Hindawi
Security and Communication Networks
Volume 2018, Article ID 5325040, 22 pages
https://doi.org/10.1155/2018/5325040

Review Article

A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents

Milad Taleby Ahvanooey,[1] **Qianmu Li**,[1] **Hiuk Jae Shim**,[2] **and Yanyan Huang**[3]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[2] *School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[3] *School of Automation, Nanjing University of Science and Technology, Nanjing, China*

Correspondence should be addressed to Milad Taleby Ahvanooey; taleby@njust.edu.cn and Qianmu Li; qianmu@njust.edu.cn

Received 6 October 2017; Revised 28 December 2017; Accepted 16 January 2018; Published 17 April 2018

Academic Editor: Pino Caballero-Gil

With the ceaseless usage of web and other online services, it has turned out that copying, sharing, and transmitting digital media over the Internet are amazingly simple. Since the text is one of the main available data sources and most widely used digital media on the Internet, the significant part of websites, books, articles, daily papers, and so on is just the plain text. Therefore, copyrights protection of plain texts is still a remaining issue that must be improved in order to provide proof of ownership and obtain the desired accuracy. During the last decade, digital watermarking and steganography techniques have been used as alternatives to prevent tampering, distortion, and media forgery and also to protect both copyright and authentication. This paper presents a comparative analysis of information hiding techniques, especially on those ones which are focused on modifying the structure and content of digital texts. Herein, various text watermarking and text steganography techniques characteristics are highlighted along with their applications. In addition, various types of attacks are described and their effects are analyzed in order to highlight the advantages and weaknesses of current techniques. Finally, some guidelines and directions are suggested for future works.

1. Introduction

Following the progressive growth of Internet and advancement of online services, digital publishing has become an essential topic and in the next-generation organizations, offices (e.g., institutions and publishers) seem to be paperless. Nowadays, various studies are in process to execute and organize some ideas such as e-commerce, e-government, and online libraries. Digital publishing has many privileges, but it has some fundamental threats such as illegal use of copyrighted documents, manipulating the data, and redistributing such information [1–3]. In this case, some protective solutions consisting of copyright protection, integrality, authenticity, and confidentiality are essential to prevent forgery and plagiarism problems. Downloading and manipulating a copyrighted text and thus reusing it without any control are easy these days; hence, copyright management is very necessary to protect such information against modifying and reproducing processes [4–6].

Digital text watermarking is a data hiding technique which conceals a signature or copyright information called watermark inside a cover text in an imperceptible way. Separating the hidden watermark from the cover text is very difficult due to the fact that watermarked text is invisible for everyone except the original owner. Recently, text watermarking has not drawn much attention from cyber security experts and researchers. There are some reasons for that: the much lower capacity of text to retain data might be one of the main reasons (i.e., compared to other digital media such as image, audio, and videos). However, there are a number of reasons why we should pay more attention to it. Firstly, the text is still a major form of universally applicable digital media. In other words, text is an important part of communication between people compared to other media. Secondly, text watermarking has no clear evaluation criteria to analyze its efficiency [6–10].

The different categories of information security systems are depicted in Figure 1. The cryptography and information

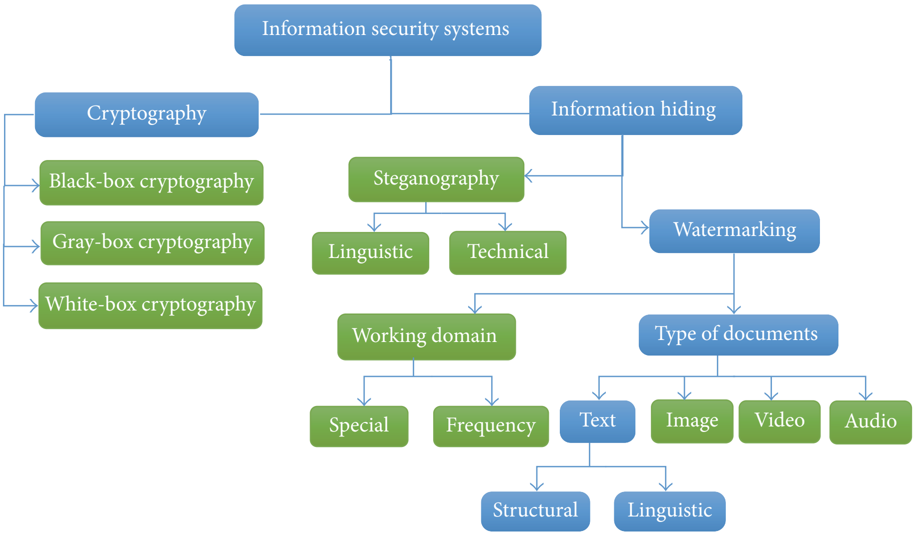


Figure 1: Different categories of information security systems.

hiding are security systems that are used to protect data from deceivers, crackers, hackers, and spies. Commonly, most of the malicious users want to leave traces from cuts, manipulations, and infections [7]. The cryptography scrambles a plain text into ciphertext which is reversible without data loss. The goal of cryptography is to prevent unauthorized access to the secret information by scrambling the content of information. On the other hand, information hiding is a powerful security technique which hides a secret data in a cover media (e.g., text, image, audio, or video) so that the trace of embedding hidden data is completely unnoticeable. The cryptography and information hiding are similar in a way that both are utilized to protect sensitive information. However, the imperceptibility is the difference between both techniques; that is, information hiding concerns how to hide information unnoticeably. Generally, the information hiding can be further categorized into steganography and watermarking. The aim of steganography is to hide a secret message in a cover media in order to transmit the secret information; therefore, the main concern is how to conceal the secret information without raising suspicion; that is, steganography needs to conceal the fact that the message is hidden. Watermarking is concerned with hiding a small data in digital files such that the hidden data is robust to alterations and adjustments. In other words, watermarking aims to protect intellectual property of digital media against unauthorized copy or access by embedding a watermark (visible or invisible) in the cove media which can remain beside the data, and it can be used whenever there is any query about the originality of media (e.g., the hidden watermark refers to the original owner) [2–10].

Over the last two decades, many information hiding techniques have been proposed in terms of text watermarking and text steganography for copyright protection [11–14], proof of ownership [15–23], and copy control and authentication [24–31]. Although the aim of steganography is different, it also can be used for the copyright protection of digital texts like watermarking.

The main contributions of this paper are summarized as follows:

(i) We present a brief overview of existing literature on text watermarking categories, architecture, applications, attacks, and evaluation criteria.

(ii) We summarize some information hiding techniques which are focused on altering the structure and content of the cover text in order to hide secret information.

(iii) We provide a comparative analysis of the summarized techniques and evaluate their efficiency based on the specified criteria.

The rest of the paper is organized as follows. In Section 2, we review text watermarking literature and related studies. In Section 3, we introduce individual text watermarking methods and analyze them based on evaluation criteria, and we give some suggestions for the future works in Section 4. Finally, Section 5 draws the conclusions.

2. Literature Review

In what follows, we describe the existing literature on text watermarking including architecture, the Unicode standard, text watermarking categories, applications, evaluation criteria, and attacks.

2.1. Text Watermarking Architecture. As shown in Figure 2, digital text watermarking includes two main phases, namely, watermark embedding and watermark extraction.

(i) Watermark Embedding. The embedding phase of text watermarking algorithm includes three stages. The first stage is generating a watermark string which includes the owner's name or other pieces of information (e.g., author and publisher). In the second stage, the watermark string is converted to a binary string, which is modified by a hash function

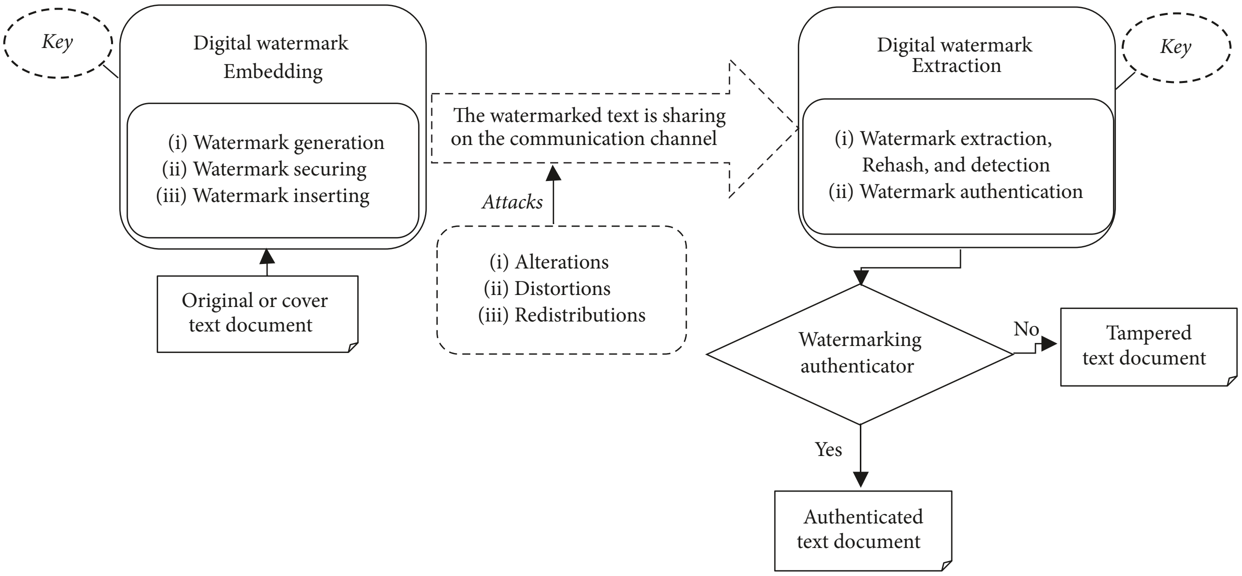

Key
Digital watermark Embedding
(i) Watermark generation
(ii) Watermark securing
(iii) Watermark inserting
Original or cover text document
The watermarked text is sharing on the communication channel
Attacks
(i) Alterations
(ii) Distortions
(iii) Redistributions
Digital watermark Extraction
Key
(i) Watermark extraction, Rehash, and detection
(ii) Watermark authentication
Watermarking authenticator
No
Tampered text document
Yes
Authenticated text document

Figure 2: Digital text watermarking (embedding and extraction) architecture.

according to an optional key, and then an invisible watermark string is generated for embedding it into special locations in the cover text. Finally, it is inserted into special locations where the watermark string will not be affected by attacks [2, 4, 6, 7].

(ii) Text Documents Attacks. Nowadays, most of the users can easily utilize various digital text files such as articles, books, and online news. Due to availability of open access to these text documents, unauthorized attacks such as copy, alterations, distortions, and redistributions are simultaneously raising. Therefore, text watermarking can be used as a security tool to prove the originality and the accuracy of text documents [6, 31].

(iii) Watermark Extraction. Generally, watermarked text documents are shared via communication channels such as web, email, or social media over the Internet. Obviously, it is essential to authenticate the originality of the text documents. Two different terms are used for this phase, that is, extraction and detection. Although authors often regarded both as similar functions in some literature, we can distinguish them in this way: whereas the extraction discovers the watermark string from the watermarked document and authenticates its integrity, the watermark detection verifies the existence of the watermark string from the watermarked text [4, 6, 7, 48].

