A Novel MS-Word Text Document Multiple Watermarking Technique for Authentication and Copyright Protection

Ayush Arora[#1], Drishti Agarwal[#2], Jeebananda Panda[#3]

[#1, 2, 3]*Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, India*

Abstract*—In today's time when everyone is surrounded by digital data, the development of internet-related technologies, like cloud databases, social media platforms and much more, is rapidly increasing which raises the problem of information security. Today anyone can easily generate identical copies of any text document. So, to protect the data or the content i.e., intellectual property of someone, some measures are needed to verify the authenticity of one's work. Copyright protection is one of the most important and difficult challenges for the researchers. So, for verification, many researchers have proposed several algorithms to embed a watermark in a text document (In simple words embedding a watermark means hiding some secret information in the document which is hard to detect). The proposed approach uses the application class property of MS-Word document, RGB colour values of text and spacing between the lines to hide the watermark in a monocoloured MS-Word text document. Several different documents are used for evaluation which got attacked by different types of changes.*

Keywords — *MS-Word Document, watermarking, copyright, vowels, security.*

I. INTRODUCTION

Nowadays, the digitalisation of information is becoming more and more common. In this modern digital world, the spread of the information in the digital format is rapidly increasing. Advancement in internet technologies is one of the factors that encourages this increase. Various forms of data such as audio, video, text and images can be quickly communicated between two points in this era. With this ease comes the concern of securing this information from unauthorized users and protecting the authenticity of the digital work. There are a number of ways which are used by hackers today to retrieve the data making it susceptible to manipulation and fraud. One of the solutions to this problem is digital watermarking which hides a secret message into the digital content and hence plays a vital role in copyright protection and authentication of the content. The idea is to place a secret message inside the text document without losing the meaning of the digital content which later on can be used for verification of ownership. The proposed method uses multiple levels of text watermarking for the same.

In the current pandemic scenario of COVID 19, mostly all schools and universities will have to take online examinations to cope up with the same. Hence, there would be an increase in the digital content. Watermarking the answer sheets of students would help them claim their authenticity. Since the proposed algorithm gives an easy and fast watermark embedding way, hence the algorithm is very useful in the current scenario.

Every organisation, small or big, has certain important text documents associated with it which are integral to its working. The problem with the existing techniques is that they are not robust enough and less secure. The challenge is two-folded. First is to detect the presence of any formatting attacks, and second is to prove ownership of the document. A new framework is proposed here which deals with these challenges and is more robust, imperceptible and easy to embed.

Main points covering contribution to this research are:

1. The work proposes a new digital text watermarking technique which uses certain properties of both application class and document class of a monocoloured MS Word Document to embed a watermark.
2. Apart from these properties, the count of certain alphabets is used for embedding the watermark thus ensuring a quick detection of any kind of formatting attack.
3. The proposed work is robustup to 99.99% against formatting attacks and is highly secured.
4. Any attacker would be unable to remove the watermark without strongly altering the watermarked document.

The rest of the paper is structured as follows: Section 2 shows the recent and related work done in the field. Section 3 deals with the detailed description of the proposed method followed by section 4 where results obtained are discussed. Section 5 concludes the paper and gives direction to the future work.

II. RELATED WORK

Watermarking in text documents is an active area of research. There is a lot of work that has been done in watermarking of audio, video and image files[1] [2]. In case of text documents some interesting work has been done by many scholars using various approaches. Kim *et al.* [3] has proposed a technique in which they grouped adjacent words to form segments and then classified the segments using word class information, then they hide the information in these segments using spacing between the words of the same class whereas Alotaibi *et al.* [4] has used the pseudo-space that is very small space between two parts of the same word they have altered this space to store the binary bits in the document. Jaman*et al.* [5] proposed an algorithm using natural language processing, and in their work, they have used various grammatical rules like verbs, conjunction, preposition, modals and articles for watermarking [6]. Some scholars have used space between words and lines to hide watermark [7] [8][9].

