Fingerprint Scheme For Digital Text

Mohamed Sameh Hassanein
School of Information Systems, Computing and Mathematics Brunel University
London, United Kingdom

Gheorghita Ghinea
School of Information Systems, Computing and Mathematics Brunel University
London, United Kingdom

Abstract

Digital content copyright protection is a key concern in the current environment where intellectual property is constantly threatened. In this paper, a novel solution is proposed to guarantee authenticity and integrity of a text document by protecting digital text. The key application of the proposal is the algorithm that protects text against alteration and tampering under "Cut & Paste" attacks. This approach provides a promising solution to current digital security challenges, through providing a starting point towards secured sharing and usage of digital content. The paper also examines key protection techniques, their uses and their drawbacks. Some of the key techniques discussed include forms of cryptography, steganography, watermarking and 2D barcodes.

1. Introduction

Could text have its own fingerprint that identifies itself and protects its content? Literature depends on language that explains its content in the form of text. Could text generate identifiers that represent a unique legal entity? With the incremental use of the internet, copyright protection has become a significant concern particularly for IP holders. Their primary goal is the protection of audio, video and printed documents from alteration, plagiarism, piracy and illegal distribution. This led the academic community to develop solutions for copyright protection. This paper provides an algorithm to protect text from alteration and tampering under "Cut & Paste" attacks. So far, significant research has been undertaken and many applications have been developed to ensure text protection [1-3, 5]. Despite all these efforts, the battle of text copyright protection is far from being over. Most solutions depend on encryption algorithms, text steganography and text watermarking techniques, which will be discussed in details in the following sections.

2. Related Work

2.1. Cryptography and Stenography

The power of cryptography lies in the fact that the resources needed to break any algorithm might vastly exceed the potential value gained for any attacker. Despite this power, cryptography cannot be an ultimate solution for the copyright problem due to multiple reasons. Cryptography provides security during transmitting the text via communication channels yet not post storage. This means that the digital content will be secure while travelling through the medium, but once it is decrypted, it will not be protected e.g. against alteration [4]. It is in the human nature to crack anything, which is kept secret, even without a logical reason, [6] ('if I am unable to see it then no one should be able to use it'). This means that cryptography would draw attention because its existence means that there is hidden information. Other techniques such as steganography and watermarking are used to avoid the drawbacks of cryptography while hiding information. Steganography techniques are quite similar to watermarking techniques and they will be illustrated later in depth. Steganography is a branch of information hiding technology, which aims to embed some information into a digital content [7]. The purpose of embedding information in digital multimedia is to assure the copyright protection through encoding some information such as the author's name, publisher's name, and so on. These methods and techniques used in text steganography are similar to those used in digital watermarking, since both belong to the same discipline, information hiding. The word steganography means "concealed writing" and comes from the Greek words "Steganos" i.e. "protected " and "graphei" i.e. "writing" [8]. Steganography is an ancient art of embedding private messages in seemingly innocuous text in a way that prevents the detection of the secret messages by a third party [8]. The main advantage of steganography compared to cryptography is that a third party could not suspect the presence of a hidden message; therefore, it would not draw any unnecessary attention [9].

2.2. Digital watermarking

Komatsu and Tominaga were the first to introduce the "Digital watermarking" term in 1988[10]. Around 1995, the interest in digital watermarking started to boom due to the scarcity of copyright protection techniques for digital content. Watermarking is the process of embedding special data into media such as image, audio, and video. The embedded information, known as a watermark, can be extracted from the multimedia contents later and used for protecting the ownership [11]. This process mainly inserts unique information that refers to the copyright owner or the originator of the content, substituting a traditional handwritten signature. Digital watermarking can either be visible or invisible, depending on the identification code embedded in the data [11]. In other words, if the mark (i.e. the digital object that identifies the intellectual property or the ownership of digital format inserted in the multimedia) can be recognized; then it is visible.

Invisible watermarking is preferable as it reduces the chance of attacking the digital watermark.

The watermarking techniques for text documents are limited compared to other digital multimedia. Text is of binary nature and is composed of blocks, lines, word patterns, separation between foreground and background. The different components of text, with the addition of structure, style and font, translates into specific meaning. Text properties determine the structure of the text-watermarking algorithms [12]. Digital text watermarking can be categorized into three main groups: image-based, syntactic and semantic approaches [11] as shown in Figure 1.