2.2. Unicode Standard. In the digital text processing system, the Unicode standard to process and display digital texts from 1987 until now has been defined. Basically, all operating systems and writing software systems have to support the Unicode standard for representation of digital texts. The Unicode standard is a universal character encoding system designed to support the worldwide display, processing, and interchange of the texts with different languages and technical disciplines. In addition, it also supports the historical and classical letters in many languages. This standard is compatible with the latest version of ISO/IEC 10646-1:2017 and has the same characters and codes of ISO/IEC 10646. As of June 2017, the latest version of Unicode is 10.0.0 is maintained by the Unicode Consortium. It includes three encoding forms such as UTF-8, UTF-16, and UTF-32 which the Unicode allows for 17 planes, each of 65,536 possible characters (or "code points"). This gives a total of 1,114,112 possible characters in different formats such as digits, letters, symbols, and a huge number of current characters in various languages around the world. Currently, the most commonly used encoding forms are UTF-8, UTF-16, and now-outdated UCS-2. UTF-8 provides one byte for any ASCII character, all of which have the same code values in both ASCII and UTF-8 encoding, and up to four bytes for other characters. UCS-2 provides a 16-bit code unit (two 8-bit) for each character but cannot encode every character in the current Unicode standard. UTF-16 extends UCS-2, using one 16-bit unit for the characters which were representable in UCS-2 and two 16-bit or (4 × 8-bit) units to process each of the further characters [11, 28–32, 52, 53].

In the Unicode standard, there are special characters used to control special entities such as zero-width joiner, nonjoiner, and special spaces (or white spaces). Practically, they have no written symbol (i.e., nonprinting characters) in the digital text processing systems. In the social media, if it employs the Unicode standard in order to process digital texts in different languages, then the Unicode control characters have transparent written symbols; otherwise they may generate some unconventional symbols [48].

In some existing literature, the researchers have utilized the Unicode control characters in order to hide the secret data into a cover text, where they provide the imperceptible embedding or a few change in the cover [11–14, 19, 22, 33, 54].

As depicted in Table 1, all the Unicode special spaces have different width and no written symbol (color) in digital

Table 1: Unicode special space characters [22, 32].

Unicode Hex code	HTML code	Name	Written symbol
U+0020	&#32;	Space	“ ”
U+00A0	&#160;	No-break space	“ ”
U+200A	&#8202;	Hair space	“🟨”
U+2000	&#8192;	En quad	“🟩”
U+2002	&#8194;	En space	“🟪”
U+2003	&#8195;	Em space	“⬜”
U+2001	&#8193;	Em quad	“🟦”
U+2004	&#8196;	Three-per-em space	“🟥”
U+2005	&#8197;	Four-per-em space	“ ”
U+2006	&#8198;	Six-per-em space	“🟦”
U+2007	&#8199;	Figure space	“🟩”
U+2008	&#8200;	Punctuation space	“🟪”
U+2009	&#8201;	Thin space	“🟦”
U+202F	&#8239;	Narrow no-break space	“ ”
U+205F	&#8287;	Medium-mathematical space	“🟦”
U+3000	&#12288;	Ideographic space	“ ”

Table 2: Unicode zero-width control characters [11, 32, 33].

Unicode Hex code	HTML code	Name	Text written symbol
U+200B	&#x200b;	Zero-width space	No symbol and width
U+200C	&#x200c;	Zero-width nonjoiner	No symbol and width
U+200D	&#x200d;	Zero-width joiner	No symbol and width
U+200E	&#x200e;	Left-to-right mark	No symbol and width
U+202D	&#x202d;	Left-to-right override	No symbol and width
U+202E	&#x202e;	Right-to-left override	No symbol and width
U+202A	&#x202a;	Left-to-right embedding	No symbol and width
U+202B	&#x202b;	Right-to-left embedding	No symbol and width
U+202C	&#x202c;	Pop-directional formatting	No symbol and width
U+180E	&#x180e;	Mongolian-vowel separator	No symbol and width

text processing (i.e., we inserted these spaces between double quotation marks and changed color to show their width).

As shown in Table 2, the zero-width characters are totally invisible. We have tested all of these characters by Java programming in the Docx, txt, and HTML files, (i.e., some of the zero-width characters are blocked in G-mail, but they can be used in web watermarking). In practice, when the zero-width characters are used in order to hide a secret data in the cover text, the default encoding of the cover text must be defined as one of the Unicode encodings like UTF-8, UTF-16, or UTF-32. In case of attack, if a malicious user copies a text which included some zero-width characters in the new host file, then these characters will be considered as the Unicode encoding and provide invisible text trace. Otherwise, they will show some unsupported characters and raise suspicions to the existence of hidden information.

2.3. Text Watermarking Categories. During past two decades, many types of research have been carried out based on structural (format based), linguistic, scanned-image watermarking and frequency of words in the cover text. Herein, we consider those methods which are focused on modifying the structure and content of the cover text. In case of text processing, watermarking techniques are divided into two main categories, linguistic and structural. The linguistic technique concerns with the special features of the text content that can be changed in a specific language, and moreover, the structural technique concerns with the layout or format of the cover text that can be modified [6, 18], although some researchers have classified the text watermarking techniques based on the features of methods such as robust, fragile, invisible, and visible [10, 55].

(i) Linguistic (natural language) technique is divided into two types: syntactic and semantic. Syntactic text watermarking involves altering the structure of text without significantly changing the original meaning of cover text. Obviously, text documents consist of several sentences, words, verbs, nouns, prepositions, adverbs, adjectives, and so on. There are various

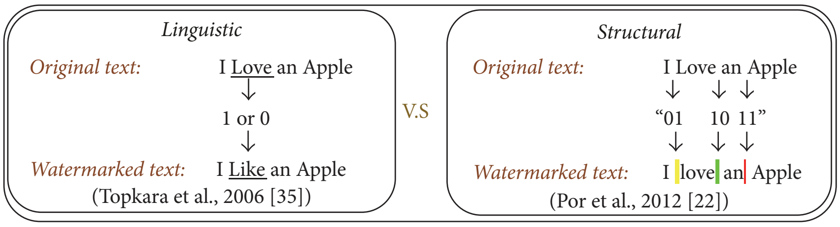


Figure 3: Comparison between linguistic and structural techniques.

syntactic compositions in sentences of text, which is determined by the language and its particular conventions [37]. Semantic text watermarking is a language-based technique which focuses on the semantic arrangement of cover text such as the spelling of words, synonyms, and acronyms in order to conceal a watermark string. The advantage of this technique is that it provides protection retyping attacks or using OCR programs; however, it alters the original meaning of text content [35].

(ii) Structural based (format based) technique involves altering the layout of text based on the Unicode or the ASCII encoding without changing the sentences or words. The structural approach alters word spacing, line spacing, font style, text color, and anything similar [44, 56].

Figure 3 depicts two examples of hiding the watermark bits into an example sentence by using the linguistic and the structural approaches.

As shown in Figure 3, the linguistic technique changes the text content and the structural technique alters the layout of text. In addition, in Section 3, we will explain two approaches in detail.

2.4. Text Watermarking Applications. Text watermarking techniques are applicable in many applications. The following points are the most important watermarking applications.

(i) Digital Copyright Protection (Proof of Ownership). Text watermarking provides passive protection tools for digital documents so that the text content cannot be illegally copied or replicated. For example, if someone copies a watermarked document/file (e.g., PDF, Docx, Latex, and RTF), then the reversibility of watermarking techniques can be used to prove the ownership of the copied documents [6, 8].

(ii) Access Control (Copy Control). Currently, the publishers and the content providers are seeking more reliable ways to control copy or access to their valuable documents, and simultaneously, they want to make the documents accessible on the Internet in order to obtain more revenue. The text watermarking is a desirable technique on the online systems that provide access control to prevent illegal copy or restrict the number of times of copying the original text [8, 57].

(iii) Tamper Proofing. These days, a huge number of text documents are available online for selling or reading for users. Therefore, these documents are prone to be exposed to a number of attacks (e.g., unauthorized access, copy, and redistribution). In this case, text watermarking can be used as a fragile tool for tamper proofing of the watermarked texts against attacks. In general, a fragile watermark is embedded into text documents, and if any type of alterations has been made, then it fails to detect the watermark [6, 18].

(iv) Text Content Authentication. The online publishing of articles and newspapers in form of plain text documents has brought several issues related to authenticating the integrity of these documents. Text watermarking can be applied as an authentication tool to verify the integrity of plain text documents [57].

(v) Forgery Detection (Prevention). Plagiarism and reproduction of text documents are serious forgery activities and are rapidly increasing. Text watermarking can be used as a forgery detection tool by embedding a watermark in the original text before the online publishing. Thus, it can prove the plagiarism and reproduction of the watermarked texts [6, 8].

2.5. Text Watermarking Evaluation Criteria. There are many things to be considered when the researchers design a watermarking algorithm. However, common criteria can be easily found in recently proposed algorithms: those are invisibility, robustness, embedding capacity, and security, which explain that an ideal watermarking algorithm should be secure and robust against attacks. In order to achieve high rates criteria, the researchers should consider the application of the method (e.g., fragile or robust). However, a suitable algorithm could be provided optimum trade-offs between the evaluation criteria according to the application requirements of method [4, 6–10].

In the following, we introduce five evaluation criteria that include some formula for analyzing the efficiency of watermarking algorithms.

(i) Invisibility (Imperceptibility). The trace of embedding a watermark in the cover text must be invisible and the watermark must be able to be extracted by the corresponding watermarking algorithm. In other words, invisibility refers to how much perceptual changes are made in the cover text after embedding a watermark. Practically, it cannot be measured numerically. The best way of measuring the degree of imperceptibility is to compare the difference between the original cover text and watermarked text [6, 11].

(ii) Embedding Capacity (EC). The number of watermark bits which can be concealed in a cover text is termed as

embedding capacity. This criterion can be measured numerically in units of bit-per-locations (BPL). Location means a specific position in the cover text where the algorithm can embed the watermark string (e.g., spaces between words and after special punctuations). Even though a watermarking algorithm provides a large embedding capacity, it is not desirable for copyright protection, if it alters the cover text profoundly [8, 21].

$$\mathrm{EC} = \mathrm{BPL} \times \text{Total Locations}. \tag{1}$$

(iii) Robustness. Many attacks may happen on the watermarked texts while they are shared on the communication channel and are referred to as hazard which could distort (or damage) the watermark [6]. In general, malicious users also may randomly manipulate or distort the embedded watermark in the watermarked text, rather than destroy or delete it. Moreover, any kind of distortion may occur deliberately or even unintentionally. A robust text watermarking algorithm makes it extremely hard to be altered or removed. The distortion robustness (DR) can be measured numerically by distortion probability [4, 5].

Distortion Probability (DP). This is the probability of how much proportion of watermark bits (WB) has been lost. The malicious users can manipulate or alter the watermarked text such that the WB may not be extracted. The lower rate of DP leads to greater robustness of watermarking algorithm. There is no specific formula for calculating the probability of distortion in the existing literature. We aim to provide a benchmark analysis of watermarking techniques which is dependent on the locations of embedding method in the cover text [12, 58].

