of occurrences of the next state transitions for every present state, in this approach we consider the state as a single character of the text document. A matrix that represents the number of occurrences of transition from a state to another is constructed by equation (1) as follows.

The matrix size is [37][39] where 37 is the number of states, and 39 is the number of possible transitions of each state except last two columns we used it to trace the total frequency of occurrence for each transition and the weights of transitions related of each state. The number of states presented in our approach is obtained as follows: [English letters = 26, space letter= 1, Integer numbers from 0 to 9= 10] which is equal to 37 states.

Total Number of Transition[i][j], for i.j=1,2, .,37 .(1)

Where,

- i: refers to PS "the present state".
- j: refers to NS "the next state".

All the nonzero transitions are identified and summated as a frequency for each state and its value used to generate watermark weights by equation (2). These weights are calculated to construct a watermark by equation (3) as follows.

The before last column in the matrix is called Frequency which store the total transitions of each state in given text by equation (2):

$$Frequency[i] = \sum_{j=1}^{61} P[i,j] \quad \text{for every state } i = 1, 2, \cdots, 37 \qquad (2)$$

And the last column in the matrix is called Weights denoted by $\mathbf{Wt[i]}$ which store the probability transitions weight of each state by equation (3):

$$\mathbf{Wt[i]} = \frac{1}{Frequency\ [i]} \text{ for every state } i = 1, 2, \cdots, 37 \ \ldots (3)$$

The total transitions frequencies of all states can be calculated by equation (4):

$$TFS = \sum_{i=1}^{n} Frequency[i] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (4)$$

Where, n = 37, the number of states represented in matrix of HMM.

The total weights of all states refers to the interrelationship between the contents of given text documents, and this value represent the watermark which we can get for it by equation (5):

$$Total\ Weight = \sum_{i=1}^{n} \mathbf{Wt[i]} \quad \ldots\ldots\ldots\ldots(5)$$

Where, n = number of states represented in matrix of HMM.

These calculations are illustrated in figure 2 based on letter-level HMM of order 1.

S/T	0	1	…	8	9	A	B	…	X	Y	Z	Space	,	…	?	!	Frequency	Weight
0	17	0		0	0	0	0		0	0	0	28	3		0	0	51	0.02
1	7	2		1	13	0	0		0	0	0	4	2		0	0	38	0.03
⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮
9	2	0		11	2	0	0		0	0	0	8	0		0	0	25	0.04
A	0	0		0	0	0	11		0	10	1	61	4		0	0	791	0.00
B	0	0		0	0	13	2		0	14	0	0	0		0	0	140	0.01
⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮
Z	0	0		0	0	4	0		0	0	0	0	0		0	0	8	0.13
Space	0	29		0	2	201	81		0	21	4	235	0		0	0	2148	0.00
,	5	2		0	0	0	0		0	0	0	121	0		0	0	138	0.01
⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮	⋮
!	0	0		0	0	0	0		0	0	0	0	0		0	0	0	0.00
																Total	12506	2.1451

Fig. 2: Watermark Generation

In our proposed algorithm the watermark is not restricted to alphabetic characters, but includes spaces, and integer numbers. The DWM generation algorithm based on HMM proceeds as illustrated in figure 3.

DWM Embedding Algorithm (**A Zero Text DWM based on Probabilistic Weights**)

- **Input**: Original Text Document (**TD_O**)
- **Output**: WMo value, and States and Transitions Matrix [37][39]

1. Read Original Text document to TD_O array.
2. Loop i = 1 to 37, // Build States matrix and compute the number of occurrences of the next state transitions from every present state to another
   - Loop j = 1 to 37,
     - Total Number of Transition[i][j]
3. Loop i = 1 to 37, // Compute Probabilistic Weights for each state
   - Loop j = 1 to 39,
     - Frequency += States [i] [j]
   - States [i] [38] = Frequency
   - States [i] [39] = 1/Frequency
   - Frequency = 0
   - WMo += States [i] [39] // Generate original watermark

WM_O: Original watermark, **TD_O**: Original text document array, **States:** States and Transitions matrix, **Frequency:** total probability of transitions, **Weight:** total probability of states.

Fig. 3: Watermark generation algorithm

3.2 DWM Detection Algorithm

Extracting the watermark from the attacked text document and matching it with the original watermark is done by the detection algorithm. The proposed detection algorithm takes the attacked text and the original watermark or original text document as inputs.