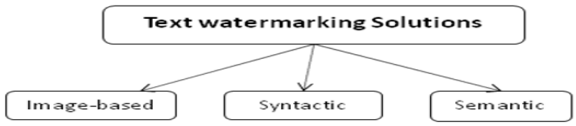

Figure 1. Text watermarking techniques

- *Image-based approach*

Brassil et al. proposed three methods to watermarking text documents applying different image-based techniques [1-3]: line-shift coding, word-shift coding and feature coding. In line-shift coding, even lines are slightly shifted upward or downward according to the required value. If the bit is zero, the line shifts downwards. Otherwise, the line shifts upwards. The odd lines act as control lines; therefore, no shifting to them. These control lines plays the role of references for measuring and comparing the distances between lines during the decoding phase. Decoding is accomplished by comparing the distance between the controlled lines and the shifted lines. Since the lines in the original document have a uniform space, the original document is not needed during the decoding phase. Similarly, in word- shifting coding, words are shifted horizontally (right and left) by modifying the space between consecutive words. Unlike line-shifting coding, word-shifting coding operates in a non-blind manner during detection, as it requires the existence of the original document as the inter-word spacing is not constant. In feature coding, some text features are modified in a certain way such as the pixel of a character to denote zeros and ones. A proposed text-watermarking algorithm approach uses the occurrences of double letters (aa-zz) in the text to embed the watermark [13].

In related work, Huang and Yan developed an algorithm based on an average inter-word distance in each line [14]. The adjusted distance is set according to a specific phase and frequency of a sine wave. However, those techniques are not robust to a simple retyping (copy paste to notepad) or font changing attack [15]. Those watermarking schemes could be effective in hard copy documents and not in a digital text document.

- *Syntactic approach*

II. Text is composed of sentences and sentences have different syntactic structures, which depend on language and its conventions. Watermarking text could happen through applying syntactic transformation on text structure [11]. Atallah et al. were the first to propose the natural language watermarking scheme using the syntactic structure of text [16]. Here, Natural Language Processing (NLP) techniques are used to analyze the syntactic and the semantic structure of text while performing any transformations to embed the watermark bits [11]. In related work, Hassan et al. proposed the natural language watermarking algorithm by performing morph syntactic alterations to the text [17]. Authors state that agglutinative languages, derived from the Latin verb agglutinare, which means "to glue together", are easier to watermark than the English language. The watermarking solution is thus applicable to agglutinative languages like Turkish, Korean, Arabic, and Urdu, since these languages provide space for watermark embedding. Syntactic solutions for English language are insufficient because the presentation of its letters can't be glue together.

- *Semantic approach*

Atallah et al. proposed the first semantic-watermarking scheme in 2000 [18] through manipulation of text meaning representation. Semantic watermarking is hiding information in the content of text documents by linguistic transformations, such as semantic transformations, syntactic transformations and synonym substitution. Synonym substitution was later proposed by replacing certain words with their synonyms without affecting the context of text. This method is based on the different spelling of words in English language between UK and US. For example, "centre" (UK) and "center" (US) or "Petrol" (UK) and "Gas" (US) [19]. The disadvantage in this approach is that it is language-dependent and does not give a practical complete text watermarking solution. This approach is not resilient to the random synonym substitution attacks. Moreover, synonyms may not always give the exact meaning of the word; therefore, affecting the value of text. Due to the sensitive nature of some documents such as legal documents, quotes, poetry, etc; text-watermarking using syntactic and semantic approaches have limited applicability and usability.

Most of the techniques and methods used to protect text are dependent on the format and structure of text which can be easily defeated by a simple rewriting attack. The rewriting attack does not need a professional to do it because it is trivial to edit the font or spacing within a document. Although there are many novel ideas that depend on the language properties, they are not sufficient to ensure text documents security. The concept of securing the content of a text-based document in order to guarantee its authenticity and integrity could be accomplished by using the characteristics and properties of the language. In other words, the security mechanism should be language- dependent to protect any alteration in its own content. Hence, text fingerprinting could be a security mechanism to ensure its integrity and authenticity. This represents the focus of the work proposed in this paper.

Table I Text Fingerprint Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	0	P	Q	R	S	T	U	V	W	X	Y	Z	SP
L.C.B																											
L.C.A																											
T.L.I																											

3. Text Fingerprint Algorithm & analysis

This section exemplifies the design and implementation of an algorithm that captures and links the content of a text based document in a matrix. The two dimensional matrix presents the content of a text document making it easy to detect any form of alteration to the original content of the text. The first dimension is the twenty-six characters of the English alphabet and space (SP). The second dimension is the relationship between the characters and their position within the text. Each character reports on the number of occurrences of the consecutive character in the alphabet based on 3 dimensions.