Iqbal *et al.* [10] uses the special properties of MS-Word documents that cannot be altered by any formatting or change in content of the document [11] [12], in their approach they hide the encrypted key to insert in these special properties. Khadam *et al.* [13] has also used these special properties to hide they watermark, in their proposed work they have generated an encrypted message, then they convert them to a binary string followed by converting to a decimal string, after which they divide that string to four equal parts followed by taking logarithms of these number, whose value they insert in the top, right, left and bottom margin of the MS-Word document.

Rizzo *et al.* [14] has proposed a technique in which they substitute Latin symbols with homoglyph symbols to prevent any visual change in the content of the text document they have used different Unicode symbols for substitution which looks nearly the same as of the symbol to be substituted, in proposed work they embed their secret password by substitution of the Latin symbols by homoglyph symbols. [15] glyphs are also used to hide the watermark.

Afify*et al.* [16] has proposed a hybrid approach which they have used the concept of natural language processing along with structural technique and hashing. They have used the most occurring word in the document to embed their watermark in the document. They have altered the space in the left of the word to insert a bit in that word along with this watermark. They have also used MD5 hashing technique to hash the complete document to check if the document has been tampered or not.

III. THE PROPOSED METHOD

In this section, the detailed algorithm and framework used for embedding watermarks in a text document is presented followed by its extraction. The document class of MS Word has many properties (WSP) which can be used to store the watermark without actually affecting the contents of the document. Some of these properties have been exploited to embed the multi-level watermark. The proposed technique is highly robust. The entire procedure is divided into three sections: The first section deals with the method of generating watermark followed by the embodiment of this watermark in the document and then finally checking against any formatting attack by extracting the watermark. Figure 1 shows the complete architecture of the text watermark system.

A. Generating Watermark:

This section describes how the watermarks to be embedded in the document are generated. Three different types of watermarks are generated. For the first type, the aim was to embed a part of the generated watermark in as many words as possible. For this, research was done and it was found that in most cases 100% of the document can be covered if the watermark is hidden using vowels and the letter ‘y’. Algorithm 1 is used for the same. The entire document is iterated over and the individual counts of all the five vowels along with the letter ‘y’ is stored. Binary strings corresponding to these counts are randomly generated. These binary strings serve the purpose of the hidden information that has to be embedded in the text document. The second type of watermark is generated by taking in account the number of lines in the document. A binary string of length equal to the number of lines in the document is generated and then embedded. The third watermark will be an encrypted key. Figure 2 shows the generation of watermark for a text document and Algorithm 1 describes the pseudo code for the same.

B. Embedding Watermark:

In the proposed approach, multi-level embedding of watermarks is being performed. The watermarks generated in part A are to be embedded in the text document. This is done using two properties of the document class. Algorithm 2 describes the entire procedure for embedding watermark.

The watermarks generated against the count of the vowels and the letter ‘y’ are stored in the document by using the font colour of the document. Each of the values 0 and 1 are assigned a particular font colour value. As the document is iterated over, the font colour of the vowels and the letter ‘y’ are changed as per the corresponding character of the binary string. This process is followed for all the six letters under consideration. Thus, using the font

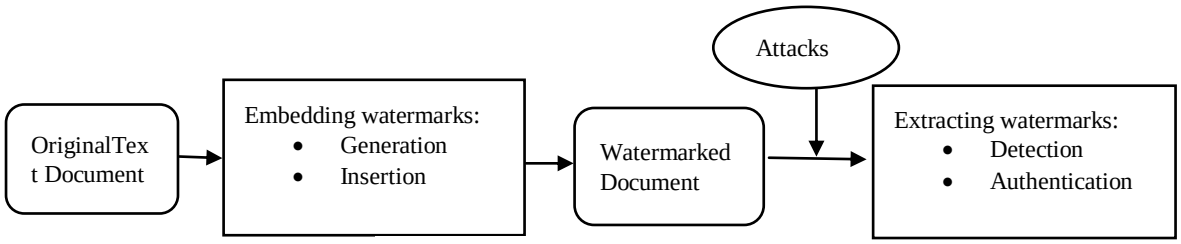

Figure 1: Architecture of TextWatermark

colour of the text document six watermarks are embedded.One of the major advantages of applying this technique is that it covers almost every word of the document, hence even a minor change in the content of the document would be easily detectable.