Let us suppose that the number of embedding positions (e.g., space characters used to embed a watermark string) in the cover text is D, and the total number of characters in the cover text is considered as T, and S is the number of sample files, then the average probability of distortion robustness can be calculated as follows:

$$\mathrm{DR} = \frac{\left[\sum_{i=1}^{S}\left(1 - \mathrm{DP}_i\right)\right]}{S}, \tag{2}$$

where $1 < D < T$, $T \in N$, and $D \in N$.

$$\mathrm{DP}\left(\mathrm{WB}\right) = \frac{D}{T}. \tag{3}$$

(iv) Security. It prevents attackers from detecting the watermark visually or from deleting the watermark from the watermarked text by providing a certain level of security [7, 8]. In fact, this measure depends on three other criteria, including invisibility, embedding capacity, and robustness. The text watermarking algorithm should provide optimum trade-offs among these criteria. Moreover, if it provides a large capacity and the trace of embedding is totally imperceptible then the security of the algorithm is equal to the above robustness formula [6, 11].

(v) Computational Cost. This is one of the least significant criteria for the next-generation computers. However, there can be many pages in some text documents; therefore, the text watermarking approaches are preferred to be computationally less complex. It is obvious that the long documents require more software or hardware resources, that is, higher computational complexity. In general, less complex algorithms are exploited for resource-limited systems such as mobile devices and embedded microprocessors [6, 8].

2.6. Text Watermarking Attacks. Currently, the availability of open access and online publishing of valuable documents (e.g., books, articles, and newspapers) has caused to be exposed to new breeds of plagiarism and forgery attacks. Therefore, malicious users can access the plain text or even protected documents by unauthorized software tools. The attacks on watermarked text documents can be divided into three three categories: tampering attacks, estimation based attacks, and reformatting attacks [4, 6, 31, 58–61].

(I) Tampering Attacks. This kind of attack includes three types of attacks: removal, insertion, and reordering attacks.

(a) Removal (Deletion). In this attack, the malicious user attempts to delete the watermark string completely from the watermarked text without affecting the original text content. Moreover, if it cannot remove the watermark string completely, then it almost destroys it [6, 60].

(b) Insertion or Distortion. After copy, attacker's aim is to alter the original text by random removal of some words and manipulate the copied text. Sometimes, malicious users try to remove the authors' names or related information from the original text. Afterward, they insert new information in the copied text in order to show their ownership. In some literature, this type of attacks is called geometric attacks [58].

(c) Reordering (Rebuilding). Another way of tampering attacks is that the malicious users change the order of words and sentences to produce a new version of document by paraphrasing its content. Thus it may lose the watermark string and fails to detect or extract it [6, 59].

(II) Estimation Based Attacks. In this kind of attacks, the attackers must know some preliminary knowledge about text watermarking and the characteristics of text processing. Estimation based attacks include removal attacks, ambiguity attacks and copy attacks [23, 61].

(a) Estimate of the Original Text. Since, the watermark string is an extra independent data in the watermarked text, attackers may design an extraction algorithm to produce a new document without a watermark string. Thus, they try to estimate the relation between the watermark string and original text and in addition write an algorithm to extract the original text without the watermark such that it does not change the original text content [6, 31].

(b) Ambiguity (Reverse). This attack aims to puzzle the detector by estimating a forged watermark from the watermarked text. Therefore, it causes ambiguity in the ownership of

the watermarked text. If the watermark is approximately estimated, then the attacker can remove the watermark from the watermarked text [61].

(c) Copy Attacks. In this attack, the aim of attacker is to estimate a watermark and extract/perform it on the target watermarked text by claiming the ownership of the copied text. As the definition implies, it needs to perform a removal attack by extracting the estimated watermark (e.g., using previous statistical knowledge of watermark locations) to extract the watermark [8, 61].

(III) Reformatting Attacks. Most of the malicious users copy the target texts from websites or articles into their own files and may try to change the font style, font color, and so on. Some of these modifications that modify the layout of text without changing its content are called reformatting attack [7].

(a) Retyping Attacks. Sometimes, the content providers protect their text documents so that the text content is read-only and no one can copy even a small part of the text content. In this case, malicious users used to retype the target text [6].

(b) Copy and Paste Attack. This is one of the most common attacks in that the malicious users copy the whole of text and paste into their own files (e.g., a simple copy of the watermarked text into another file).

3. Text Watermarking Existing Techniques

Text watermarking techniques have various strategies and schemes which are dependent on the applications of the methods. In other words, the aim of watermarking determines whether the algorithm should be a fragile or robust tool; thus, it can be used to prove the integrity or originality of the text accordingly. Practically, developing a robust algorithm is not easy and requires considering the balance among multiple criteria that must be taken into account. Currently, there are a few text watermarking techniques which have been introduced; hence it is hard to find literature addressing its limitations. In this section, we introduce some related works which are focused on altering the structure and content of text in order to embed the hidden information [6, 7]. From the text processing point of view, text watermarking algorithms can be classified into one of the categories in Figure 4, namely, linguistic (natural language) and structural (format based).

3.1. Linguistic (Natural Language). This type of watermarking techniques modifies the content of a text document to hide a watermark binary string. In recent years, a few natural languages based algorithms have been introduced. As explained in Section 2.3, the semantic or syntactic analysis of the text contents is used to embed the watermark bits in the natural language (NL) watermarking. It generally changes the structures of text including nouns, adjectives, verbs, prepositions, pronouns, idioms, synonyms, and any available objects to conceal the watermark. Moreover, this type of techniques is designed to maintain the original meaning of the cover text by change the semantic or the syntax of its content [7, 27].

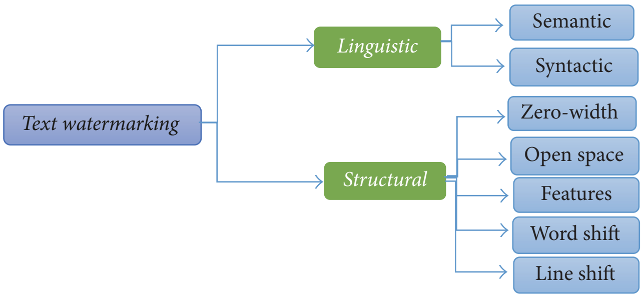


FIGURE 4: Different types of text hiding techniques.

Topkara et al. (2006) provided a new syntactic watermarking technique based on the syntax of the cover text especially in English language, which performs syntactic sentence-paraphrasing. In this work, the original sentence is analyzed by XTAG parser and then send for feature verification. Finally, the embedding algorithm inserts the watermark bits into the sentences by paraphrasing their contents [34].

Topkara et al. (2006) presented another semantic watermarking method by embedding synonym substitutions in the English language text documents. This method utilizes heuristic measures of quality based on conformity to a language model. While there are many ways to produce a substitution on a word, the algorithm prioritizes the means according to a quantitative resilience measure and uses them from the priority list. In this research, the authors attempted to increase the capacity and reduce the distortions against attacks in the watermarked text [35].

Meral et al. (2009) proposed a morphosyntax-based NL technique which embeds a watermark (binary string) based on a syntax tree in the Turkish text documents. The algorithm embeds watermark bits under the control of WordNet (or dictionary) to prevent semantic drops (e.g., altering the meaning of the original text). In this technique, the watermark bits are embedded by altering the changeable sentences in the cover text. These alterations include conjunct order change and adverb displacement. The direction of words (forward or backward) indicates the watermark bit either “1” or “0.” However, the capacity of this technique is low, and its achievable capacity is almost one bit per sentence [36, 51].

Kim (2008) suggested a syntactic text watermarking for the Korean language text documents. This work consisted of four stages. First, it creates a syntactic dependency tree of the cover text. Second, it selects target syntactic constituents to move words. Third, the algorithm embeds watermark bits (if the current bit does not match with the movement bit of the target constituent, the method moves the syntactic constituent in the syntactic tree). Finally, the algorithm produces the watermarked text from the modified syntactic

Table 3: Implementation of NL techniques on the highlight examples.

Algorithm	Original cover text	Watermarked text
Topkara et al., 2006 [34]	I love an apple	My favorite fruit is apple
Topkara et al., 2006 [35]	I love an apple	I like an apple
Meral et al., 2009 [36]	I love an apple Bir elmayı seviyorum	I like an apple Bir elmayı severim
Kim, 2008 [37]	I like apples in autumn 나는 가을철에 사과를 좋아한다	(in autumn) (I) (apples) (like) 가을철에 나는 사과를 좋아한다
Kim et al., 2010 [38]	(the departure) (was delayed) 출항이 지연되었다	(the departure) (was delayed) 출항이 지연이 되었다
Halvani et al., 2013 [39]	I love an apple Ich liebe einen Apfel	I like an apple Ich mag einen Apfel
Mali et al., 2013 [40]	You could go to school	You should go to school
He et al., 2009 [41]	Tom's leg is injured by falling 汤姆的腿[被]摔伤了。	Tom fell down and his leg is injured 汤姆[被]摔了, 他的腿摔伤了。

tree. The disadvantage of this method is that it is only to agglutinative languages such as Korean and Turkish; moreover, text reordering may change the meaning of the original text [37].

Kim et al. (2010) proposed another NL watermarking algorithm based on syntactic displacement and morphological division in the Korean language text documents. The authors utilized displacing syntactic adverbials attribute that most languages allow displacement of syntactic adverbials within its part. Moreover, they claim that proposed method does not change the general meaning of the sentences, but practically it alters the meaning of text slightly [38].

Halvani et al. (2013) proposed four methods to hide the watermark bits either by lexical or syntactic alteration, and those are designed for the German language. The first syntactic transformation applies enumeration modulation (EM) to embed the watermark bits using the grammatical rules "constituent movements." The second method uses conjunctions modulation (CM) method which is based on grammar rule (constituent movement) and focuses only on two nouns separated by an optimal conjunction. The third method is based on prefix expansion which modifies the negations of the words. The fourth method utilizes a lexical transformation to insert the watermark bits by altering words. The alteration is based on three grammatical rules such as repeated letters, connected anglicisms, and inflected adjectives. The advantage of these methods is compatibility to some other languages such as Spanish, English, or French. However, these methods also have the same problems as other NL algorithms: those methods almost change the original meaning [39].

Mali et al. (2013) introduced a novel NL watermarking method. In this approach, English grammatical rules are applied to produce watermark bits. The algorithm generates watermark bits based on a combination of the total conjunctions, pronouns, modal verbs, and author ID found in the cover text. Then, the watermark bits are encrypted with AES. This algorithm was designed for web pages' text verification. In addition, a receiver can authenticate the watermark by the extraction algorithm. Since this method modifies grammatical rules, it also changes the original meaning of sentences [40].

Lu et al. (2009) introduced a new watermarking technique, which embeds the watermark bits into the pragmatics properties of cover text by rewriting sentences. The method avoids syntactic and semantic analysis of text content and utilizes a transformation templates based on special pragmatic rules by part-of-speech (PaS) tags order in the Chinese language. It classifies sentences into subsets for embedding the watermark bits. For example, if the current subset is even, then the sentence represents a bit "1"; otherwise, the current subset is odd, then sentence indicates a bit "0." The authors aimed to paraphrase sentences without altering the original meaning of the Chinese text. However, this work is relatively weak against tampering attacks [41].

Practically, the NL watermarking is complicated since not every language supports syntactic or semantic alterations. Moreover, most of algorithms cannot be applied to sensitive documents because the original meaning or even word choice of text can be altered to some extent.

In this study, we analyze the efficiency of summarized techniques in terms of evaluation criteria that are explained in Section 2. In addition, we considered a rating factor for the NL techniques in terms of their criteria: for example, low, medium, and high scale for the capacity; low, modest, and high for the robustness; imperceptible, middle, and visible for the invisibility. The rate of each technique is estimated based on its embedding method. Language compatibility refers to the specific language to which the corresponding method can be applied.