1) L.C.B (Letter Count Before): counting the number of occurrence of the consecutive letter in text before the first occurrence of the current letter;

2) L.C.A (Letter Count After): counting the number of occurrence of the consecutive letter in text after the first occurrence of the current letter;

3) T.L.I (Total letter Index): the summation of a letter positions in text.

Table I explains the relationship between a subset of the text which is 27 characters and three logical relationships. These logical relationships aim to use the subset of text to report reference points and to save the positions of characters among text. These three logical relationships use each letter to report on the next letter according to its position in the text, while Total letter Index is a counter for the positions that the letter has occupied in text. For example:

(A, L.C.B):- counts the number of B characters occurring before the first A in text

(A, L.C.A):- counts the number of B characters occurring after the first A in text

(A, T.L.I):- Σ of the position of B characters in text

(SP, L.C.B):- counts the number of A characters occurring before the first SP in text

(SP, L.C.A):- counts the number of A characters occurring after the first SP in text

(SP, T.L.I):- Σ of the position of SP characters in text

In other words, all characters in the text are linked to each other according to their position among the text. This matrix, therefore, presents the relationship between the letters in the text and acts like a map for the content of a text base document.

The proposed algorithm for generating the Text Fingerprint matrix of presented in Figure [2].

This matrix will be used as the original reference to the document as the 27 letters present facts about relationships between words, strings and substrings in the text. It also stores the position of the letters in text in each letter T.L.I. Any alteration to the original text content could be detected. Furthermore, it could be analyzed to know what exactly has changed through using the detecting alteration algorithm, presented in Figure [3].We illustrate the robustness of our text fingerprint algorithm with two scenarios featuring "Cut & Paste" attacks on text documents. Highlighting how the authenticity and the integrity of text are preserved by Text Fingerprint algorithm (Table I).

1. Capture the first occurrence of the 26 alphabets and space in the document if found; if not found it sets the unfounded alphabet to negative one i.e. not present
2. Store the value of the captured 26 alphabets and space in a linked list (index)
3. Create the matrix presented in Figure 3
4. Get the character and its position from the text document
5. Add the character position(value) to the character T.L.I
6. Compare the character position to the previous alphabet stored in the linked list. If < previous alphabet increment L.C.B.; else increment L.C.A
7. Get next character and it's position from the document
8. If not end of document go to step 5
9. Else matrix is completed

Figure 2. Text Fingerprint algorithm

1. Generate the fingerprint matrix
2. Retrieve the original generate matrix
3. Compare both matrices if they are identical or not:
   a. Subtract the original matrix from the generated matrix
   b. Compare the difference (new matrix) to the original matrix
4. If not identical; then the original text document has been altered
5. If the document has been altered; the algorithm will highlight the altered text
6. Analyzing Text violation through analyzing :-
   a. L.C.A and L.C.B (letters has been added/removed)
   b. T.L.I (position of letters in original text has changed)

Figure 3. Fingerprint alteration detection algorithm

Given the original text: Alice lends Bob money.

Text Fingerprint will receive the associated text index in Table (II) as an input to generate the original fingerprint matrix using Text Fingerprint Algorithms mentioned in Figure 2. Table (III) presents the processed input after executing steps 4 & 5 in Text Fingerprint Algorithms mentioned in Figure 2. The output after executing all steps in Text Fingerprint Algorithmsin Table IV.

Table II. The associated text index

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Text	A	l	i	c	e		l	e	n	d	s		B	o	b		m	o	n	e	y	.

Table III. Characters first occurrence and their positions in the associated text

Character	A	B	C	D	E	I	L	M	N	O	S	Y	sp
1st occurrence	1	13	4	10	5	3	2	17	9	14	11	21	6
Position	1	13,15	4	10	5,8,20	3	2,7	17	9,19	14,18	11	21	6,12,16

Table IV. Original fingerprint matrix

Chars	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	sp
L-C-B	0	1	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
L-C-A	2	0	1	1	0	0	0	1	0	0	2	1	1	2	0	0	0	1	0	0	0	0	0	1	0	3	0
T-L-I	28	4	10	33	0	0	0	3	0	0	9	17	28	32	0	0	0	11	0	0	0	0	0	21	0	34	1

Table V. The associated text index

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
B	O	b		l	e	n	d	s		A	l	i	c	e		m	o	n	e	y	.

