International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 2, Issue 5, Nov 2013, 1-10

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

A NOVEL CONTENT BASED ZERO WATERMARKING ALGORITHM FOR TAMPER-PROOFING PLAINTEXT DOCUMENT

SAKSHIKHULLAR & BIJENDRA SINGH

NSIT, Delhi, India

ABSTRACT

With the advent of internet, mobiles & other communication media, it has become extremely important to protect digital data against illegal copying, tampering, forgery, illicit redistribution etc. Digital watermarking provides authentication and tamper protection for multimedia contents over the internet. Text is an important medium travelling over the internet, since most digital content are as plain text. In this paper, we propose a novel content based zero-watermarking algorithm for tamper proofing text documents. A watermark gets generated based on the occurrence of vowels in the text. It is zero based since the algorithm generates watermark using the content of the text itself which gets registered with a certifying authority.

The plaintext file is divided into blocks. Within this block, the vowel count is calculated for each block. The algorithm automatically chooses the first two vowels it sees in the block. Keeping one constant, it measures the distance to each occurrence of second vowel's distance & averages it. The vowel sum is divided by this average distance to obtain W0 which is the first position in the watermark. And likewise the process is repeated for the remaining blocks to obtain the final Watermark WM. This watermark is then registered by the original owner to the CA. A trusted certifying authority is an essential requirement in this algorithm with whom, the original copyright owner registers his/her watermark. Whenever the content/text ownership is in question, this trusted third party acts as a decision authority.

Experimental results demonstrate the effectiveness of the algorithm against tampering attacks by identifying watermark distortion rates on 9 different samples. It has provided very satisfactory results against insertion, deletion, re-ordering and substitution attacks.

KEYWORDS: Authentication, Tampering, Text Document Security, Watermarking, Watermark Embedding, Watermark Extraction, Zero Watermarking

INTRODUCTION

The widespread use of Internet and other communication technologies has made it possible to reproduce, copy, tamper, and distribute digital contentillegally. Today, digital media is facing authentication, forgery, and copyright protection issues.

Digital watermarking is used as a solution to protect copyrights and authenticity of digital content. Digital watermarking solutions for images, audio, and video are already in existence but the number of solutions for plain text is quite less.

A digital watermark is a sort of a hidden information of the original owner which is embedded in the digital data. This hidden information can be later extracted by the original owner to check for tamper or authentication. This digital data can be an image, text, audio or a video.

Digital Watermarking is the process that embeds such a hidden information called a watermark into a multimedia object such that it can be detected or extracted later to make an assertion about its content

In general, any watermarking scheme (algorithm) consists of two parts.

- The watermark embedding algorithm
- The extraction &and comparator for verification.

In past a lot of research & work is done in the field of image, audio &video watermarking .The work done in the area of text watermarking is comparatively less. Text is such an important medium since most digital content such as web sites, e-books, papers, journals, news, emails and messages are as plain text.

Text being a specialized medium requires specialized copyright protection and authentication solutions. Traditional watermarking algorithms modify the contents of the digital medium to be protected by embedding a watermark. This traditional watermarking approach is not applicable for plain text. A specialized watermarking approach such as zero-watermarking would do the needful for text [14].

In this paper, we propose a novel zero watermarking algorithm which utilizes the contents of text itself for its authentication. A zero-watermarking algorithm does not change the original data, but utilize the original data to construct original watermark information [21].

PROPOSED ALGORITHM

We propose a zero-watermarking approach in which the host text document is not altered to embed watermark, rather the characteristics of text are utilized to generate a watermark [14]. The watermark is fragile in nature and is used to authenticate text documents. The watermark generation and extraction process is illustrated in Figure 1.

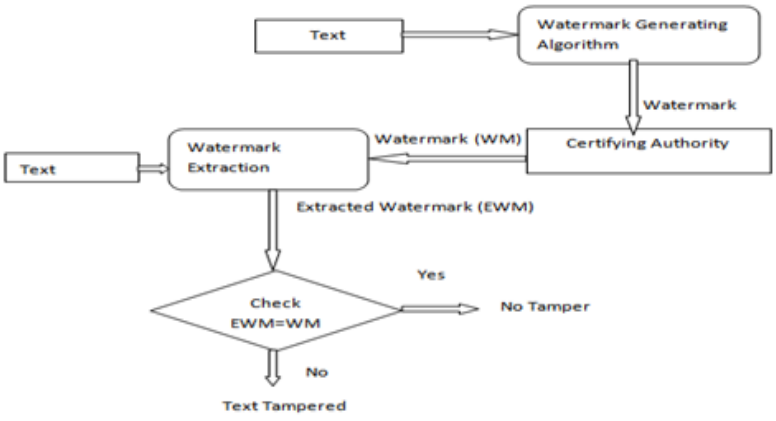

Figure 1: Flowchart of the Proposed Algorithm

It is a zero-watermarking scheme, since watermark is not actually contained in the text, rather it gets generated using the contents of text [14]. The watermarking process involves two stages:

- Embedding Algorithm
- Extraction Algorithm

Watermark embedding is done by the original owner and extraction done later by a Certifying Authority (CA) to prove ownership. The certifying authority is a neutral trusted party that registers the original copyright owner of the document [14]. It checks for tamper& authentication issues by acting as a decision making authority.

The algorithm can be broadly stated as:

- Initially read the text document.
- Divide the text document into block size of 1000 or 2000 characters depending on the text size.
- Count the occurrence of each of the vowels in their respective blocks & add them. This done because it's very likely, that even if a slight tampering is done on the text, the vowel count is likely to change. Taking advantage of this, it makes our algorithm very efficient against insertion, deletion, & substitution attack.
- Now choose any (e.g. first two vowels in the experiment) two vowels to measure their average inter-space distance within a block while keeping one vowel constant & taking it as a reference This is done to prevent against reordering attacks. This distance between is likely be changed because of change in order.
- Divide the sum with this avg. inter vowel distance to obtain W0 which is the first position in the watermark and repeat the procedure for the remaining blocks.
- Output the watermark & register it with the certifying authority.

Watermark is registered with the Certifying Authority (CA) and is used is the extraction algorithm to authenticate text document.

Embedding Algorithm

The algorithm which embeds the watermark in the text is called embedding algorithm. The watermark embedding algorithm requires original text file as input. The plaintext file is then divided into a 2000 character blocks. Within this block, the vowel count is calculated for each block. The algorithm automatically chooses the first two vowels it sees in the block. Keeping one constant, it measures the second vowel's distance & averages it. The vowel sum is divided by this average distance to obtain W0 which is the first position in the watermark. The process is repeated for the remaining blocks to finally obtain a watermark array. This watermark is then registered by the original owner to the CA, along with the author name, current date & time.

```
1.Read the text T &WM
2. Partition T into 2000 character blocks B.
3. Initialize vowel_occurrence=0; first_occurrence= '\0', second_occurrence='\0';
4. B-count=total no. of blocks B in the text.
5.for (inti=0; i<B-count) repeat steps (5.1 to 5.3)
5.1 for (int j=0; j<block_size) repeat
vowel_occurrences++;
            if (first_occurrence == '\0')
first_occurrence = first seen vowel
else if
(f (first_occurrence!='\0'&&second_occurrence=='\0')
ssecond_occurrence = second seen vowel;
5.2 foreach (int position in table[second_occurrence])
difference += position - table[first_occurrence][0];
5.3 WM [i]= vowel_occurence /
( (difference) / table[second_occurence].Count);
6. Output the watermark WM.
```

T: plaintext file, **B-count:** no. of blocks, **block_size:** size of block B, **vowel_occurrence:** total vowel occurrence in a block, **first_occurrence:** first seen vowel, **second_occurrence:** second seen vowel, **table[second_occurrence]:** lists positions of the second seen vowel, **table[first_occurrence]:** position of the first seen vowel, **WM :** watermark.

Extraction Algorithm

The algorithm which extracts the watermark from the text is called extraction algorithm. The proposed extraction algorithm takes the plain text and the original watermark as the input. The watermark is generated from the text by the

extraction algorithm and is then compared with the original watermark registered with the CA. The text may be attacked or un-attacked. In case of an attack the extracted watermark would not match the original one, hence such a text tampering can be detected. The watermark will get distorted in the presence of tampering attacks with text. Tampering can be insertion, deletion, paraphrasing or reordering of words and sentences in text [14]. However, the text would be called as un-attacked in case the extracted watermark matches the original one. The extraction algorithm is as follows:

```
1.Read the text T &WM.
2. Partition T into 2000 character blocks B.
3.Initialize vowel_occurrence=0; first_occurrence='\0',second_occurrence='\0';
4. B-count=total no. of blocks B in the text.
5.for (int i=0; i<B-count) repeat steps.(5.1 to 5.3)
5.1 for (int j=0; j<block_size) repeat
vowel_occurrences++;
 if (first_occurrence == '\0')
first_occurrence = first seen vowel
else if
(f (first_occurrence!='\0'&&second_occurrence=='\0')
ssecond_occurrence = second seen vowel;

5.2 foreach (int position in table[second_occurrence])
difference += position - table[first_occurrence][0];
5.3 EWM [i]= vowel_occurence /
( (difference) / table[second_occurence].Count);

6. Output the watermark EWM.
7.          if (EWM not equals WM)
Tamper = YES
          else
Tamper=NO
8. Output EWM.
EWM: Extracted Watermark
```