Algorithm 1: Watermark Generation

```
Input: MS-Word Doc (T)
Output: s_a, s_e, s_i, s_o, s_u, s_y, s_line // Secret binary string
Start:
Data: c_a, c_e, c_i, c_o, c_u, c_y // Count of character a, e, i, o, u, y
Variable Declaration:
        num_char = Number of characters in T
        c_line = number of lines in T
Initialization:
        c_a = 0, c_e = 0, c_i = 0, c_o = 0, c_u = 0, c_y = 0
        for(i = 1 to num_char) do
            curr_char = T[i]
            if(curr_char == 'a' or curr_char == 'A') then
                c_a = c_a + 1
            elif(curr_char == 'e' or curr_char == 'E') then
                c_e = c_e + 1
            elif(curr_char == 'i' or curr_char == 'I') then
                c_i = c_i + 1
            elif (curr_char == 'o' or curr_char == 'O') then
                c_o = c_o + 1
            elif(curr_char == 'u' or curr_char == 'U') then
                c_u = c_u + 1
            elif(curr_char == 'y' or curr_char == 'Y') then
                c_y = c_y + 1
            end if
        end for
        s_a <- Random binary string of length c_a
        s_e <- Random binary string of length c_e
        s_i <- Random binary string of length c_i
        s_o <- Random binary string of length c_o
        s_u <- Random binary string of length c_u
        s_y <- Random binary string of length c_y
        s_line <- Random binary string of length c_line

        end
```

To ensure extra security, the second type of generated watermark is embedded in the document using another property of the document class which is the space between the lines. Again 0 and 1 are assigned different values and the entire document is iterated over each line. Corresponding to the value of the character in the string, space between the lines is altered.This second type of watermark embedding still allows to prove ownership of the digital data even if some formatting attack changes the font colour of the document. The third kind of watermark which is inserted is the encrypted key. It is inserted using the application class of MS-Word document and remains unaffected by any change in the content and formatting of the document.Apart from ownership verification, this step becomes crucially important in the case when some formatting attack alters both the font colour and spacing between lines of the entire document. After the watermarks are embedded, the watermarked document is in the MS Word format only.

Algorithm 2: Watermark Embedding

```
Input: MS-Word Doc (T), s_a, s_e, s_i, s_o, s_u, s_y, s_line, Secret Key (SK)
Output: Watermarked Document(T')
Start:
Data:
        RGB_1, RGB_2 // RGB color values
        Space_1, Space_2 // Line space values
Variable Declaration:
        num_char = Number of characters in T
        c_line = number of lines in T
        WSP = MS-Word Special Property
Initialization:
        i_a = 0, i_e = 0, i_i = 0, i_o = 0, i_u = 0, i_y = 0
        WSP = SK
        for(i = 1 to num_char) do
            curr_char = T[i]
            if(curr_char == x) then // where x is a, e, i, o, u, y
                if(s_x[i_x] == '0') then
                    T[i].Color = RGB_1
                else
                    T[i].Color = RGB_2
                end if
                i_x = i_x + 1
            end if
        end for
        for(i = 1 to c_line) do
            if(s_line == '0') then
                T.Line[i].Space = Space_1
            else
                T.Line[i].Space = Space_2
            end if
        end for

        end
```

C. Watermark Extraction:

Watermark extraction is the reverse process of embedding watermark. It is the step where any possible format attacks are detected and ownership of the document is proved. The hidden watermark is extracted from the text document and compared with