Table VI. Characters first occurrence and their positions in the associated text

Chars	A	B	C	D	E	I	L	M	N	O	S	Y	SP
1st occurrence	1	1	14	8	6	13	5	17	7	2	9	21	4
Position	11	1,3	14	8	6,15,20	13	5,12	17	7,19	2,18	9	21	4,10,16

Table VII. Generated fingerprint matrix

Chars	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	SP
L-C-B	2	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
L-C-A	0	1	0	2	0	0	0	1	0	0	2	1	1	1	0	0	0	1	0	0	0	0	0	1	0	3	0
T-L-I	4	14	8	41	0	0	0	13	0	0	17	17	26	20	0	0	0	11	0	0	0	0	0	20	0	28	11

Table IX. The associated text index

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
A	l	i	c	e		l	e	n	d	s		A	l	e	x		m	o	n	E	Y	.

Table X. Characters first occurrence and their positions in the associated text

Chars	A	C	D	E	I	L	M	N	O	S	X	Y	SP
1st occurrence	1	4	10	5	3	2	18	9	19	11	16	22	6
Position	1,13	4	10	5,8,15,21	3	2,7,14	18	9,20	19	11	16	22	6,12,17

Table XI. Generated fingerprint matrix

Chars	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	SP
L-C-B	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
L-C-A	0	1	1	2	0	0	0	1	0	0	3	1	1	1	0	0	0	1	0	0	0	0	1	1	0	3	1
T-L-I	0	4	10	49	0	0	0	3	0	0	23	18	29	19	0	0	0	11	0	0	0	0	16	22	0	35	14

Scenario (1): Bob lends Alice money.

The word "Bob" and "Alice has been swapped using "Cut & Paste" attack on the original text base document. Then the text will be Bob gave Alice money. All chars associated within text did not change but their position in text. This will leading to a different understanding from the original context. The result of the attack is Alice could be forced to pay twice. The attack changed the liabilities, meaning Alice cannot claim back her money and might be forced to pay it again. The text fingerprint proposed in this paper can prevent such an attack. This will be illustrated in Table VIII using the Fingerprint alteration detection algorithm. Text Fingerprint will receive the associated text index in Table V as an input to generate the fingerprint matrix using Text fingerprint algorithms in Figure 2

Table VI presents the processed input in table V after executing steps 4 & 5 in text fingerprint algorithms. Text fingerprint algorithms after executing all steps will produce the Generated fingerprint matrix in Table VII, while the original Fingerprint matrix in Table IV was kept safe and unaltered is retrieved. The difference between the matrices in Table IV and VII will show what happened to the original text, by analyzing the changes that the generated fingerprint did to the predefined relationship in the original fingerprint matrix. These changes will be highlighted in the analysis in Table VII after executing all the steps in Figure 3.

Table VIII Fingerprint alteration detection algorithm

Associated text index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Original text	A	l	i	c	e		L	e	n	d	s		B	o	b		M	o	n	e	Y