EXPERIMENTAL RESULTS

We used 9 samples of variable sized text .These samples have been collected from e-books, project reports and web pages. We have performed 3 kinds of experiments / tests.

- Random insertion & Deletion attacks on 8 texts of different sizes.
- Taking a constant text size (e.g. 1000 words) in our experiments & then analyzing the tampering at different percentages of insertion & deletion attacks,
- In our third set of experiments we have analyzed tampering due to re-ordering attacks

Tamper Distortion Rate (TDR)

Tampering is analyzed in terms of this factor. This is the mean square error between the original watermark & the extracted watermark. It is a relative measure. Higher the value of TDR, greater is the original watermark distorted or changed.

Tamper Analysis at Random Insertion and Deletion Attacks on Different Text Sizes

Insertion and deletion of words and sentences was performed at multiple randomly selected locations in text. We have chosen two small size text [SST], two medium size text [MST], two Large Size texts [LST] & two Very Large size texts [VLST].

[Table1] shows the sample label number as in dataset ,number of words in original text, the insertion and deletion volume, and the number of words in the text after attack.

Table 1: Tamper Analysis at Random Insertion and Deletion Attacks on Different Text Sizes

S. No.	WC	Attacked Text		WCA	TDR	T.D.
		INS	DEL			
SST1	196	4%	11%	182	0.00034966	YES
SST2	239	5%	8%	229	0.0000173	YES
MST1	531	14%	20%	512	0.00001366	YES
MST2	421	39%	7%	558	0.00011261	YES
LST1	1198	8%	3%	1260	0.00025328	YES
LST2	1158	19%	13%	1229	0.00018105	YES
VLST1	4647	3.5%	4%	4632	0.0005493	YES
VLST2	5927	4%	6%	5812	0.00068700	YES

S. No: Sample No., **WC:** Original Text Word Count, **WCA:** Attacked Text Word Count, **TDR:** Tamper Distortion Rate, **TD:** Tamper Detected, **INS:** Insertion Attack, **DEL:** Deletion Attack

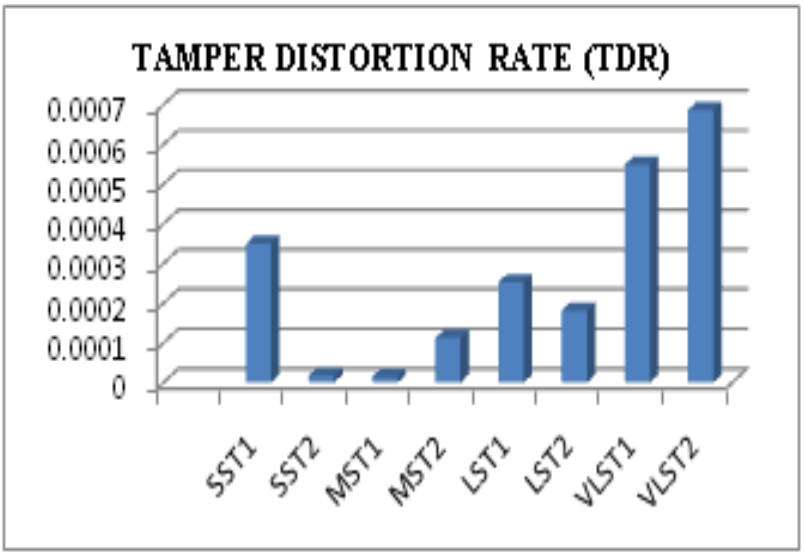

Figure 2: Watermark Distortion Rate at Random Attacks on Different Text Sizes

As it can be quite clearly seen in the table and the graph that tampering is detected by our algorithm even for a small insertion & deletion attack at [VLST1] or for a large insertion attack in [MST2]. Interestingly, the TDR is very high even for small tampering attacks. The algorithm has performed exceptionally well for very large size texts, where in small attacks have caused a large change in the watermark.