The watermark is generates from the attacked text by the extraction algorithm and is then, compared with the original watermark stored in watermark database or by generate original watermark from the original text document. In case of storing original watermark in database, we have also stored document ID, current date and time in related record in database. Multiple watermark stored conflicts with database can be resolved by keeping record of time and date. The author having former stored entry will be regarded as the original author.

The watermark will be accurately detected by this algorithm in the absence of attack on text, and text document will be called authentic text without tampering. The watermark will get distorted in the presence of tampering attacks with text.

The distortion percentage of extracted watermark refers to tampering amount occurred by attacks on contents of attacked text document, this value represent in WDP which we can get for it by equation (6):

$$\mathrm{WDP} = \left|\frac{\mathrm{WM_A} - \mathrm{WM_O}}{\mathrm{WM_O}}\right| * 100 \quad \ldots\ldots\ldots\ldots \qquad (6)$$

The matching percentage between original and extracted watermarks can be calculated in WMP by equation (7):

$$\mathrm{WMP} = 1 - \mathrm{WDP} \quad \ldots\ldots\ldots\ldots \qquad (7)$$

The extraction algorithm is illustrated in figure 4.

DWM Extracting Algorithm (A Zero Text DWM based on Probabilistic Weights)

- **Input:** Original watermark WMo/Original Text Document, Attacked Text Document
- **Output:** WMo value, EWM value, WMP, WDP, and Attacked States and Transitions Matrix [61][63]

1. Read Original Text document to TD_A array.
2. Loop i = 1 to 37, // Build States matrix and compute the number of occurrences of the next state transitions from every present state to another
   - Loop j = 1 to 37,
     - Total Number of Transition[i][j]
3. Loop i = 1 to 37, // Compute Probabilistic Weights for each state
   - Loop j = 1 to 39,
     - Frequency += States [i] [j]
   - States [i] [38] = Frequency
   - States [i] [39] = 1/Frequency
   - Frequency = 0
   - WMo += States [i] [39] // Generate attacked watermark
4. IF $WM_A = WM_O$
   - Print "Document is authentic and no tampering occurred"
     - WDP = 1
   - Else
   - Print "Document is not authentic and tampering occurred"
     - WDP = $\left|\frac{WM_A - WM_O}{WM_O}\right| * 100$
     - WMP = 1 - WDP

WM_O: Original watermark, **WM_A:** Extracted/Attacked watermark, **aStates:** Attacked states and transitions matrix, **TD_A:** Attacked text document array, **WMP:** Watermark Matching Percentage, **WDP:** Watermark Distortion Percentage.

Fig. 4: Watermark detection algorithm

4. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed approach exploits Hidden Markov Model for text analysis in order to get the weights and interrelationships between contents of text document by finding the probability of occurrences of letters in a given text to generate a watermark and embeds it logically using zero-watermarking technique.

In this paper, we used 6 samples of variable size text as used in [19]. These samples have been collected from Reuters' corpus, which has been categorized into three classes according to their length, namely Small Size Text (SST), Medium Size Text (MST), and Large Size Text (LST).

A summary of the datasets used in [19] and the insertion and deletion attacks performed on these datasets is shown in table 1. For comparison purposes we designed our datasets and attacks similar to those performed in [19].

Table 1: Original and Attacked Text Samples with Insertion and Deletion Percentage [19]

Sample Text No	Original Text	Attack		Attacked Text
	Word Count	Insertion	Deletion	Word Count
1 : [SST2]	421	26%	25%	425
2 : [SST4]	179	44%	54%	161
3: [MST2]	559	49%	25%	696
4: [MST4]	2018	14%	12%	2048
5: [MST5]	469	57%	53%	491
6: [LST1]	7993	9%	6%	8259

To evaluate the performance of our approach we conducted a series of experiments that test our approach against the following well known attacks such as random insertion, and deletion of words and sentences. These kinds of attacks were applied at single and multiple locations in our datasets. The performance results of our approach under all the mentioned attacks are as follows:

4.1 Insertion Attack

We have performed localized and dispersed insertion attacks on each sample of our datasets. In what follows we present the performance of our proposed approach under the localized and dispersed insertion attacks.

4.1.1 Localized Insertion Attack

Single insertion of words or sentences was made randomly at a single location in the original text document. The percentage of localized insertion was selected randomly between 0.5 and 21%. The results are presented in table 2 and discussed below.