Altered text		B	o	b		l	e	N	d	s		A	l	i	c	e		m	o	n	e	y
Operation	Action																					
T-L-I (SP)	-10	A																				
L-C-B (A)	2	A																				
L-C-A (A)	-2	A		(B)																		
T-L-I (A)	-28	Missing two Bs. Combination =28																				
L-C-B (B)	-1	Char "C" has shifted 10 positions forward in text, since there is only one char "C" in text. Char "C" came before the first occurrence of char "B" and not the opposite. T-L-I= 4.																				
L-C-A (B)	1																					
T-L-I (B)	+10	A		(B)	C																	
L-C-B (C)	1	Char "D" has shifted 2 positions backward in text, since there is only one char "D" in text. Char "D" came after the first occurrence of char "C" and not the opposite. T-L-I= 10.																				
L-C-A (C)	-1																					
T-L-I (C)	-2	A		(B)	C					D												
L-C-B (D)	-1	Char "E" has 3 letters in text two at least have change their position in text. The combinations of 3 "E"s result in increase in T-L-I by 8.																				
L-C-A (D)	1																					
T-L-I (D)	+8	A		(B)	C		(E)				D					(E)					(E)	
T-L-I (H)	+10	Char "I" has shifted 10 positions forward in text, since there is only one char "I" in text. Char "I" came in T-L-I= 3.																				
		A		I	C		(E)				D					(E)					(E)	
L-C-A (K)	0	Char "L" has 2 letters in text. The combinations of 2"L"s result in increase in T-L-I by 8.																				
T-L-I (K)	+8	A		I	C	(L)	(E)				D		(L)			(E)					(E)	
L	0	A		I	C	(L)	(E)				D		(L)			(E)		M			(E)	
T-L-I (M)	-12	Char "N" has 2 letters in text. The combinations of 2 "N"s result in decrease in T-L-I by 12.																				
		A		I	C	(L)	(E)	(N)			D		(L)			(E)		M		(N)	(E)	
L-C-B (N)	1	Char "O" has 2 letters in text. Fingerprint matrix says "O"s came after the 1st "N". Shift "O" before 1st "N" 12 positions if empty.																				
L-C-A (N)	-1																					
T-L-I (N)	-12	A		I	C	(L)	(E)	(N)			D		(L)		(O)	(E)		M	(O)	(N)	(E)	
T-L-I (R)	-2	Char "S" has shifted 2 positions backward in text, since there is only one char "S" in text. Char "S" came in T-L-I= 11.																				
		A		I	C	(L)	(E)	(N)			D	S	(L)		(O)	(E)		M	(O)	(N)	(E)	
X	0	A		I	C	(L)	(E)	(N)			D	S	(L)		(O)	(E)		M	(O)	(N)	(E)	Y
T-L-I (Z)	-6	Char "SP" has 3 elements in text. The combinations of 3 "SP"s result in decrease in T-L-I by 6. One "SP" missing																				
		A		I	C	(L)	(E)	(N)		()	D	S	(L)		(O)	(E)		M	(O)	(N)	(E)	Y
Letters A, I, C, D, M,S, Y are in there right place, since letters O, N, E between M & Y then they are at their place. Position 2, 5, 6,7,8,9,12,13,14 and15 might be altered. Position 16 is a (SP) before the word money.																						
Exclude right letters		A		I	C	(L)	(E)	(N)		()	D	S	(L)		(O)	(E)		M	O	N	E	Y
Char "O" T-L-I =32-18=14. . Position 2, 5, 6, 7,8,9,12,13 and 15 might be altered.																						
Allocate "N"		A		I	C	(L)	(E)	(N)		()	D	S	(L)		O	(E)		M	O	N	E	Y
Char "N" T-L-I =28-19=9. Position 2, 5, 6, 7, 8, 12, 13 and 15 might be altered.																						
Allocate "B"		A		I	C	(L)	(E)			N	D	S			O	(E)		M	O	N	E	Y
Char "B" T-L-I= 28. The only combination is 13 and 15. Position 2, 5, 6, 7, 8 and 12 might be altered.																						
Allocate "L"		A		I	C	(L)	(E)			N	D	S		B	O	B		M	O	N	E	Y
Char "L" T-L-I= 9. The only combination is 2 and 7. Position 5, 6, 8 and 12 might be altered.																						
Allocate "E"		A	L	I	C		(E)	L		N	D	S		B	O	B		M	O	N	E	Y
Char "E" T-L-I= 33-20=13 The only combination is 5 and 8. Position 6 and 12 might be altered.																						
Allocate "SP"		A	L	I	C	E		L	E	N	D	S		B	O	B		M	O	N	E	Y
Char "SP" T-L-I= 34-16=18 The only combination is 6 and 12. Text altered and changes have been detected.																						
Done		A	L	I	C	E		L	E	N	D	S		B	O	B		M	O	N	E	Y

Scenario (2): Alice lends Alex money.

The word "Bob" in text has been removed and the word "Alex" is inserted. Text is injected by a third entity (Alex) while second entity (Bob) is removed. Editing the content of text is another kind of "Cut & Paste" attack on text base documents. Most chars associated within text did not change, except letters for the word (B, O, B) replace with letters for the word (A, L, E, X). This will leading to a different understanding from the original context. The result of the attack is that Alice could lose their money that she gave to Bob. The attack changed the liabilities, meaning Alice cannot claim her money from Bob because she gave it to someone else. The text fingerprint proposed in this paper can prevent such an attack. This will be illustrated in Table XII using the Fingerprint alteration detection algorithm. Text Fingerprint will receive the associated text index in Table IX as an input to generate the fingerprint matrix using Text Fingerprint Algorithms mentioned in Figure 2.