Tamper Analysis at Different Percentages of Insertion & Deletion Attacks

In this experiment, we have taken a standard text size of 1000 words [STA], on which we have performed insertion & deletion attacks at different percentages. [Table 2] shows the TDR for deletion attacks on the Standard text [STA] of 1000 words

Table 2: Tamper Analysis at Different Percentages of Deletion Attacks

S No.	WC	DEL	WCA	TDR	TD
1.	STA(1000)	2%	980	0.00047863	YES
2.	STA(1000)	5%	950	0.00064670	YES
3.	STA(1000)	10%	900	0.00045335	YES
4.	STA(1000)	15%	850	0.00075988	YES

S. No.: Sample No., **WC:** Original Text Word Count, **WCA:** Attacked Text Word Count, **TDR:** Tamper Distortion Rate, **TD:** Tamper Detected

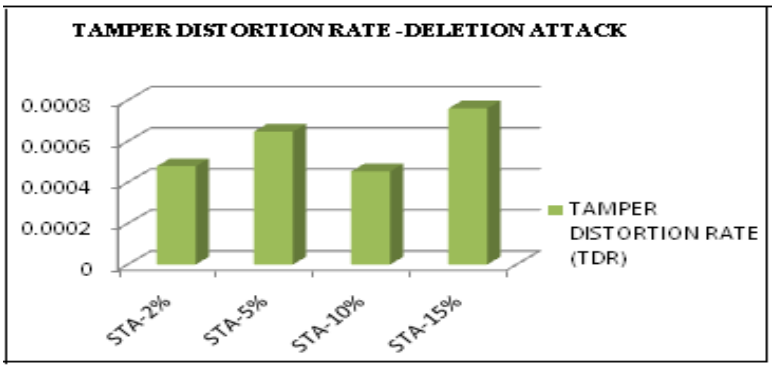

Figure 3: W.D.R of STA [1000] at Different Percentages of Deletion Attacks

As it can be clearly seen with the exception of STA-10%, that as the deletion attack % increases, the TDR also increases, however, at 10% deletion, this is not valid. Factors like deletion of numeric data or a larger dispersed attack can be responsible.

[Table 3] shows the TDR for insertion attacks on the Standard text [STA] of 1000 words

Table 3: Tamper Analysis at Different Percentages of Insertion Attacks

S No.	WC	INS	WCA	TDR	TD
1.	STA(1000)	2%	1020	0.00011018	YES
2.	STA(1000)	5%	1050	0.00021325	YES
3.	STA(1000)	10%	1100	0.00045335	YES
4.	STA(1000)	15%	1150	0.00046555	YES

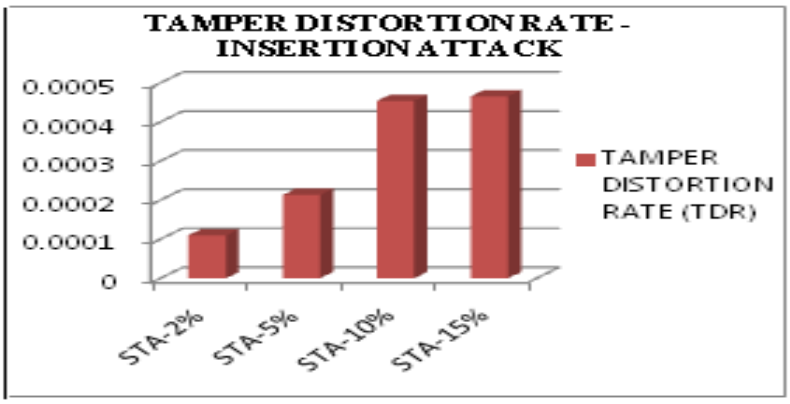

Figure 4: W.D.R of STA [1000] at Different Percentages of Insertion Attacks

As it can be observed, that TDR increases with an increase in Insertion Attack %.

[Table 4] analyzes tampering caused due to both the insertion & deletion attacks on the text. The same STA [1000] is being used.