To demonstrate the hiding process of above techniques, we implemented them on highlight examples (e.g., this process only embeds one bit in the cover text), which are depicted in Table 3. Moreover, the comparative analysis of the evaluated techniques in terms of criteria is summarized in Table 4.

As shown in Table 4, all the NL based techniques modify the cover text contents to hide the watermark bits. Thus, this

Table 4: A comparative analysis of NL techniques.

Algorithm	Embedding capacity (1 bit per)	Invisibility	DR	Language compatibility
Topkara et al., 2006 [34]	Sentences	Imperceptible	Low	English
Topkara et al., 2006 [35]	Synonym of words	Imperceptible	Low	English
Meral et al., 2009 [36]	Synonym of words	Imperceptible	Low	Turkish
Kim, 2008 [37]	Synonym of words	Imperceptible	Low	Korean
Kim et al., 2010 [38]	Displacement of adverbs	Imperceptible	Low	Korean
Halvani et al., 2013 [39]	Synonym of words	Imperceptible	Low	German
Mali and et al., 2013 [40]	Grammatical words	Imperceptible	Low	English
He et al., 2009 [41]	Sentences	Imperceptible	Low	Chinese

type of alteration needs some rules or locations to search target words for paraphrasing the text. In each technique, the authors use predefined dictionaries or "WordNet" to find and replace the target words. It causes high computational cost due to requiring an extra dictionary. Moreover, the NL based techniques are mostly incompatible with different languages. However, they perfectly protect the watermarked texts against retyping attacks but have low robustness against tampering attacks.

3.2. Structural (Format Based). This type of text watermarking alters the structural layouts or properties of the text in order to hide the watermark bites. As we already explained in Section 2.3, the structural layouts consist of spaces in between paragraphs, lines, words, curved letters, letter extensions, and characters with diacritical marks. Any other property can be utilized to change the layout or the format of cover text in an unnoticeable way. Recently, various techniques have been introduced by researchers which are employed by the modification of the text layouts to carry the embedded watermark bits.

Bender et al. (1996) presented the first open space data hiding technique which uses white space in text documents. White space based embedding algorithm considers three different locations: interword spaces, intersentence spaces, and end-of-line spaces. In case of interword spacing, the algorithm inserts additional spaces between two words; for example, two spaces between words represent a bit value of "1" in the watermark bits, while a single space represents a bit value of "0." For intersentence spacing, the "0" can be represented by inserting one space between sentences, and the "1" by inserting double spaces. In case of end-of-line spaces, two spaces are inserted to represent one bit (per line), four spaces represent two bits, (e.g., six spaces three bites), and so on. This technique is completely applicable for different languages; however, the disadvantage of the method is low capacity since only one or two bits per location can be embedded. Moreover, this method is not able to preserve the embedded bits against tampering and retyping attacks [44].

Brassil et al. (1999) proposed another watermarking technique based on modifying the appearance of different elements of cover text. The method considers three different structural layouts such as line-shift, word-shift, and text formatting. In this work, the embedding algorithm inserts watermark bits by shift (line or word) such that it moves a word (line) to downward (left or right) or upward (top or down) and changes the height of corresponding character. The extraction algorithm analyzes the lines or words of the watermarked text (scanned image) to detect the orientation of movements. Even though reformatting of the digital watermarked text causes the watermark bits to be lost, it provides a new perspective for structure-based text watermarking techniques.

Lee and Tsai (2008) introduced a data hiding approach for secure communication through web text documents. The algorithm embeds a secret message between words using special spaces. In this study, the method converts a secret message to a binary string based on ASCII codes. Therefore, it embeds the binary string into the web document by replacing nine special spaces between words according to a 3-bit group coding, of which the spaces are listed in Table 5.

The authors provided a secret communication on web pages by using undefined characters in Unicode or ASCII (e.g., "&#x32" and "&#160"), which makes an unpleasant text in the output text. For example, let "Apr. 21, 2017" be as the original text and "&#x32" be the special space for hiding "010" bits, then the watermarked text might be "Apr. &#x3221, 2017" and also a browser will show us "Apr.㈡, 2017" (i.e., we tested the sample by the HTML language). Lee and Tsai presented a new way of data hiding scheme to replace the between-word locations by using different coding; however, the algorithm was performed on the Internet Explorer version 6; hence the result of output message was invisible. During our test, we utilized the latest version of common web browsers such as Chrome, and Firefox; therefore, this algorithm showed us unpleasant characters between words in the output text after embedding [42].

Cheng et al. (2010) proposed a robust watermarking algorithm. The method utilizes the color feature of cover text in order to embed the watermark bits based on the watermark fragments and regrouping strategy. This technique has advantages such as improved robustness, capacity, and invisibility; however, it is vulnerable to highlighting words in the pdf or the MS word files since highlighted words on the watermarked text will definitely change the text color; consequently, it does not provide robustness against reformatting attacks [45].

Table 5: Special spaces based 3-bits group coding [42].

Number	Name	Reference type	Code type	Code inserted in HTML	Bits encode
(1)	(Normal) space	Normal space	ASCII	Typed space (with 20 h inserted)	000
(2)	(Normal) space	Numeric character reference	Unicode	&#x20;	001
(3)	(Normal) space	Numeric character reference	Unicode	&#x32;	010
(4)	(Normal) space	Numeric character reference	Unicode	&#x32	011
(5)	Nonbreak space	Numeric character reference	Unicode	&#xA0;	100
(6)	Nonbreak space	Numeric character reference	Unicode	&#160;	101
(7)	Nonbreak space	Numeric character reference	Unicode	&#160	110
(8)	Nonbreak space	Character entity reference	HTML	&nbsp;	111
(9)	Nonbreak space	Character entity reference	HTML	&nbsp	Unused

Gutub et al. (2007) suggested a new text watermarking method by using Kashida or extension feature of the Arabic language. A Kashida or extension character is utilized to adjust text by changing word length; however, it does not change the meaning at all (e.g., ســــلام = سـلام). A Kashida is added before or after the characters containing points (pointed characters) to hide a bit "1" and is added before or after the characters without points (unpointed characters) to hide a bit "0" [49]. Gutub et al. (2010) also introduced a new Kashida based watermarking method which employs a special pattern for embedding the watermark bits. It improves embedding capacity by adding one Kashida after any compatible character, which represents a bit "0" and double Kashidas for a bit "1." This algorithm is designed to provide proof of ownership and authentication for web text documents [47].

Alginahi et al. (2013) presented a new Kashida based watermarking approach for hiding a secret data through the Arabic text. In this work, one Kashida represents a bit "1" and "0" by omitting it before specific characters (ء، ا، أ، إ، آ، ئ، د، ذ، ر، ز، و، ؤ) [20]. Later, Alginahi et al. (2014) utilized two set of characters according to their frequency of repetition in the Arabic digital texts. In this study, the authors proposed two methods (A and B) in order to embed the watermark bits by adding one Kashida for a bit "1" and omitting a Kashida for a bit "0" in special locations into the text. The embedding locations are after 14 characters with high repetition (ا، ل، ي، و، م، ن، ت، ر، ب، ع، ه، د، س، ک) for method A and after 15 characters with low repetition (غ، ز، ث، ش، ظ، ط، ض، خ، ذ، ص، ج، ح، ق، ف، ة) for method B [21].

Al-Nofaei et al. (2016) proposed a Kashida based steganography technique for Arabic digital texts. This method improves the feature of hiding data within the Kashida character in the Arabic text documents using whitespaces between words. In practice, this method provides high imperceptibility and better capacity compare to other techniques; however, its robustness is low against tampering and retyping attacks [50].

However, all the Kashida based hiding techniques [20, 21, 47, 49, 50, 62–66] provide high imperceptibility and optimum capacity and are able to apply to text documents in the Arabic, Persian, and Urdu languages, but they cannot retain the watermark bits against tampering and retyping attacks.

Chou et al. (2012) suggested a reversible data hiding scheme for HTML files based on adding specific space characters between words in the cover text. In this method, English sentences are divided into several textural segments and every textural segment includes some blank characters (between-word locations). Then, Cartesian production strategy is utilized to create the pairs of spaces, and the blank characters are replaced by the new pairs of spaces (according to watermark bits) [46]. This method improves the embedding capacity (i.e., five bits per location) of the method introduced in [42]. However, it generates large gaps between words and is vulnerable to tampering and retyping attacks.

Por et al. (2012) introduced a data hiding method called UniSpaCh, which employs the Unicode special spaces in order to hide secret information into the Microsoft word files. The method utilizes specific locations such as interword, intersentence, end-of-line, and end of paragraph spaces to conceal a secret message [22]. In addition, a combination of double spaces is utilized for embedding the secret bits as depicted in Table 6.

The merit of this method is that a combination of spaces provides more embedding capacity than the previous methods. However, this method also cannot avoid generating unpleasant gaps in the watermarked text and is vulnerable to tampering and retyping attacks [22].

Mir (2014) proposed another open space based text watermarking method by using the structural properties of HTML language. The algorithm utilizes a hash function to generate watermark bits and, in addition, the hashed watermark bits are converted to an invisible string by replacing three special space characters (i.e., u202F, and u205F, and u200A), then it embeds the invisible string in the <meta> tag of a HTML file. The authors claimed that it can be applied in multilingual text files and provides high robustness. However, there are disadvantages: those characters generate unpleasantly large

Table 6: Binary classification model in UniSpaCh [22].

Symbol	Spaces	Sequence
	Representation scheme for interword spacing and intersentence spacing (Group A)	
“ ”	Normal	00
“🟨”	Thin + normal	01
“🟦”	Six-per-em + normal	10
“🟥”	Hair + normal	11
	Representation scheme for end-of-line and interparagraph spacing (Group B)	
“🟥”	Hair	00
“🟦”	Six-per-em	01
“🟪”	Punctuation	10
“🟥”	Thin	11

Table 7: Binary classification model in [15].

Character name	Two-bit classification	The character written symbol	Character Unicode code
Zero-width space	00	No symbol and width	U+200B
Zero-width joiner	01	No symbol and width	U+200C
Zero-width nonjoiner	10	No symbol and width	U+200D
Mongolian-vowel separator	11	No symbol and width	U+180
LRE or RLE	End of the watermark message	No symbol and width	U+200A or U+200B

gaps between words, and it is also vulnerable to tampering, copy and paste, and retyping attacks [67].

Taleby Ahvanooey and Tabasi (2014) introduced an invisible watermarking technique by adding hidden Unicode characters in Microsoft word files. In this method, a watermark string is firstly converted to 8-bit ASCII code. Each two bits' pair of the binary watermark sequence is represented by the zero-width characters as shown in Table 7. Then, the zero-width characters are embedded after special punctuation characters (e.g., dot (.), comma (,), and semicolon (;)). In this work, the researchers aim to protect the watermark against tampering attacks by hiding many times of watermark string into the original cover text. In order to point to the number of watermark strings, it inserts a zero-width character (LRE or RLE) after embedding each watermark string according to the language of text (English or Persian).