Table X presents the processed input in Table IX after executing steps 4 & 5 in text fingerprint algorithms. Text fingerprint algorithms after executing all steps will produce the Generated fingerprint matrix in Table XI, while the original Fingerprint matrix in Figure IV was kept safe and unaltered is retrieved. The difference between the matrices in Table IV and XI will show what happened to the original text, by analyzing the changes that the generated fingerprint did to the predefined relationship in the original fingerprint matrix. These changes will be highlighted in the analysis in Table XII after executing all the steps in Figure 3.

Table XII Fingerprint alteration detection algorithm

Associated text index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Original text	A	L	I	C	E		L	E	N	D	S		B	O	B		M	O	N	E	Y	.

Altered text		A	L	I	C	E		L	E	N	D	S		A	L	E	X		M	O	N	E	Y
Operation	Action																							
L-C-A(SP)	1	Letter "A" injected in text after the 1st occurrence of char "SP" and there is only "A" in text at position 1. Char "A" at position 13 inserted.																						
T-L-I (SP)	+13	A																						
L-C-A(A)	-2	Missing two Bs. Combination =28																						
T-L-I (A)	-28	À																						
L-C-B (B)	1	Char "C" has a referencing problem due to missing of letter B but position did not change.																						
L-C-A (B)	-1	A			C																			
C	0	A			C						D													
L-C-A (D)	1	Letter "E" injected in text after first char "D" in text.																						
T-L-I (D)	+15	A			C	E			E						(E)						(E)			
H	0	A		I	C	E			E						(E)						(E)			
L-C-A (K)	1	Letter "L" injected in text. T-L-I=9 while altered T-L-I= 23. Extra "L" inserted at position 13 (23-9=14)																						
T-L-I (K)	13	A	L	I	C	E		L	E		D					(E)						(E)		
T-L-I (L)	1	Letter "M" shifted 1 position forward																						
		A	L	I	C	E		L	E							(E)		M				(E)		
T-L-I (M)	1	One of the two Ns shifted 1 position forward																						
		A	L	I	C	E		L	E	(N)	D					(E)		M			(N)	(E)		
L-C-A (N)	-1	Missing one O. Two O's Combination =32																						
T-L-I (N)	-14	A	L	I	C	E		L	E	(N)	D					(E)		M		(O)	(N)	(E)		
L-C-A (W)	1	Letter "x" injected at position 16																						
T-L-I (W)	16	A	L	I	C	E		L	E	(N)	D					(E)		M		(O)	(N)	(E)	(Y)	
T-L-I (X)	+1	Letter "Y" shifted 1 position forward, since it comes after Shift letters "E, N and O" BY 1 position forward and position 18 is empty because of M. Shift the four letters backward by 1.																						
R	0	A	L	I	C	E		L	E	(N)	D	S												
		A	L	I	C	E		L	E	(N)	D	S				(E)		M	O	N	E	Y		
T-L-I (Z)	+1	One of the 3 Char "SP" shifted 1 position forward at position 16(34-6-12=16)																						
		. Position 9, 13, 14 and 15 might be altered																						
Allocate "SP"		A	L	I	C	E		L	E	(N)	D	S				(E)		M	O	N	E	Y		
Char "B" T-L-I= 28. The only combination is 13 and 15. Position 9 and 14, might be altered.																								
Allocate "B"		A	L	I	C	E		L	E	(N)	D	S		B		B		M	O	N	E	Y		

	Since char "e" replace by char "b" and current char "e" position make T-L-I= 0(33-5-8-20=0). Position 9 and 14 might be altered.																				
Validate "E"	A	L	I	C	E		L	E	(N)	D	S		B		B		M	O	N	E	Y	
	Since char "N" T-L-I= 28(28-19-9=0). Position 14 might be altered.																				
Validate "N"	A	L	I	C	E		L	E	N	D	S		B		B		M	O	N	E	Y	
	Missing one O. Combination =32. The combination is 18 and 14																				
Allocate "O"	A	L	I	C	E		L	E	N	D	S		B	O	B		M	O	N	E	Y	
Done	A	L	I	C	E		L	E	N	D	S		B	O	b		M	O	N	E	Y	

The complexity of the relationship and its applied analysis presents a solution for text copyright protection by depending on the style of the author in writing and the mathematical operations to create a fingerprint. The mathematical operations includes creating logical relationships, treating the elements of text (characters) as numbers in a matrix and keeping a zipped reference copy of the original contexts in text. This information extracted from the key could grant author's intellectual property and enforce copyright protection in text base document.