Table (2): Distortion of retrieved watermark with variable sizes of text under localized insertion attack

Sample Text No	IA Rate	WM_O	(WM_A)	WDP	WMP
[SST2]	6%	3.8642	3.0918	19.99%	80.01%
[SST4]	21%	8.5169	5.253	38.32%	61.68%
[MST2]	6%	2.7321	2.6816	1.85%	98.15%
[MST4]	1%	1.1184	1.1177	0.06%	99.94%
[MST5]	7%	2.6268	2.0836	20.68%	79.32%
[LST1]	0.5%	0.2648	0.2627	0.79%	99.21%

For each dataset in table 2, for example [MST2], the watermark(Content Weight) for the original text (before attack) was computed to be equal to 2.7321then under the insertion attack with rate 6% the retrieved watermark (Content Weight) was found to be 2.6816, which indicates that high tampering was detected. This shows that the proposed approach is very effective in detecting insertion tampering. The last column of the table gives the matching of the retrieved watermark (WMP) which is equal to 98.15 %. This value is close to one hundred percent which is the optimal value. For the rest of the datasets it can be seen from the table that the content weight changes due to tampering whether it's low or high.

4.1.2 Dispersed Insertion Attack

Multiple insertions of words or sentences were made randomly at a multiple locations in the original text document. The percentage of dispersed insertion was selected randomly between 1 and 40%. The results are presented in table 3 and discussed below.

Table (3): Distortion of retrieved watermark with variable sizes of text under dispersed insertion attacks

Sample Text No	IA Rate	WM_O	(WM_A)	WDP	WMP
[SST2]	17%	3.8642	3.5774	7.42%	92.58%
[SST4]	40%	8.5169	6.8455	19.62%	80.38%
[MST2]	11%	2.7321	2.4912	8.82%	91.18%
[MST4]	3%	1.1184	1.0926	2.31%	97.69%
[MST5]	16%	2.6268	1.8713	28.76%	71.24%
[LST1]	1%	0.2648	0.2643	0.19%	99.81%

Similar to localized insertion, multiple insertion of words or sentences was made randomly on each dataset in table 3, for example [MST2] which is the same dataset mentioned above in localized insertion, as seen from the table under multiple insertion attack with rate 11% that the retrieved watermark (content weight) changes which was found to be 2.4912 and the accuracy was found to be 91.18%, and the same with others datasets as seen from the table that the content weight changes due to tampering whether its low or high, which indicates that high tampering was detected. The accuracy of the retrieved watermark (WAR) is high with middle and large size of datasets and the accuracy is low with small dataset as shown in the last column of table 3. This shows that the performance of our proposed approach is high in detecting multiple insertions tampering with middle and large size of datasets due to insertion rate whether it's low or high.

4.2 Deletion Attack

We have performed localized and dispersed deletion attacks on each sample of our datasets. In what follows we present the performance of our proposed approach under the localized and dispersed deletion attacks.

4.2.1 Localized Deletion Attack

Single deletion of words or sentences was made randomly at a single location in the original text document. The percentage of localized deletion was selected randomly between 0.5 and 13%. The results are presented in table 4 and discussed below.

Table (4): Distortion of retrieved watermark with variable sizes of text under localized deletion attack

Sample Text No	DA Rate	WM_O	WM_A	WDP	WMP
[SST2]	5%	3.8642	3.9	0.93%	99.07%
[SST4]	13%	8.5169	9.3986	10.35%	89.65%
[MST2]	6%	2.7321	3.4782	27.31%	72.69%
[MST4]	3%	1.1184	1.1619	3.89%	96.11%
[MST5]	6%	2.6268	2.6964	2.65%	97.35%
[LST1]	0.6%	0.2648	0.2651	0.11%	99.89%

For each dataset in table 4, for example [MST2], under the localized deletion attack with the same rate of localized insertion which is 6% the retrieved watermark (Content Weight) was found to be 3.4782 with the accuracy value which is equal to 72.69%, which indicates that high tampering was detected. This shows that the proposed approach is very effective in detecting deletion and insertion tampering. For the rest of the datasets it can be seen from the table that the content weight changes due to tampering whether it's low or high.

4.2.2 Dispersed Deletion Attack

Multiple deletions of words or sentences were made randomly from a multiple locations in the original text document. The percentage of dispersed deletion was selected randomly between 5 and 26%. The results are presented in table 5 and discussed below.