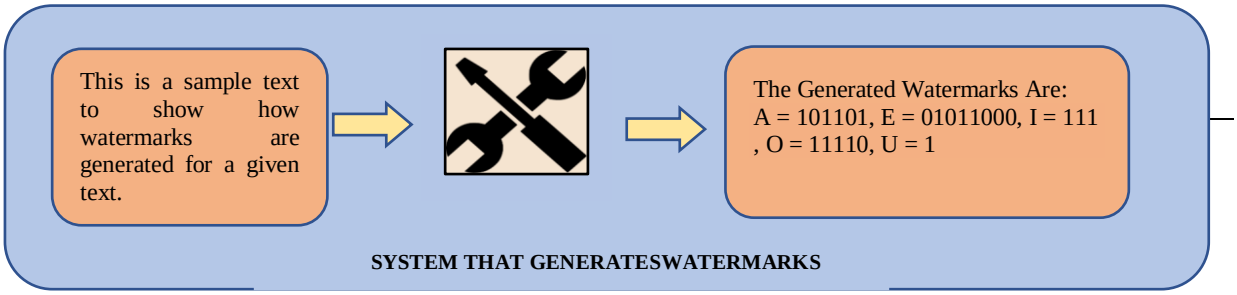

Figure 2: Generating Watermarks

the embedded watermark for the same. The detailed process is described in Algorithm 3.

The system iterates over the entiredocument and stores the colour code of the five vowels and the letter 'y' in the form of binary strings. Hence six binary strings are generated from the document which are then compared with the original binary strings that were embedded in the document.Any kind of change in the content of the document generates a different binary key of one or more letters, thus marking the presence of some formatting attack. Second type of watermark is extracted by iterating over all the lines of the document. Again, a binary string is generated from the input document depending on the value of the line spaces which is then compared with the string embedded in the document. Third type of watermark is extracted from the application class property of MS-Word document. This helps the user acclaim his/her ownership of the document.

Algorithm 3: Watermark Extraction

```
Input: MS-Word Doc (T')
Output: s_a, s_e, s_i, s_o, s_u, s_y, s_line, SK
Start:
Data: RGB_1, RGB_2 // RGB color values
        Space_1, Space_2 // Line space values
Variable Declaration:
        num_char = Number of characters in T'
        c_line = number of lines in T'
        WSP = MS-Word Special Property
Initialization:
        i_a = 0, i_e = 0, i_i = 0, i_o = 0, i_u = 0, i_y = 0
        SK = WSP
          for(i = 1 to num_char) do
            curr_char = T'[i]
            if(curr_char == x) then // where x is a, e, i, o, u, y
              if(T'[i].Color == RGB_1) then
                s_x += '0'
              elif(T'[i].Color = RGB_2) then
                s_x += '1'
        else
                throw error('Document is tempered')
              end if
            end if
          end for
          for(i = 1 to c_line) do
            if(T.Line[i].Space = Space_1) then
              s_line += '0'
            elif(T.Line[i].Space = Space_2) then
              s_line += '1'
            else
```

IV. RESULTS AND DISCUSSION

Multiple experiments were performed to test the system. Following two parameters were used as the basis of evaluation:

1. Robustness
2. Imperceptibility

The first two types of watermark embedded are text dependent whereas the third type of watermark that is inserted for the purpose of experiments is "This is a secret message."

A. Robustness

Any document is subject to several types of formatting attacks. A system is said to be robust if it is able to recover its embedded watermark even after such attacks. The multilevel watermarking done helps achieve a very high rate of robustness because even if the first two watermarks are lost, the third one is recovered as it is. The two parameters used to compute robustness are Pattern matching rate (PMR) and Watermark Distortion Rate (WDR) and are stated as follows:

$$PMR = \frac{No.\,of\ patterns\ matched\ correctly}{No.\,of\ watermark\ patterns}$$

$$WDR = 1 - PMR$$

The proposed technique is 99.99% robust against formatting attacks. Table I shows the robustness value for various documents. The table values were obtained after applying all types of formatting attacks to the documents. Formatting attacks like bold, italics, underline, font size, font type were applied and it was found that all types of watermarks were recovered with 100% accuracy. A message showing "First two watermarks recovered. Third watermark is: This is a secret message" pops up. However, change in font colour destroys one of the watermarks and change in spacing between lines removes the other type of watermark. Yet the third type of watermark was always recovered making the system highly robust against formatting attacks. Figure 3 shows two cases: one is when all three watermarks are recovered and the other is when one of the first two watermarks are destroyed but third is recovered as it is.

.