Moreover, the extraction algorithm verifies the length of the watermark bits with the extracted watermark bits and the location of LRE or RLE to check the accuracy of watermark detection. This work was designed to authenticate e-text and prove the ownership of Microsoft word files. Even though the method provides a high degree of invisibility, optimum robustness, and low capacity, it has the same disadvantage with other structural methods which is vulnerable to retyping attacks [15]. Later, Taleby Ahvanooey et al. (2015) proposed a method which improves the embedding capacity (16 bits per location) over the previous work by selecting different locations of embedding in to the cover text, for example, after special punctuation characters, between blank lines and paragraphs [16].

Alotaibi and Elrefaei (2016) designed a new watermarking technique to conceal secret information in Arabic text documents. This method groups characters according to the dotting feature of the Arabic alphabets as depicted in Table 8 [43].

Table 8: Two grouping of Arabic letters [43].

Pointed letters	Unpointed letters
ش ز ذ خ ج ت ث ب ي ن ق ف غ ظ ض ة	ص س ر د ح ا و ه م ل ک ع ط

In this study, the authors utilized a pseudospace to mark the watermark bits according to pointed and unpointed letters. The pseudospace (or ZWNJ: zero-width nonjoiner, “U+200C”) is a zero-width character which separates joined letters in the Persian/Arabic and does not have written symbol and width. If it is located between two joinable letters, then they will be separated (e.g., می‌خواهم = میخواهم). In order to hide the watermark bits, this algorithm inserts a combination of the space between words and the letter before it. If the watermark bit is zero and the letter is unpointed (for simplicity, {0, unpointed}), then the pseudospace is embedded. If {1, unpointed}, no pseudospace is embedded. In case of {1, pointed}, the pseudospace is embedded. If {0, pointed}, no pseudospace is embedded. This method provides high invisibility; however, it has relatively low capacity (one bit per pseudospace) and also is vulnerable to tampering and retyping attacks [43].

Another invisible watermarking approach was suggested by Taleby Ahvanooey et al. (2016), which belongs to the structural watermarking category. In this technique, the watermark message (web page's URL) is converted to a binary

Table 9: Unicode zero-width control character symbols in [11].

HTML Hex code	Unicode Hex code	Unicode char name
Right-to-left override	U+202E	&#x202e;
Left-to-right override	U+202D	&#x202d;
Pop directional Formatting	U+202C	&#x202c;
Right-to-left override	U+202E	&#x202e;

Table 10: Unicode groups pattern binary in [11].

Zero-width group embedding HTML code	Three-bit classification
&#x202e; &#x202e; &#x202e;	000
&#x202d; &#x202d; &#x202e;	001
&#x202e; &#x202d; &#x202e;	010
&#x202d; &#x202e; &#x202e;	011
&#x202c; &#x202e; &#x202e;	100
&#x202c; &#x202e; &#x202c;	101
&#x202c; &#x202d; &#x202d;	110
&#x202c; &#x202d; &#x202e;	111

Table 11: Space characters used in [19].

Character name	Hex code	Space text-face
Pseudospace (ZWNJ)	U+200C	No width and no face
Thin space	U+2009	“	”
Hair space	U+200A	“	”
Zero-width space	U+200B	No width and no face

string and the string is further encoded by a hash function. Then the hashed bits are embedded by invisible zero-width characters as shown in Tables 9 and 10. This method hides a watermark string at the end of each sentence which can be used as a tool to prove the ownership of web text documents [11].

This technique was designed to protect web pages against forgery and plagiarism attack that provides high invisibility, high capacity, and optimum robustness. Moreover, it is applicable to multilingual text documents. Since the algorithm embeds the invisible watermark one time after each dot character (.) or at the end of each sentence in the cover text, it is robust to tampering attacks, but it still is vulnerable against retyping attacks.

Alotaibi and Elrefaei (2016) proposed two open space based watermarking algorithms in Arabic texts [19]. In the first method, the dotting feature presented in [43] is utilized to improve the capacity of the previous work. In order to mark the watermark bits, the pseudospace (ZWNJ) is employed to embed before and after normal space depending on the character which can be pointed or unpointed. In the second method, as shown in Table 11, the four space characters are used to embed beside normal space.

In addition, each 4 bits from the watermark bits (or binary sequence) are embedded by corresponding space characters and order: the 1st bit is represented by pseudospace, the 2nd bit by thin space, the 3rd bit by hair space, and the 4th bit by zero-width space. Therefore, the existence of any four space character means a bit “1,” otherwise a bit “0.” For example, if only zero-width space is found between words, then it represents “0001.” The second method can be applied to multilingual text documents due to the space letter which is one of the writing structures. This method suffers from low robustness due to embedding four spaces beside each normal space in the cover text. For example, if an attacker alters or deletes a part of the text (include normal spaces) then it causes to fail the whole watermark string by extraction algorithm because of the normal space without other spaces referring to four bits “0000” in the watermark bits. Moreover, the authors claimed that their methods have high imperceptibility but they used two spaces with the deferent length which makes more gaps between words in the watermarked text [19].

Rizzo et al. (2016) presented a text watermarking technique which is able to embed a password based watermark in the Latin-based texts. This technique blends the original text and a user password through a hash function in order to compute the watermark. Then, it employs the homoglyph Unicode characters and special spaces in order to embed the watermark bits in the cover text. The authors claimed that this technique can hide a watermark (64 bit) into a short text with only 46 characters and, moreover, it provides high imperceptibility and high capacity. However, it is vulnerable against reformatting (e.g., changing the font type of watermarked text causes the watermark bits to be lost), tampering, and retyping attacks [29]. Due to utilizing the homoglyph Unicode characters, this method has low robustness against all the conventional attacks. Later on, Rizzo et al. [48] utilized the same method [29] to embed a watermark string in social media platforms.

Currently, the structural watermarking category is not greatly preferred since the watermarked documents are not robust enough against conventional attacks such as insertion, removal, reformatting, and reordering. In addition, sometimes even a simple text converting (e.g., webpage to doc file) causes the watermark detection by structural techniques to fail. However, it is obvious that the structural techniques provide imperceptibility and higher embedding capacity.

To demonstrate the hiding process of structural techniques, we implemented them on highlight examples that are depicted in Table 12. Herein, the implementation means evaluation of selected techniques based on their embedding methods.

Obviously, almost all the structural techniques provide high imperceptibility and better embedding capacity compare to the NL techniques.

To have a fair comparison between structural techniques, we considered those techniques which are able to apply to multilingual text documents. The Kashida based techniques are excepted due to focusing on the specific feature of the Arabic language which can be applied only in Arabic, Persian, and Urdu texts.

Table 12: Implementation of structural techniques on highlight examples.

Algorithm	Original text	Watermarked text	Embedded bits
Bender et al., 1996 [44]	Tom's leg is injured by falling.	Tom's leg is injured by falling.	5
Lee and Tsai, 2008 [42]	Tom's leg is injured by falling.	Tom's leg is &#x32injured by &#160falling.	15
Cheng et al., 2010 [45]	我喜欢一个苹果 I like an apple.	我喜欢一个苹果 I like an apple.	12
Chou et al., 2012 [46]	Tom's leg is injured by falling.	Tom's leg is injured by falling.	25
Gutub et al., 2010 [47]	احرارا, امهاتهم , ولدتهم , وقد , الناس, استعبدتم, متى	احرارا, امـهاتهم , ولـدتـهم , وقـد , الناس, اسـتـعـبدتم, مـتى	12
Por et al., 2012 [22]	Tom's leg is injured by falling.	Tom's leg ▌ is ▌injured▌ by falling.	10
Taleby Ahvanooey and Tabasi, 2014 [15]	Tom's leg is injured by falling.	Tom's leg is injured by falling.	4
Taleby Ahvanooey et al., 2015 [16]	Tom's leg is injured by falling.	Tom's leg is injured by falling.	32
Taleby Ahvanooey et al., 2016 [11]	Tom's leg is injured by falling.	Tom's leg is injured by falling.	Total bits of watermark
Alotaibi and Elrefaei, 2016 [19]	Tom's leg is injured by falling.	Tom's ▌leg is ▌injured ▌▌by falling.	20
Rizzo et al., 2016 [29] Rizzo et al., 2017 [48]	All the world	All ▌the ▌World	10

In addition, we evaluated the selected techniques in terms of criteria by implementing them on a simulated dataset. This dataset is made by copying randomly two sentences from referenced websites as depicted in Table 13. The detailed structures of copied texts are summarized in Table 14.

Assume that we aim to protect the documents in the dataset by hiding a watermark binary (60 bits) into their text contents. Therefore, we can analyze the efficiency of selected techniques in terms of criteria. Table 15 indicates the embedding capacity of evaluated techniques, which are calculated by using (1). Moreover, Figure 5 illustrates the embedding capacity of evaluated structural techniques.

As shown in Table 15 and Figure 4, the embedding capacity evaluation results conducted on the dataset demonstrate that some techniques provide high capacity and others are not able to hide whole of the watermark bits (60 bits). For example, in the Por et al. (2012), it is able to embed 148 bits into (Doc1), 150 bits (Doc2), 12 bits (Doc3), and 86 bits (Doc4).

Assuming that if a malicious user tampers a character or a word of the watermarked text content, then whether the watermark bits can be detected from the watermarked text by extraction algorithm?

To answer this question, we evaluated the approximate distortion robustness of each technique based on the embedding locations and the document features in Table 14, by using (2) separately. The DR evaluation results are shown in Table 16. In addition, Figure 6 illustrates both the average capacity and the distortion robustness of evaluated techniques.

Table 17 depicts a comparative analysis of structural techniques in terms of criteria and language compatibility along with their limitations. Although the structural techniques have been improved especially in invisibility and embedding capacity, they still have modest robustness and are vulnerable to tampering and retyping attacks. As shown in Tables 4 and 17, we analyzed the distortion robustness criterion of evaluated techniques according to their limitations against tampering attacks by considering the probability of losing the embedded watermark bits. Furthermore, we evaluated the embedding capacity of each technique based on its embedding locations (bits per doc) and the invisibility of each technique is rated based on the difference between the original text and the watermarked text.

In order to highlight the merits and demerits of evaluated techniques, six types of conventional attacks are considered

Table 13: Text document examples.

References	Name	Copied text content
http://ww.yjc.ir	Doc 1	به گزارش خبرنگار فوتبال و فوتسال گروه ورزشی باشگاه خبرنگاران جوان؛ مهدی طارمی مهاجم محروم پرسپولیس از باشگاه الغرافه قطر پیشنهاد دریافت کرده است و این باشگاه قطری طی نامه ای از باشگاه پرسپولیس درخواست کردند که رقم پیشنهادی خود را برای انتقال این بازیکن اعلام کنند. طبق برنامه جلسه هیئت مدیره باشگاه پرسپولیس امروز برگزار می شود و قرار است در مورد انتقال طارمی به باشگاه قطری تصمیم گیری شود.
https://www.nytimes.com	Doc 2	WASHINGTON—White House officials on Monday mustered a sweeping defense of their less-is-more public disclosure practices, arguing that releasing information on a wide array of topics would strike a blow against personal privacy and impede President Trump's ability to govern. This stance, critics say, represents a shift from Mr. Trump's own drain-the-swamp campaign message and his promise to decrease the influence of lobbyists, special interest groups and big political donors
http://www.chinadaily.com	Doc 3	其实, 只要从宪法法院法官的“政治倾向”上来看, 弹劾现任总统绝不是一件容易的事。因为宪法法院的九名法官 (现在在任八名) 中需要六名以上持“赞成”意见才能弹劾总统
http://www.spiegel.de	Doc 4	Der spanische König Felipe VI. hat die politischen Entscheidungsträger in Katalonien zu verantwortlichem Handeln aufgerufen. “In Katalonien darf der Weg nicht erneut zu Konfrontation oder Ausschluss führen”, warnte Felipe in seiner Weihnachtsansprache, die am Abend vom spanischen Staatsfernsehen ausgestrahlt wurde

Table 14: The detailed structures of samples copied texts.