Table 4: Tamper Analysis at Different Percentages of Deletion & Insertion Attacks

S.No	Original Text	INS	DEL	TDR	TD
1.	STA(1000)	2%	2%	0.00059868	YES
2.	STA(1000)		5%	0.00017992	YES
3.	STA(1000)		10%	0.00741323	YES
4.	STA(1000)	5%	2%	0.00066543	YES
5.	STA(1000)		5%	0.00085028	YES
6.	STA(1000)		10%	0.00029088	YES
7.	STA(1000)	10%	2%	0.00057754	YES
8.	STA(1000)		5%	0.00035169	YES
9.	STA(1000)		10%	0.00045467	YES

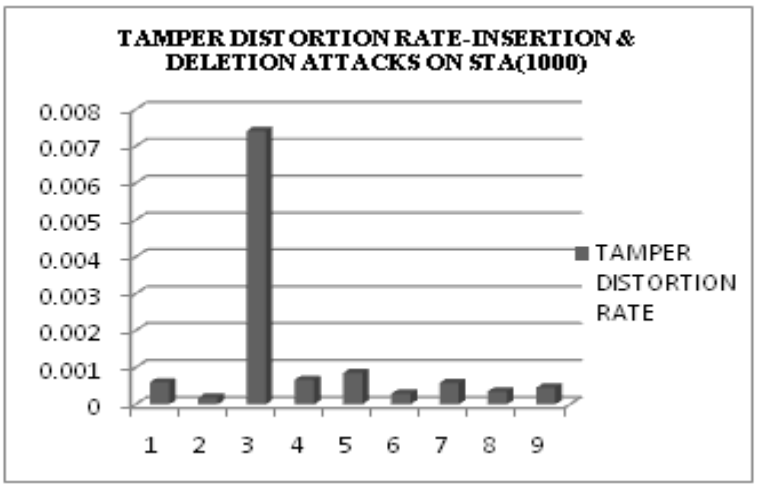

Figure 5: W. D. R of STA [1000] at Different Percentages of Insertion &Deletion Attacks

With an exception of test case 3, we observe that the watermark distortion rate is averagely distributed in the range 0.00 to 0.001.The algorithm is able to detect a very small tampering as in test case 1 with a very good distortion rate. It can be clearly observed that watermark distortion rate is very high even when the insertion and deletion volume is low. Text is sensitive to any modifications made by the attacker. High distortion rate indicates that the text has been tampered and is not authentic. This proves that the accuracy of watermark gets adversely affected even with minor tampering and watermark fragility proves that text has been attacked.

Tamper Analysis at Different Percentages of Re-Order Attacks

In our third set of experiments we see how well the algorithm performs against re-ordering attacks. We have chosen 5 texts of different sizes & changed the order of words/ sentences in that text. [Table 5] shows the results.

Table 5: Tamper Analysis at Different Percentages of Re-Order Attack

S No.	WC	Re-Order	WCA	TDR	TD
1.SST2	239	57%	239	0.00013443	YES
2.MST2	421	4%	421	0.00001474	YES
3.STA	1000	5%	1000	0.00000171	YES
4.LST1	1198	7%	1198	0.00022015	YES
5.VLST1	4647	13.5%	4647	0.0004699	YES

S. No.: Sample No., **WC:** Original Text Word Count, **WCA:** Attacked Text Word Count, **TDR:** Tamper Distortion Rate, **TD:** Tamper Detected, **RE-ORDER:** Re-order attack

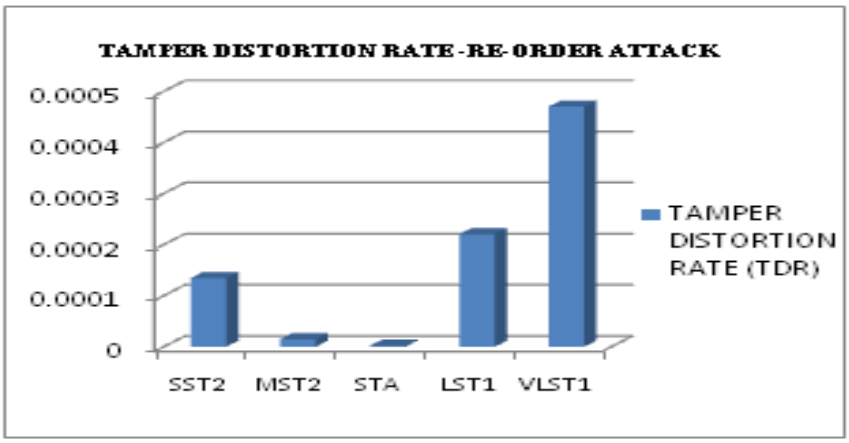

Figure 6: W.D.R at Different Percentages of Re-Ordering Attacks

Our algorithm is able to detect attacks caused by changing the order of words/ sentences in the text. Even in a very small re-order attack of 4% in MST2 it's able to detect tamper. The above experiments have shown that the algorithm performs exceptionally well under insertion, deletion, substitution & re-ordering attacks. It can work for all sizes if text & detect both minor & major attacks