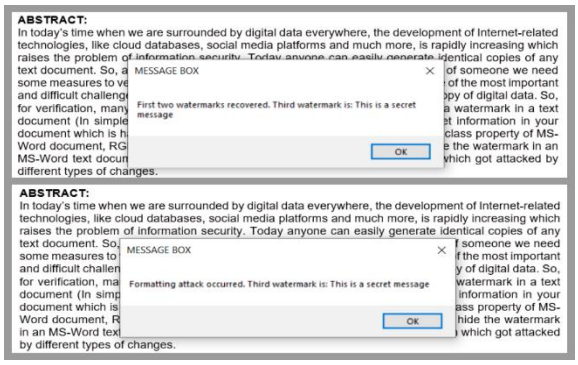

Figure 3: Top Figure: When all three Watermarks are Recovered. Bottom figure: When one of the First Two Watermarks is Lost but Third is Recovered as it is.

Table I: Robustness Achieved for Various Documents

DOCUMENT	ROBUSTNESS ACHIEVED
Document 1	100%
Document 2	100%
Document 3	99.99%

B. Imperceptibility

A system is said to be imperceptible if it allows embedding watermarks in a text document without affecting the contents of the documents. It is both a primary and fundamental requirement of the watermark. The two measures used to measure imperceptibility are Peak Signal to Noise Ratio (PSNR) and Similarity (SIM) percentage which are formalized as follows:

$$PSNR = 20log_{10}\frac{O_{doc}(Max)}{RMSE}$$

$$SIM = [1 - \frac{RMSE}{O_{doc}(Max)}] \times 100$$

where O_{doc}(Max) is a maximum pixel value in a document and RMSE (root mean squared error) is calculated as follows:

$$RMSE = \sqrt{\frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[O_{doc}(i.j) - W_{doc}(i,j)]^2}$$

Figure 4 shows the images of the document before and after embedding the watermarks. The images can be seen to be similar showing the high imperceptibility of the system. Extensive experiments were done to measure PSNR and SIM values. To calculate the PSNR and SIM values a python system has been used in which the original image and the watermarked image of the document are loaded as a matrix by OpenCV. O_{doc}(max) value is the maximum value of the matrix. NumPy is used to find the RMSE value followed by PSNR and SIM values.The PSNR values were found to be between 33.499 and 45.821. And SIM values were found between 97.886 and 99.2. The acceptable value of PSNR should be above 30 to attain a high level of imperceptibility. Table II shows PSNR and SIM values for different documents.

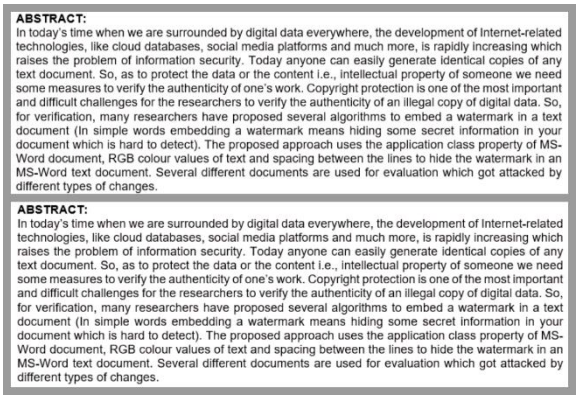

Figure 4: Top figure: Original document. Bottom figure: Watermarked Document.

Table II: PSNR and SIM Values for Different Documents

DOCUMENT	PSNR	SIM
Document 1	33.499	97.886
Document 2	38.467	98.13
Document 3	45.821	99.2

V. CONCLUSION AND FUTURE WORK

A robust and secure method is proposed to authenticate the digital contents as the system embeds multiple watermarks. Several techniques are proposed in this field. But the proposed system achieves higher levels of security as it uses multiple level of watermarks. Through experiments it was seen that the proposed work is low in capacity but highly robust, imperceptible and also fast in embedding of watermark. In the future, the work will be extended for detection of all types of the formatting attacks and location of changes in the document. Apart from detection, it is proposed to come up with a system that can get back the original document. And specifically, for online examination due to COVID 19 pandemic scenario, watermarking can be performed on images of diagrams provided by the studentto claim their authenticity.