4. Implementation and experimental findings

A prototype system is implemented to verify the proposed method. The program is developed using visual Basic.NET programming language and Microsoft Word (word processor). Experimental results showed that Text Fingerprint algorithm has authenticated the integrity of electronic documents but where to hide Text Fingerprint was the issue that rose up whenever the text file size increased. After trying to hide Text Fingerprint in different parts of the file structure and inside text image contents (images, icons, etc.) if present. The storage location of Text Fingerprint became a concern to find a place without affecting text display or increasing file size. Barcode technology can handle this issue, actually barcode and human-readable text are often printed together. The cost of creating and processing the barcode symbol is negligible due to the availability of bar code applications in websites and mobile phones with cameras. A 2D barcode can verify the integrity of an offline document by encoding secret information such as Text Fingerprint inside the barcode symbol. Therefore, the solution that we propose is the use of 2D barcode as a cover envolpe for the Text Fingerprint.

5. 2d barcode for solution enhancement

A 2D barcode (two-dimensional) is the result of ongoing research for storing more data in barcodes to support information distribution and detection without accessing the database. The conventional barcode (one-dimensional) has low information density and function as a key to databases, while 2D barcode structure is made of horizontal and vertical axis of the data array. Consequently, a higher information density in 2D barcode can be used to encode explicit information rather than a database key. Furthermore, a variety of languages and graphical information can be expressed in this higher information density. There are four widely known 2D barcodes that are ISO standard, displayed in Figure 4 and will be briefly introduced;

1-Maxi Code: is a fixed length matrix-type code developed in 1989 to enhance the internal logistics management such as automated package sorting and efficient customer services by US courier company, United Parcel Service (UPS). Maxi Code is an open system, which means it is available to the general public for use and/or modification from its original design. Maxi Code can carry any of 256 ASCII characters and hold up maximum 93 characters or 135 numeric characters per symbol. The symbol is composed of a central bulls-eye locator and offset rows of hexagonal elements. The dimensions of the symbol are approximately 1.11 x 1.054 inches.

2-Data Matrix: is a 2D barcode which use dot matrix printer to print black and white elements representing 1 and 0, respectively, forms data patterns. The symbol is square and can range from 0.001 inch per side up to 14 inches per side. Data Matrix can carry any of 256 ASCII characters and density storage depends on the symbol size. For example, 500 numeric characters can be encoded in a 1-inch square or 500 ASCII characters can be encoded in a 1.4-inch square.

3-PDF417: is a 2D barcode which is composed of the combination of four bars and four spaces and the shape of the symbol, rectangular, can be adjusted by setting the width and allowing the height to grow with the data. PDF417 can store up to 1,800 ASCII characters or 1,100 binary characters per symbol. PDF-417 symbol can encode large amounts of data into several PDF-417 symbols which are logically linked.

4-QR Code (Quick Response Code): is a 2D barcode constructed of black square pattern on white background and three main squares in the bottom left, top left, and top right corners are locator patterns. QR Code was developed in Japan by the Nippondenso Company. The data density storage is determined by QR code version number (1 to 40). This indicates the number of rows and columns the QR Code can use. It can grow in increments of 4 cells per side from 21x21 cells in version 1 to 177x177 cells in version 40. The symbol can encode up to 7,089 numeric characters, 4,296 alphanumeric characters, or 2,953 bytes.

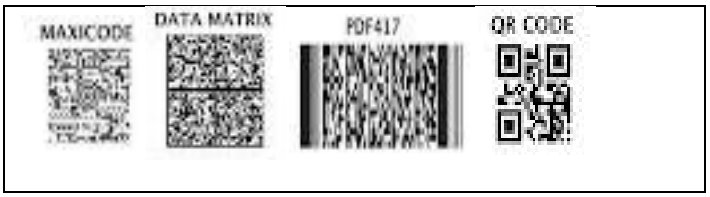

Figure 4. 2D Barcodes :Maxi Code, Data Matrix, PDF417, QR Code

The proposed solution is to encode the Text Fingerprint matrix inside QR Code for the following reasons.