CONCLUSIONS & FURTHER WORK

The existing text watermarking solutions for text authentication are not applicable under random tampering attacks and on all types of text. With the small amount of attack, it becomes impossible to identify the existence of attack and to prove authenticity of information. In this paper, we have developed a zero-text watermarking algorithm, which utilizes the contents of text to generate a watermark and this watermark is later extracted to prove the authenticity of text document. We have evaluated the performance of the algorithm for 3 types of experiments. Firstly, we performed Random tampering attack in dispersed form on 9 variable size text samples. Secondly, we analyzed the performance under different percentages of insertion & deletion attacks. Finally, the algorithm's performance was analyzed in case of re-ordering attacks. Results show that the algorithm always detects tampering even when the tampering volume is low.

REFERENCES

1. X. Zhou, Z. Wang, Zhao, S. Wang, "Performance Analysis and Evaluation of Text watermarking", IEEE, 2009.
2. Frank Hartung, Martin Kutter, "Multimedia Watermarking Techniques", Proceedings of the IEEE, Vol. 87, No. 7, pp.1085 – 1103, July 1999.
3. Z. Jalil, A. M. Mirza, "A Review of Digital Watermarking Techniques for Text Documents", IEEE, 2009.
4. J. 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.
5. S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document Identification for Copyright Protection Using Centroid Detection," IEEE Transactions on Communications, Mar. 1998, vol. 46, no.3, pp 372-381.
6. Brassil J, Low S H, Maxemchuk N F, "Copyright Protection for the electronic Distribution of Text Documents" Proceedings of the IEEE, July 1999,
7. N. F. Maxemchuk, S. H. Low, "Performance Comparison of Two Text Marking Methods," IEEE Journal of Selected Areas in Communications (JSAC), May 1998. vol. 16 no. 4 1998. pp. 561-572.
8. D. Huang and 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, Dec 2001.
9. M. J. Atallah, V. Raskin, M. C. Crogan, C. F. Hempelmann, F.Kerschbaum, D. Mohamed, and S.Naik, "Natural language watermarking: Design, analysis, and a proof-of-concept implementation", Proceedings of the Fourth Information Hiding Workshop, vol. LNCS 2137, 25-27 April 2001, Pittsburgh, PA.
10. Hassan M. Meral et al.,"Natural language watermarking via morphosyntactic alterations", Computer Speech and Language, 23,107-125, 2009.
11. M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, "Natural Language Watermarking and Tamper proofing", Fifth Information Hiding Workshop, vol. LNCS, 2578, October, 2002, Noordwijkerhout, The Netherlands, Springer-Verlag.

12. Xingming Sun, Alex JesseyAsiimwe, "Noun-Verb Based Technique of Text Watermarking Using Recursive Decent Semantic Net Parsers", Lecture Notes in Computer Science (LNCS) 3612: 958-961, Springer Press, August 2005.
13. Z. Jalil, A. M. Mirza ,"Text Watermarking Using Combined Image-plus- Text Watermark" , IEEE, 2010.
14. ZuneraJalil, Anwar M. Mirza and Maria Sabir, "Content based Zero-Watermarking Algorithm for Authentication of Text Documents",(IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, February 2010.
15. ZuneraJaliI, HamzaAziz,Saad Bin Shahid, Muhammad Arif and Anwar M. Mirza , "A Zero Text Watermarking Algorithm based on Non-Vowel ASCII Characters", 2010 International Conference on Educational and Information Technology (ICEIT 2010).
16. Jaseena K.U., Anita John, "Text Watermarking using Combined Image and Text for Authentication and Protection", International Journal of Computer Applications (0975 – 8887) Volume 20– No.4, April 2011.
17. Z. Jalil, A. M. Mirza ,"An Invisible Text Watermarking Algorithm Using Image Watermark", Innovations In Computing Science and Software Engineering, 2010.
18. Youn-Won Kim et al., "A text watermarking algorithm based on word classification and inter-word space statistics", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03), 2003.
19. Shao Xiwen, "The application of digital watermarking technology in the field of e-commerce", 2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering.
20. http://en.wikipedia.org/wiki/Watermarking..
21. Anbo Li, Bing-xian Lin, Ying Chen, "Study on copyright authentication of GIS vector data based on Zero-watermarking", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. VII. Part B4, pp.1783-1786, 2008.
22. Zhou, Xinmin , Zhao, Weidong, Wang, Zhicheng, Pan, Li , "Security theory and attack analysis for text watermarking", 2009 International Conference on E-Business and Information System Security, EBISS 2009.