Table (5): Distortion of retrieved watermark with variable sizes of text under dispersed deletion attacks

Sample Text No	DA Rate	WM_O	(WM_A)	WDP	WMP
[SST2]	12%	3.8642	4.2087	8.92%	91.08%
[SST4]	26%	8.5169	6.1519	27.77%	72.23%
[MST2]	10%	2.7321	3.002	9.88%	90.12%
[MST4]	5%	1.1184	1.1589	3.62%	96.38%
[MST5]	12%	2.6268	2.9613	12.73%	87.27%
[LST1]	6%	0.2648	0.308	16.31%	83.69%

Multiple deletion of words or sentences was made randomly on each dataset in table 5 with various deletion rate, for example [MST2] dataset, the retrieved watermark was changes and tampering was detected under multiple deletion attacks with the same approximately values of multiple insertion rate and accuracy as shown in table 3. Thus, the same with others datasets as seen from the table that the content weight changes due to tampering whether it's low or high. This shows that the performance of our proposed approach is high in detecting multiple deletions tampering due to deletion rate whether it's low or high.

4.3 Insertion and Deletion Attack

We have performed randomly insertion and deletion attacks in the same time on each sample of our datasets. We have applied our proposed approach using the same 6 variable text datasets used in [19] and [20] with insertion and deletion rates. The average matching and distortion of retrieved watermark on all samples of the text is shown in table 6.

Table (6): Distortion and matching of extracted watermark

Sample Text No	IA Rate	DA Rate	WM_O	WM_A	WDP	WM P
[SST2]	26%	25%	3.8642	3.9313	1.74%	98.26%
[SST4]	44%	54%	8.5169	6.7346	20.93%	79.07%
[MST2]	49%	25%	2.7321	1.9887	27.21%	72.79%
[MST4]	14%	12%	1.1184	1.1607	3.78%	96.22%
[MST5]	57%	53%	2.6268	2.6575	1.17%	98.83%
[LST1]	9%	6%	0.2648	0.2356	11.03%	88.97%

As shown in table 6 and graphically represented in figure 5, we can say the watermark distortion percentage is always greater than 1 with all text samples which indicate that watermark has been distorted as a result of text tampering and is not authentic. This proves that the text is sensitive to any modification made by various attackers and the accuracy of watermark gets affected even when the tampering volume is low.

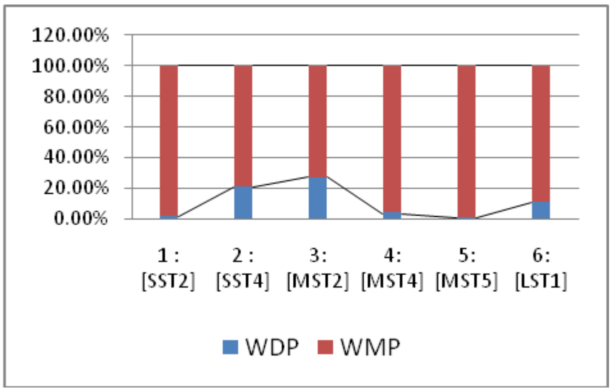

Fig. 5: watermark matching and distortion percentage for all text samples

The proposed algorithm can be used for content authentication and tampering or forgery detection in all open digital text documents. Even, the proposed algorithm does not prevent copying or prove copyright of text documents.

5. CONCLUSION

Text document is one of the most popular digital media exchanges in daily life especially via the internet. Thus, more and more texts need to be protected. None the proposed text watermarking approaches in the past are not robust against random tampering attacks and on all types of text documents. In this paper, We have proposed a novel zero-watermarking algorithm for content authentication of digital text documents, which using hidden Markov model to analyze the contents of text and utilizes it to generate a watermark since this watermark is logically embedded in the text, and the host text document is not altered to embed watermark, and this watermark is later extracted to detects tampering and verify the authenticity of text document. Performance of our proposed algorithm evaluated for random insertion, deletion, and reorder attacks in localized and dispersed form on 6 variable text datasets, and compared with other two recent approaches. Results show that our proposed algorithm is more secure and has better robustness to WLZW and CBZW especially on middle and large samples of text used in our experiments. Also, results show that our proposed approach always detects insertion and deletion tampering attacks occurred randomly on different size of text documents even when the tampering volume is very low, low, middle, or high, and detect re-order tampering attacks occurred on middle size of text documents.

This work can further be extended to include the high order level of hidden Markov model for natural language processing and analysis the contents of text document and utilize of it to generate a better watermark.

6. REFERENCES

[1] Z. JaliI, A. Hamza, S. Shahidm M. Arif, A. Mirza, A Zero Text Watermarking Algorithm based on Non-Vowel ASCII Characters. International Conference on Educational and Information Technology (ICET 2010), IEEE.