- QR Code is a 2D barcode with 3k bytes data storing which is more than Text Fingerprint matrix size. For example, the Text Fingerprint matrix generated from 100K bytes text document (almost 60,000 characters) is less than 2k bytes.
- QR Code employs a Reed-Solomon error correction algorithm to detect and correct data errors due to a dirtied or damage area. This will improve the process of an offline document integrity and authenticity verification during checking the embedded Text Fingerprint in QR Code with the document contents.
- QR Code can be printed in an area less than 10 mm square.

6. Conclusion

The paper has illustrated how text could have its own fingerprint to identify and protect itself. The algorithm used could be developed and enhanced to protect a text document against copying, plagiarism, piracy and illegal distribution. The algorithm used could be embedded inside QR Code printed inside the text files to generate a relationship matrix to "watermark" the text. This watermark approach will protect the content of the text document if tampered with and encourage usage of electronic versions of printed literature, while protecting copyrights. Although, the detection algorithm discussed in the paper can be advanced further to facilitate superior security levels, it illustrates a more structured form of obtaining a digital fingerprint. QR Code the security envelope of Text Fingerprint provides a promising solution to current digital security challenges, and provides a starting step to secured sharing, usage and application of digital content.

7. References

[1] "Hiding Information in Document Images," J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, Proceedings of the 29th Annual Conference on Information Sciences and Systems, Johns Hopkins University, March 1995, pp 482

[2] J. T. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman,"Electronic Marking and Identification Techniques to Discourage Document Copying," IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, October 1995, pp. 1495.

[3] 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.

[4] Wang Yun-Cai ; Niu Ju-Fen, "Research on digital content copyright protection system", 2009 IEEE International Conference on Network Infrastructure and Digital Content, November 2009, pp-1050.

[5] Jinjie Gu , Yuzhu Cheng, "Research on security protection mechanisms of digital content", 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), July 2010, pp-155

[6] Fred Piper, Sean Murphy, "Cryptography". Oxford University Press Inc., 2002, pp 75-106.

[7] Qadir, M.A. , Ahmad, I. "Digital text watermarking: secure content delivery and data hiding in digital documents" 39th Annual 2005 International Carnahan Conference on Security Technology, 2005. CCST '05, February 2006, pp. 101.

[8] H.Singh, P. K.Singh, K.Saroha, "A Survey on Text Based Steganography", Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development, February 2009

[9] Ganeshkumar, V. , Koggalage, R.L.W, "A language independent algorithm to send secret messages using steganography", 2010 International Conference on Advances in ICT for Emerging Regions (ICTer), October 2010, pp-15.

[10] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker. "Digital Watermarking and steganography", Morgan Kaufmann publishers, Seattle, Washington, USA, 2008.

[11] Z. Jalil and A. M. Mirza, "A Review of Digital Watermarking Techniques for Text Documents," 2009 International Conference on Information and Multimedia Technology (ICIMT 2009), IEEE Computer Society, December 2009, pp. 230, doi:10.1109/ICIMT.2009.11

[12] Y.Won Kim, K.Ae Moon, I.Seok Oh, "A text watermarking algorithm based on word classification and inter-word space statistics" 2003 International Conference on Document Analysis and Recognition, IEEE Computer Society, September 2003, pp. 775.

[13] Jalil, Z. ; Mirza, A.M. ; Jabeen, H, "Word length based zero-watermarking algorithm for tamper detection in text documents", 2010 2nd International Conference on Computer Engineering and Technology (ICCET), April 2010, pp-382.

[14] D. Huang and H. Yan, "Inter-word distance changes represented by sine waves for watermarking text images," IEEE Trans. Circuits and Systems for Video Technology, Vol.11, No.12, pp.1237, December 2001.

[15] Micic, A. ; Radenkovic, D. ; Nikolic, S. "Autentification of Text Documents Using Digital Watermarking " 7th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services,05. September. 2005, pp503.

[16] 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.

[17] Hassan M. Meral et al., "Natural language watermarking via morphosyntactic alterations", Computer Speech and Language, 23,107-125, 2009.

[18] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U.Topkara, and K. E. Triezenberg, "Natural Language Watermarking and Tamperproofing", Fifth Information Hiding Workshop, vol.LNCS, 2578, October, 2002, Noordwijkerhout, The Netherlands,Springer-Verlag.

[19] M. Shirali-Shahreza, "Text Steganography by Changing Words Spelling", In Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT 2008), Phoenix Park, Korea, 2008.