Name	Characters	Dots (.)	Punctuation characters	Words	Spaces	Paragraphs	Lines	Language
Doc 1	390	2	3	71	70	1	3	Persian
Doc 2	482	3	14	70	70	1	4	English
Doc 3	81	2	10	79	2	1	3	Chinese
Doc 4	316	3	7	40	39	1	3	German

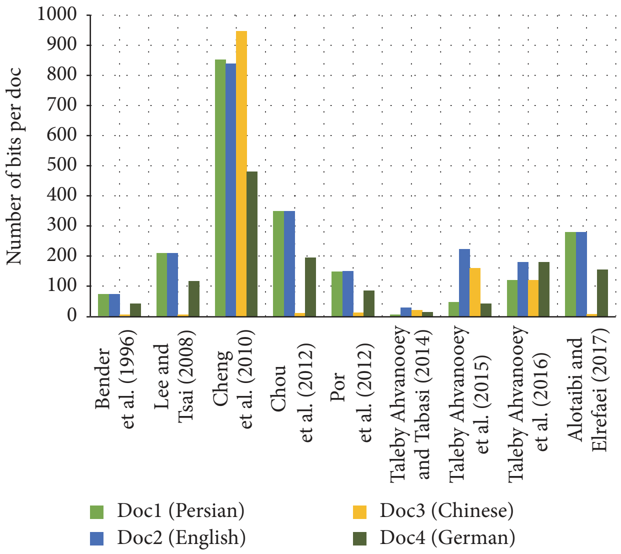


Figure 5: The embedding capacity of structural techniques (bits per doc).

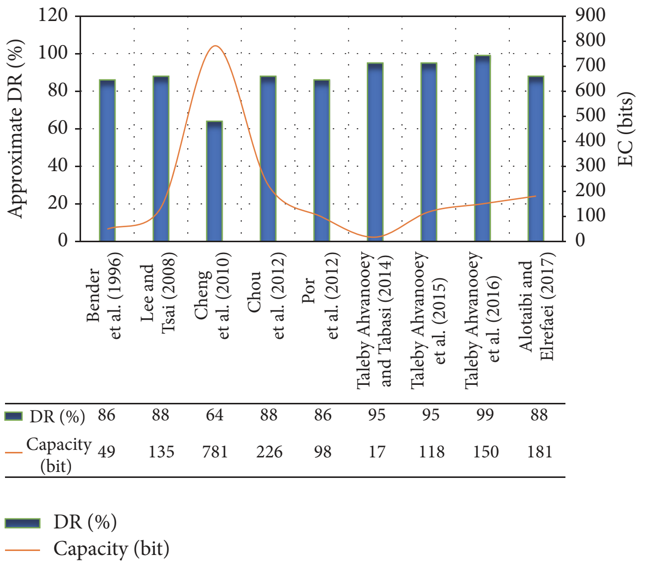


Figure 6: The overlap between the average embedding capacity and DR results.

for evaluating their limitations such as insertion, removal, reordering, reformatting, retyping, and copy and paste attacks. Assume that a malicious user copies a portion (or whole) of watermarked text which contained the watermark string in a new host file and randomly alters it in terms of conventional attacks. In this case, if even one bit of the watermark is changed, then it causes the detection of the watermark string by the corresponding extraction algorithm to fail. The evaluation results conducted on the watermarked texts are listed in Table 18.

As depicted in Table 18, almost all the evaluated techniques have some different limitations; however, some of them provide more safety than others. In practice, the programmers must consider the priority of criteria in case of fragile or robust and, thus, select a suitable technique based on the security limitations which can provide more safety in that application.

4. Suggestions for the Future Works

Information hiding is a very powerful and flexible technique that can be employed in various ways to protect valuable information in different areas such as copyright protection, secure communication, and authentication. Although the efficiency of text hiding techniques has drawn attention much from academic researchers, it is still lacking a precise analysis modeling that can take the intrinsic criteria of the text hiding industry into account during the evaluating efficiency. As we already pointed out, there are four common criteria for efficiency analysis, which are dependent on the way of embedding. In other words, the embedding methods generally determine how to analyze the efficiency of the text hiding techniques.

Therefore, to evaluate the efficiency of a certain algorithm, it is required to be compared with previous works within the same category (e.g., linguistic or structural). In addition, we have outlined some various limitations of two major categories of text hiding techniques in Table 19, which provide a better understanding of the state-of-the-art and hopefully help in developing future works.

Practically, the linguistic techniques have more limitations compared to structural techniques. Due to extra dictionaries (WordNet) and high computational cost, moreover, a few researchers focused on linguistic based methods in recent years. Over the last two decades, many structural techniques have been introduced to improve the efficiency of text hiding techniques by considering the optimum trade-offs between criteria. However, the robustness of these techniques needs to be more improved against tampering attacks in terms of security requirements. In the following, we suggest some

Table 15: Embedding capacity analysis of structural techniques (bits per doc).

Algorithm name	Doc 1	Doc 2	Doc 3	Doc 4	Average capacity	Summary of embedding methods (BPL)
Bender et al., 1996 [44]	73	74	5	42	49	One bit per locations (interword spaces, end of the lines and between sentences)
Lee and Tsai, 2008 [42]	210	210	6	117	135	3 bits per locations (interword spaces)
Cheng et al., 2010 [45]	852	840	948	480	781	12 bits per locations (word colors)
Chou et al., 2012 [46]	350	350	10	195	226	5 bits per locations (interword spaces)
Por et al., 2012 [22]	148	150	12	86	98	2 bits per locations (interword, intersentence, end-of-line, and end of paragraph)
Taleby Ahvanooey and Tabasi, 2014 [15]	6	28	20	14	17	2 bits per locations (after punctuation characters)
Taleby Ahvanooey et al., 2015 [16]	48	224	160	42	118	16 bits per locations (after punctuation characters, between sentences and beginning of the paragraphs)
Taleby Ahvanooey et al., 2016 [11]	120	180	120	180	150	Total bits of watermark string per locations (one time watermark bits after dots (.))
Alotaibi and Elrefaei, 2016 [19] (second)	280	280	8	156	181	4 bits per locations (interword spaces)

Table 16: Approximate DR (%) of structural techniques against tampering attacks.

Algorithm name	Doc 1	Doc 2	Doc 3	Doc 4	Average robustness
Bender et al., 1996 [44]	81	85	93	86	≅86
Lee and Tsai, 2008 [42]	82	85	97	88	≅88
Cheng et al., 2010 [45]	81	85	02	87	≅64
Chou and et al., 2012 [46]	82	85	97	87	≅88
Por et al., 2012 [22]	81	84	93	86	≅86
Taleby Ahvanooey and Tabasi, 2014 [15]	99	97	88	98	≅95
Taleby Ahvanooeyet al., 2015 [16]	99	97	88	98	≅95
Taleby Ahvanooeyet al., 2016 [11]	99	99	98	99	≅99
Alotaibi and Elrefaei, 2016 [19]	82	85	97	88	≅88

directions aimed at guiding cyber security researchers on the best options to utilize various types of text hiding techniques depending on the characteristics of the applications. However, we have to mention that these suggestions are general and empirically derived rules of thumb; these guidelines must not be considered rigidly or dogmatically.

(i) Where the main concern is protecting the valuable documents against retyping attacks, the NL based technique is the best tool to provide that requirement.

(ii) Wherein the main concern is protecting digital text documents against tampering, reformatting, and reordering attacks, the structural techniques can be applied as a fragile or robust tool for different applications (e.g., copyright protection, authentication, and proof of ownership).

(iii) Since the zero-width characters provide high invisibility and compatibility with other languages in different file formats (e.g., web, Word, and PDF), they can be used as an imperceptible way in order to hide secret information through the Unicode digital texts.

(iv) Provide high/low robustness by considering the specific locations of the text that have high/low distortion probability against tampering attacks (e.g., at the end of the sentences or first of the paragraphs)

(v) Use new binary encoding (lossless compression) algorithms to convert a watermark string to a binary string (2 bits, 3 bits, 4 bits, etc.).

(vi) Hash functions could be used to secure the watermark bits against unpredictable estimate based attacks.

(vii) The structural techniques can be applied as a security tool in the version control systems (VCS) for protecting the open source programs against reverse engineering.

(viii) The ideal text watermarking algorithm should provide optimum trade-offs among the three criteria (embedding capacity, invisibility, and robustness) to achieve high-level of security.

(ix) To sum up, which kind of techniques provides more accuracy for copyright protection of text documents? We cannot give a precise and perfect answer to this question. The researchers must take into account many things like various merits and demerits of text hiding techniques, together with the guidelines that we have collected. In addition, they should ponder whether the text hiding techniques could be appropriate or not for their applications. When the researcher realizes that some of the merits of a specific technique can provide a valuable benefit to the specific needs of the application at issue; thus it should probably be given a try.

5. Conclusion

This case study provides a comparative analysis of existing information hiding techniques, especially on those ones that are focused on altering the structure and content of digital texts for copyright protection. We looked at a range of available approaches and attacks over the digital text documents in order to explain current security issues in the copyright protection industry. Moreover, we outlined two categories of text watermarking techniques based on how to process digital texts to embed the watermark bits, namely, linguistic (or natural language) and structural (format based). Linguistic techniques alter the text content and sometimes even the original meaning of sentences for embedding the watermark, which is not desirable and hence hard to apply. Using this kind of methods is not suitable to protect sensitive documents. The structural techniques utilize some characteristics of text such as layout features (e.g., interwords spaces and interline spaces), and format (e.g., text color, text font, and text height). Format based methods do not retain the watermark against reformatting, conversion, and even sometimes a simple copy of the text into another file. Those structural techniques utilize Unicode control characters for embedding (e.g., zero-width spaces and special spaces) the watermark bits into the original text and are able to protect the watermarked text against reformatting, tampering, and copy attacks to some extent. This kind of techniques can be applied to sensitive documents due to having shown a greater degree of imperceptibility and optimum robustness. Finally, we have suggested some of the guidelines and directions that could merit further attention in future works.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Table 17: Structural watermarking techniques criteria analysis.