REFERENCES

[1] Mayer, Joceli, Paulo VK Borges, and Steven J. Simske. "*Text Watermarking.*" In Fundamentals and Applications of Hardcopy Communication, pp. 43-113. Springer, Cham, 2018.

[2] Nematollahi, Mohammad Ali, Chalee Vorakulpipat, and Hamurabi Gamboa Rosales. "*Text Watermarking.*" In *Digital Watermarking*, pp. 121-129. Springer, Singapore, 2017.

[3] Kim, Young-Won, Kyung-Ae Moon, and Il-Seok Oh. "*A Text Watermarking Algorithm based on Word Classification and Inter-word Space Statistics.*" In *ICDAR*, pp. 775-779. 2003.

[4] Alotaibi, Reem A., and Lamiaa A. Elrefaei. "*Utilizing word space with pointed and un-pointed letters for Arabic text watermarking.*" In 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim), pp. 111-116. IEEE, 2016.

[5] Jaman, Khandokar Nafis, Zannatun Nayem, Bristi Rani Roy, Nayreet Islam, Faisal R. Badal, and Subrata K. Sarker. "*A Text Watermarking Algorithm Developed Using Natural Language Processing.*" (2019).

[6] Yingjie, Meng, Liu Huiran, Shang Tong, and Teng Xiaoyu. "*A zero-watermarking scheme for prose writings*." In 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 276-282. IEEE, 2017.

[7] Kuribayashi, Minoru, Takuya Fukushima, and Nobuo Funabiki. "*Robust and Secure Data Hiding for PDF Text Document.*" IEICE TRANSACTIONS on Information and Systems 102, no. 1 (2019): 41-47.

[8] Taha, Ahmed, Aya S. Hammad, and Mazen M. Selim. "*A high capacity algorithm for information hiding in Arabic text.*" Journal of King Saud University-Computer and Information Sciences (2018).

[9] Liang, Ooi Wei, and VahabIranmanesh. *"Information hiding using whitespace technique in Microsoft word."* In 2016 22nd International Conference on Virtual System & Multimedia (VSMM), pp. 1-5. IEEE, 2016.

[10] Iqbal, Muhammad Munwar, Umair Khadam, Ki Jun Han, Jihun Han, and Sohail Jabbar. "*A Robust Digital Watermarking Algorithm for Text Document Copyright Protection based on Feature Coding*." In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1940-1945. IEEE, 2019.

[11] Ghafir, Ibrahim, Jibran Saleem, Mohammad Hammoudeh, Hanan Faour, Vaclav Prenosil, Sardar Jaf, Sohail Jabbar, and Thar Baker. "*Security threats to critical infrastructure: the human factor*." The Journal of Supercomputing 74, no. 10 (2018): 4986-5002.

[12] Muzammal, Syeda Mariam, Munam Ali Shah, Hasan Ali Khattak, Sohail Jabbar, Ghufran Ahmed, Shehzad Khalid, Shahid Hussain, and Kijun Han. *"Counter measuring conceivable security threats on smart healthcare devices."* IEEE Access 6 (2018): 20722-20733.

[13] Khadam, Umair, Muhammad Munwar Iqbal, Muhammad Awais Azam, Shehzad Khalid, Seungmin Rho, and Naveen Chilamkurti. "*Digital Watermarking Technique for Text Document Protection Using Data Mining Analysis*." *IEEE Access* 7 (2019): 64955-64965.

[14] Rizzo, Stefano Giovanni, Flavio Bertini, and Danilo Montesi. "*Content-preserving text watermarking through unicode homoglyph substitution*." In Proceedings of the 20th International Database Engineering & Applications Symposium, pp. 97-104. 2016.

[15] Xiao, Chang, Cheng Zhang, and Changxi Zheng. "*Font Code: Embedding information in text documents using glyph perturbation.*" ACM Transactions on Graphics (TOG) 37, no. 2 (2018): 1-16.

[16] Afify, Amira Eid, Ahmed Emran, and Ahmed Yahya. *"A Tamper proofing text watermarking shift algorithm for copyright protection*." Arab Journal of Nuclear Sciences and Applications 52, no. 3 (2019): 126-133.