[2] Suhail M. A., Digital Watermarking for Protection of Intellectual Property. A Book Published by University of Bradford, UK, 2008.

[3] L. Robert, C. Science, C. Government Arts, A Study on Digital Watermarking Techniques. International Journal of Recent Trends in Engineering, Vol. 1, No. 2, pp. 223-225, 2009.

[4] X. Zhou, S. Wang, S. Xiong, Security Theory and Attack Analysis for Text Watermarking. International Conference on E-Business and Information System Security, IEEE, pp. 1-6, 2009.

[5] T. Brassil, S Low, and N. F. Maxemchuk, Copyright Protection for the Electronic Distribution of Text Documents. Proceedings of the IEEE, vol. 87, no. 7, July 1999, pp. 1181-1196.

[6] M. Atallah, V. Raskin, M. C. Crogan, C. F. Hempelmann, F. Kerschbaum, D. Mohamed, and S.Naik, Natural language watermarking: Design, analysis, implementation. and Proceedings of the a Fourth Hiding Workshop, vol. LNCS 2137, 25-27 , 2001.

[7] N. F. Maxemchuk and S Low, Marking Text Documents. Proceedings of the IEEE International Conference on Image Processing, Washington, DC, Oct 26-29, 1997, pp. 13- 16.

[8] D. Huang, H. Yan, Interword distance changes represented by sine waves for watermarking text images. IEEE Trans. Circuits and Systems for Video Technology, Vol.11, No.12, pp. 1237 1245, 2001.

[9] N. Maxemchuk, S. Low, Performance Comparison of Two Text Marking Methods. IEEE Journal of Selected Areas in Communications (JSAC), vol. 16 no. 4, pp. 561-572, 1998.

[10] S. Low, N. Maxemchuk, Capacity of Text Marking Channel. IEEE Signal Processing Letters, vol. 7, no. 12 , pp. 345 -347, 2000.

[11] M. Kim, Text Watermarking by Syntactic Analysis. 12th WSEAS International Conference on Computers, Heraklion, Greece, 2008.

[12] H. Meral, B. Sankur, A. Sumru, T. Güngör, E. Sevinç , Natural language watermarking via morphosyntactic alterations. Computer Speech and Language, 23, pp. 107-125, 2009.

[13] Z. Jalil, A. Mirza, A Review of Digital Watermarking Techniques for Text Documents. International Conference on Information and Multimedia Technology, pp. 230-234 , IEEE, 2009.

[14] M. AtaIIah, C. McDonough, S. Nirenburg, V. Raskin, Natural Language Processing for Information Assurance and Security: An Overview and Implementations. Proceedings 9th ACM/SIGSAC New Security Paradigms Workshop, pp. 5 1-65, 2000.

[15] H. Meral, E. Sevinc, E. Unkar, B. Sankur, A. Ozsoy, T. Gungor, Syntactic tools for text watermarking. In Proc. of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, pp. 65050X-65050X-12, 2007.

[16] O. Vybornova, B. Macq., Natural Language Watermarking and Robust Hashing Based on Presuppositional Analysis. IEEE International Conference on Information Reuse and Integration, IEEE, 2007.

[17] M. tallah, V. Raskin, C. Hempelmann, language watermarking and tamperproofing. Proc. of al.. Natural 5th International Information Hiding Workshop, Noordwijkerhout, Netherlands, pp.196-212, 2002.

[18] U. Topkara, M. Topkara, M. J. Atallah, The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym Substitutions. In Proceedings of ACM Multimedia and Security Conference, Geneva, 2006.

[19] Z Jalil, A. Mirza, H. Jabeen, Word Length Based Zero-Watermarking Algorithm for Tamper Detection in Text Documents. 2nd International Conference on Computer Engineering and Technology, pp. 378-382, IEEE, 2010.

[20] Z Jalil, A. Mirza, M. Sabir, Content based Zero-Watermarking Algorithm for Authentication of Text Documents. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010.

[21] Z. Jalil , A. Mirza, T. Iqbal, A Zero-Watermarking Algorithm for Text Documents based on Structural Components. pp. 1-5 , IEEE, 2010.

[22] S. Ranganathan, A. Johnsha, K. Kathirvel, M. Kumar, Combined Text Watermarking. International Journal of Computer Science and Information Technologies, Vol. 1 (5), pp. 414-416, 2010.