Algorithm name	Embedding capacity	Invisibility	DR	Limitations	Language compatibility
Bender et al., 1996 [44]	Low	Middle	Low	Unpleasant gaps between words	Multilingual
Lee and Tsai, 2008 [42]	Low	Middle	Modest	Unpleasant characters between words	Multilingual
Cheng and et al., 2010 [45]	Very high	Middle	Modest	Highlight worlds	Multilingual
Gutub and et al., 2007 [49]	Low	Imperceptible	Modest	Unpleasant wide "Keshida" between words	Exclusive (Arabic/Persian)
Gutub and et al., 2010 [47]	Medium	Imperceptible	Modest	Unpleasant wide "Keshida" between words	Exclusive (Arabic/Persian)
Alginahi et al., 2013 [20]	Medium	Imperceptible	Modest	Unpleasant wide "Keshida" between words	Exclusive (Arabic/Persian)
Alginahi et al., 2014 [21]	High	Imperceptible	Modest	Unpleasant wide "Keshida" between words	Exclusive (Arabic/Persian)
Chou et al., 2012 [46]	Medium	Imperceptible	Modest	Unpleasant gaps between words	Multilingual
Por et al., 2012 [22]	Medium	Imperceptible	Modest	Unpleasant gaps between words	Multilingual
Mir, 2014 [12]	High	Imperceptible	Modest	Only applicable to HTML files	Multilingual
Taleby Ahvanooey and Tabasi, 2014 [15]	Low	Imperceptible	High	Depends on punctuation characters	Multilingual
Taleby Ahvanooey et al., 2015 [16]	Medium	Imperceptible	High	Depends on punctuation characters	Multilingual
Alotaibi and Elrefaei, 2016 [43]	Low	Imperceptible	Modest	Depends on pointed and unpainted characters	Exclusive (Arabic/Persian)
Taleby Ahvanooey et al., 2016 [11]	High	Imperceptible	High	Depends on dots (.) characters	Multilingual
Alotaibi and Elrefaei, 2017 [19]	Low	Imperceptible	Modest	Unpleasant wide "Keshida" and gaps between words	First: (Arabic/Persian) Second: (Multilingual)
Rizzo et al. 2016 [29] Rizzo et al. 2017 [48]	High	Imperceptible	Low	Depends on font type (homoglyph Unicode characters) and unpleasant gaps between words	Exclusive (English)
Al-Nofaei et al. (2016) [50]	Medium	Imperceptible	Modest	Unpleasant wide "Keshida" and gaps between words	Exclusive (Arabic/Persian)

Table 18: A comparative analysis of evaluated techniques against conventional attacks.

Algorithm name	Security limitations	Robustness against conventional attacks: yes (√) and no (×)					
		Insertion	Removal	Reformatting	Reordering	Retyping	Copay & paste
Topkara et al. (2006) [34]	Medium safety (3)	×	×	√	×	√	√
Topkara et al. (2006) [35]	Medium safety (3)	×	×	√	×	√	√
Meral et al. (2007 & 2009) [36, 51]	Medium safety (3)	×	×	√	×	√	√
Kim (2008) [37]	Medium safety (3)	×	×	√	×	√	√
Kim et al. (2010) [38]	Medium safety (3)	×	×	√	×	√	√
Halvani et al. (2013) [39]	Medium safety (3)	×	×	√	×	√	√
Mali et al. (2013) [40]	Medium safety (3)	×	×	√	×	√	√
Lu et al. (2009) [41]	Medium safety (3)	×	×	√	×	√	√
Bender et al. (1996) [44]	Medium safety (3)	√	×	√	×	×	√
Lee and Tsai (2008) [42]	Medium safety (3)	√	×	√	×	×	√
Cheng et al. (2010) [47]	Easy to lose (1)	√	×	×	×	×	×
Gutub et al. (2007) [49]	Medium safety (3)	√	×	√	×	×	√
Gutub et al. (2010) [47]	Medium safety (3)	√	×	√	×	×	√
Alginahi et al. (2013) [20]	Medium safety (3)	√	×	√	×	×	√
Alginahi et al. (2014) [21]	Medium safety (3)	√	×	√	×	×	√
Chou et al. (2012) [46]	Medium safety (3)	√	×	√	×	×	√
Por et al. (2012) [22]	Medium safety (3)	√	×	√	×	×	√
Mir (2014) [12]	Unsafe (0)	×	×	×	×	×	×
Taleby Ahvanooey and Tabasi (2014) [15]	Optimum safety (4)	√	×	√	√	×	√
Taleby Ahvanooey et al. (2015) [16]	Optimum safety (3)	√	×	√	√	×	√
Alotaibi and Elrefaei (2016) [43]	Medium safety (3)	√	×	√	×	×	√
Taleby Ahvanooey et al. (2016) [11]	Optimum safety (4)	√	×	√	√	×	√
Alotaibi and Elrefaei (2017) [19]	Medium safety (3)	√	×	√	×	×	√
Rizzo et al. (2016) [29] Rizzo et al. (2017) [48]	Easy to lose (2)	×	×	√	×	×	√
Al-Nofaei et al. (2016) [50]	Medium safety (3)	√	×	√	×	×	√

Table 19: Comparison of the two major techniques (linguistic and atructural).

Factors	Linguistic (natural language)	Structural
Language compatibility	Exclusive special language	Multilingual
Embedding capacity	Low	Medium and high
Meaning alteration of text content	Alters the meaning	No effect on text content
Invisibility	Imperceptible	Imperceptible
Computational complexity	Very large (due to the search algorithm in dictionary and replacement of words)	Medium (embedding in special locations)
Robustness against conventional attacks	Low	Modest
Security	Medium safety	Optimum safety

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (no. 61374186), National Key Research and Development Program: Key Projects of International Scientific and Technological Innovation Cooperation between Governments (no. 2016YFE0108000), CERNET Next-Generation Internet Technology Innovation Project (NGII20160122), Project of ZTE Cooperation Research (2016ZTE04 11), and Jiangsu Province Key Research and Development Program (BE2017100, BE2017739).

References

[1] Z. Jalil and A. M. Mirza, "A review of digital watermarking techniques for text documents," in *Proceedings of the Proceeding of the International Conference on Information and Multimedia Technology (ICIMT '09)*, pp. 230–234, 2009.

[2] M. Pal, "A survey on digital watermarking and its application," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 153–156, 2016.

[3] A. H. Abdullah, "Data security algorithm using two-way encryption and hiding in multimedia files," *International Journal of Scientific &amp; Engineering Research*, vol. 5, no. 12, pp. 471–475, 2014.

[4] M. A. Qadir and I. Ahmad, "Digital text watermarking: Secure content delivery and data hiding in digital documents," *IEEE Aerospace and Electronic Systems Magazine*, vol. 21, no. 11, pp. 18–21, 2006.

[5] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[6] M. H. Alkawaz, G. Sulong, T. Saba, A. S. Almazyad, and A. Rehman, "Concise analysis of current text automation and watermarking approaches," *Security and Communication Networks*, vol. 9, no. 18, pp. 6365–6378, 2016.

[7] N. A. A. S. Al-Maweri, R. Ali, W. A. Wan Adnan, A. R. Ramli, and S. M. S. A. Abdul Rahman, "State-of-the-art in techniques of text digital watermarking: Challenges and Limitations," *Journal of Computer Science*, vol. 12, no. 2, pp. 62–80, 2016.

[8] P. Singh and R. S. Chadha, "A survey of digital watermarking techniques, applications and attacks," *International Journal of Engineering and Innovative Technolog*, vol. 2, no. 9, pp. 165–175, 2013.

[9] M. Agarwal, "Text Steganographic Approaches: a comparison," *International Journal of Network Security and Its Applications*, vol. 5, no. 1, pp. 9–25, 2013.

[10] J. Guru and H. Damecha, "Digital Watermarking Classification: A Survey," *International Journal of Computer Science Trends and Technology*, vol. 2, no. 5, pp. 122–124, 2014.

[11] M. Taleby Ahvanooey, H. Dana Mazraeh, and S. H. Tabasi, "An innovative technique for web text watermarking (AITW)," *Information Security Journal*, vol. 25, no. 4-6, pp. 191–196, 2016.

[12] N. Mir, "Copyright for web content using invisible text watermarking," *Computers in Human Behavior*, vol. 30, pp. 648–653, 2014.

[13] K. F. Rafat and M. Sher, "Secure digital steganography for ASCII text documents," *Arabian Journal for Science and Engineering*, vol. 38, no. 8, pp. 2079–2094, 2013.

[14] E. Sruthi, A. Scaria, and A. T. Ambikadevi, "Lossless Data Hiding Method Using Multiplication Property for HTML File," *International Journal for Innovative Research in Science and Technology*, vol. 1, no. 11, pp. 420–425, 2015.

[15] M. Taleby Ahvanooey and S. H. Tabasi, "A new method for copyright protection in digital text documents by adding hidden unicode characters in persian/english texts," vol. 4, pp. 4895–4900, 2014.

[16] M. Taleby Ahvanooey, S. H. Tabasi, and S. Rahmany, "A Novel Approach for text watermarking in digital documents by Zero-Width Inter-Word Distance Changes," vol. 4, pp. 550–558, 2015.

[17] Z. Jalil and A. M. Mirza, "A robust zero-watermarking algorithm for copyright protection of text documents," *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers,Series A/Chung-kuo Kung Ch'eng Hsuch K'an*, vol. 36, no. 2, pp. 180–189, 2013.

[18] M. Bashardoost, M. S. M. Rahim, and N. Hadipour, "A novel zero-watermarking scheme for text document authentication," *Jurnal Teknologi*, vol. 75, no. 4, pp. 49–56, 2015.

[19] R. A. Alotaibi and L. A. Elrefaei, "Improved capacity Arabic text watermarking methods based on open word space," *Journal of King Saud University - Computer and Information Sciences*, 2016.

[20] Y. M. Alginahi, M. N. Kabir, and O. Tayan, "An enhanced Kashida-based watermarking approach for Arabic text-documents," in *Proceedings of the 2013 10th International Conference on Electronics, Computer and Computation, ICECCO 2013*, pp. 301–304, Turkey, November 2013.

[21] Y. M. Alginahi, M. N. Kabir, and O. Tayan, "An enhanced kashida-based watermarking approach for increased protection in arabic text-documents based on frequency recurrence of characters," *International Journal of Computer and Electrical Engineering*, vol. 6, no. 5, pp. 381–392, 2014.

[22] L. Y. Por, K. Wong, and K. O. Chee, "UniSpaCh: A text-based data hiding method using Unicode space characters," *The Journal of Systems and Software*, vol. 85, no. 5, pp. 1075–1082, 2012.

[23] M. Dalla Preda and M. Pasqua, “Software watermarking: a semantics-based approach,” *Electronic Notes in Theoretical Computer Science*, vol. 331, pp. 71–85, 2017.

[24] J. Gu and Y. Cheng, “A Watermarking scheme for natural language documents,” in *Proceedings of the 2010 2nd IEEE International Conference on Information Management and Engineering, ICIME 2010*, pp. 461–464, China, April 2010.

[25] R. J. Jaiswal and N. N. Patil, “Implementation of a new technique for web document protection using unicode,” in *Proceedings of the 2013 International Conference on Information Communication and Embedded Systems, ICICES 2013*, pp. 69–72, India, February 2013.

[26] T.-Y. Liu and W.-H. Tsai, “A new steganographic method for data hiding in microsoft word documents by a change tracking technique,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 24–30, 2007.

[27] A. A. Mohamed, “An improved algorithm for information hiding based on features of Arabic text: A Unicode approach,” *Egyptian Informatics Journal*, vol. 15, no. 2, pp. 79–87, 2014.

[28] N. A. Salem Al-maweri, W. A. Wan Adnan, A. R. Ramli, K. Samsudin, and S. M. Ahmad Abdul Rahman, “Robust Digital Text Watermarking Algorithm based on Unicode Extended Characters,” *Indian Journal of Science and Technology*, vol. 9, no. 48, pp. 1–14, 2016.

[29] S. G. Rizzo, F. Bertini, and D. Montesi, “Content-preserving Text Watermarking through Unicode Homoglyph Substitution,” in *Proceedings of the 20th International Database Engineering &amp; Applications Symposium (IDEAS ’16)*, pp. 97–104, 2016.

[30] Y. Zhang, H. Qin, and T. Kong, “A novel robust text watermarking for word document,” in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP ’10)*, pp. 38–42, October 2010.

[31] H. O. N. Hebah, “Digital watermarking a technology overview,” *International Journal of Research and Reviews in Applied Sciences*, vol. 6, no. 1, pp. 98–102, 2011.

[32] J. Klensin, “Unicode Control characters, Fileformate,” 2017, http://www.fileformat.info/info/unicode/char/search.htm.

[33] A. M. Alhusban and Q. O. Jehad, “A meliorated kashida based approach for arabic text steganography,” *International Journal of Computer Science and Information Technology*, vol. 9, no. 2, pp. 99–109, 2017.

[34] J. M. Topkara, U. Topkara, and M. J. Atallah, “Words are not enough: sentence level natural language watermarking,” in *Proceedings of the 4th ACM International Workshop on Contents Protection and Security (MCPS ’06)*, pp. 37–46, Santa Barbara, Calif, USA, October 2006.

[35] U. Topkara, M. Topkara, and M. J. Atallah, “The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions,” in *Proceedings of the Multimedia and Security Workshop (MM ’06)*, pp. 164–174, Geneva, Switzerland, September 2006.

[36] H. M. Meral, B. Sankur, A. Sumru Özsoy, T. Güngör, and E. Sevinç, “Natural language watermarking via morphosyntactic alterations,” *Computer Speech and Language*, vol. 23, no. 1, pp. 107–125, 2009.

[37] M. Y. Kim, “Text watermarking by syntactic analysis,” in *Proceedings of the 12th WSEAS International Conference on Computers, (ICC 08, World Scientific and Engineering Academy and Society, Heraklion*, pp. 904–909, Greece, 2008.

[38] M.-Y. Kim, O. R. Zaiane, and R. Goebel, “Natural language watermarking based on syntactic displacement and morphological division,” in *Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference Workshops, COMPSACW 2010*, pp. 164–169, Republic of Korea, July 2010.

[39] O. Halvani, M. Steinebach, P. Wolf, and R. Zimmermann, “Natural language watermarking for German texts,” in *Proceedings of the 1st ACM Workshop on Information Hiding and Multimedia Security, IH and MMSec 2013*, pp. 193–201, France, June 2013.

[40] M. L. Mali, N. N. Patil, and J. B. Patil, “Implementation of text watermarking technique using natural language watermarks,” in *Proceedings of the 3rd International Conference on Communication Systems and Network Technologies, CSNT 2013*, pp. 482–486, 2013.

[41] H. LU, M. Guangling, F. DingYi, and G. XiaoLin, “Resilient Natural Language Watermarking Based on Pragmatics,” in *Proceedings of the IEEE Youth Conference on Information, Computing and Telecommunication*, 2009, YC-ICT ’09.

[42] I. S. Lee and W. H. Tsai, “Secret communication through web pages using special space codes in HTML files,” *International Journal of Applied Science and Engineering*, vol. 6, pp. 141–149, 2008.

[43] R. A. Alotaibi and L. A. Elrefaei, “Utilizing word space with pointed and un-pointed letters for Arabic text watermarking,” in *Proceedings of the 18th UKSim-AMSS International Conference on Computer Modelling and Simulation, UKSim 2016*, pp. 111–116, 2016.

[44] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–335, 1996.

[45] W. Cheng, H. Feng, and C. Yang, “A robust text digital watermarking algorithm based on fragments regrouping strategy,” in *Proceedings of the 2010 IEEE International Conference on Information Theory and Information Security, ICITIS 2010*, pp. 600–603, China, December 2010.

[46] Y.-C. Chou, C.-Y. Huang, and H.-C. Liao, “A reversible data hiding scheme using cartesian product for HTML file,” in *Proceedings of the 2012 6th International Conference on Genetic and Evolutionary Computing, ICGEC 2012*, pp. 153–156, 2012.

[47] A. A.-A. Gutub, F. Al-Haidari, K. M. Al-Kahsah, and J. Hamodi, “E-text watermarking: Utilizing ’Kashida’ extensions in Arabic language electronic writing,” *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 1, pp. 48–55, 2010.

[48] S. G. Rizzo, F. Bertini, D. Montesi, and C. Stomeo, “Text Watermarking in Social Media,” in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM ’17)*, 2017.

[49] A. A.-A. Gutub, L. Ghouti, A. A. Amin, T. M. Alkharobi, and K. I. Mohammad, “Utilizing extension character “Kashida” with pointed letters 469 for Arabic text digital watermarking,” in *SECRYPT*, pp. 329–332, 2007.

[50] S. M. Al-Nofaie, M. M. Fattani, and A. A. A. Gutub, “Capacity Improved Arabic Text Steganography Technique Utilizing “Kashida” with Whitespaces,” in *Proceedings of the The The 3rd International Conference on Mathematical Sciences and Computer Engineering (ICMSCE 2016)*, pp. 38–44, 2016.

[51] H. M. Meral, E. Sevinç, E. Ünkar, B. Sankur, A. S. Özsoy, and T. Güngör, “Natural language watermarking via morphosyntactic alterations,” in *Proceedings of the SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents*, p. 65050X, San Jose, CA, USA, 2007.

[52] P. T. Daniels, “The Unicode Standard,” 2017, http://www.unicode.org/standard/standard.ht.

[53] M. Crispin, "Unicode, Wikipedia (the free encyclopedia)," 2017, https://en.wikipedia.org/wiki/Unicode.

[54] M. Shirali-Shahreza, "Pseudo-space Persian/Arabic text steganography," in *Proceedings of the 13th IEEE Symposium on Computers and Communications, ISCC 2008*, pp. 864–868, Morocco, July 2008.

[55] M. Kaur and K. Mahajan, "An existential review on text watermarking techniques," *International Journal of Computer Applications*, vol. 120, no. 18, pp. 29–32, 2015.

[56] 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, pp. 1181–1196, 1999.

[57] R. Petrović, B. Tehranchi, and J. M. Winograd, "Security of copy-control watermarks," in *Proceedings of the TELSIKS 2007 - 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pp. 117–126, Serbia, September 2007.

[58] Z. Jalil, A. M. Mirza, and T. Iqbal, "A zero-watermarking algorithm for text documents based on structural components," in *Proceedings of the International Conference on Information and Emerging Technologies (ICIET '10)*, pp. 1–5, Karachi, Pakistan, June 2010.

[59] M. Bashardoost, M. S. Mohd Rahim, T. Saba, and A. Rehman, "Replacement Attack: A New Zero Text Watermarking Attack," *3D Research*, vol. 8, no. 1, article no. 8, 2017.

[60] F. M.Ba-Alwi, M. M. Ghilan, and F. N. Al-Wesabi, "Content authentication of english text via internet using Zero Watermarking technique and Markov model," *International Journal of Applied Information Systems*, vol. 7, no. 1, pp. 25–36, 2014.

[61] M. Tanha, S. D. S. Torshizi, M. T. Abdullah, and F. Hashim, "An overview of attacks against digital watermarking and their respective countermeasures," in *Proceedings of the 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec 2012*, pp. 265–270, Malaysia, June 2012.

[62] A. A. A. Gutub and M. M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions," vol. 1, pp. 502–505, 2007.

[63] L. J. Li, H. Su, E. P. Xing, and F. F. Li, "Object bank: a high-level image representation for scene classification semantic feature sparsification," *NIPS Proceedings*, vol. 2, no. 3, p. 5, 2010.

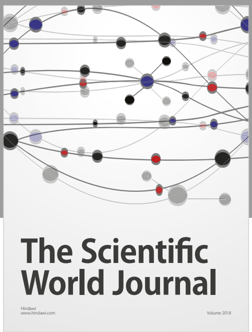
[64] J. J. Marin, M. Vergel, and A. Carnero, "Improved method of arabic text steganography using the extension "kashida" character," *Bahria University Journal of Information and Communication Technology*, vol. 3, no. 1, pp. 68–72, 2010.

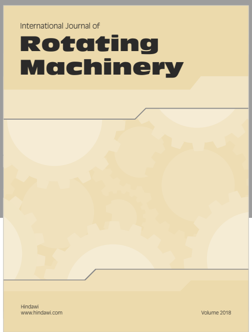
[65] A. Al-Nazer and A. Gutub, "Exploit Kashida Adding to Arabic e-Text for High Capacity Steganography," in *Proceedings of the IEEE 2009 Third International Conference on Network and System Security*, pp. 447–451, 2009.

[66] S. M. Al-Nofaie, M. M. Fattani, and A. A. A. Gutub, "Merging two steganography techniques adjusted to improve arabic text data security," *Journal of Computer Science and Computational Mathematics*, vol. 6, no. 3, pp. 60–65, 2016.

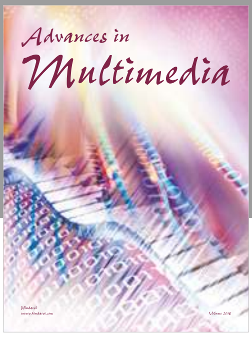
[67] C. P. Sumathi, T. Santanam, and G. Umamaheswari, "A Study of Various Steganographic techniques used for information hiding," *International Journal of Computer Science and Engineering Survey*, vol. 4, no. 6, pp. 9–25, 2013.

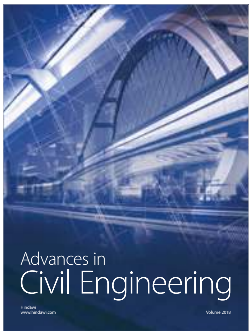
Journal of
Engineering

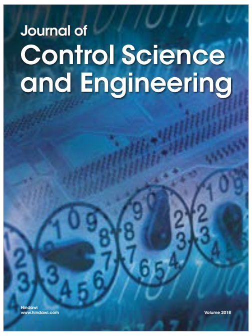

The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

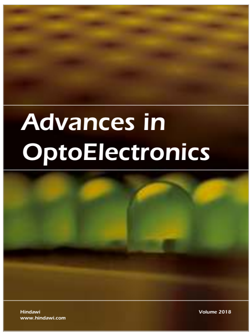

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

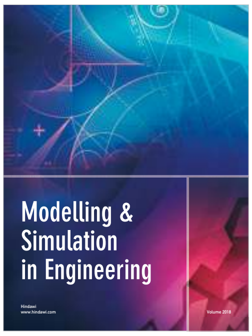
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

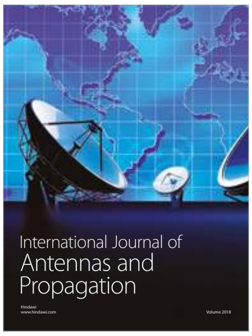
VLSI Design

International Journal of
Navigation and Observation

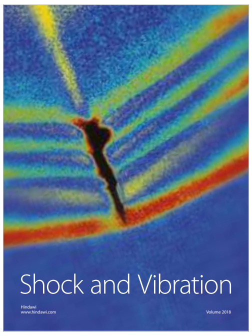

Modelling &
Simulation
in Engineering

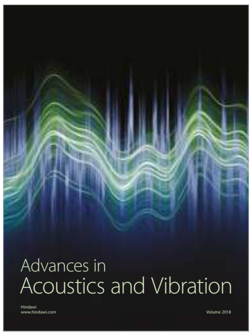
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration