Text Authorship Verification through Watermarking

Stefano Giovanni Rizzo, Flavio Bertini, Danilo Montesi
Department of Computer Science and Engineering
University of Bologna, Italy
{stefano.rizzo8, flavio.bertini2, danilo.montesi}@unibo.it

***Abstract*—While a plethora of digital contents are daily generated and shared online, authorship verification has become an imperative task. In comparison to other media watermarking techniques, text watermarking is a more challenging task. The changes in text would strongly affect the visual form and the meaning, text might be very short (eg. social media posts) and it cannot be always converted into image. In this paper we propose a novel text watermarking method for authorship verification based on Unicode confusable substitution. The proposed method substitutes latin symbols with homoglyph characters. It ensures length preservation and visual indistinguishability among the original text and the watermarked one. We successfully evaluate our approach using a real dataset of 1.8 million of New York Times articles. The results show the effectiveness of our method providing an average length of 101 characters needed to embed a 64bit password based watermark.**

***Keywords*—Authorship Analysis, Copyright Protection, Tampering Detection, Text Watermark, Unicode Confusable**

I. Introduction

Recently, the proliferation of social media and mobile devices are contributing to a massive sharing of digital contents. Many of these digital contents are copyrighted and the authors demand that their intellectual property rights are correctly protected. There are various illicit actions concerning digital contents, like tampering, forgery, illegal copying and theft to name a few. Consequently, different techniques are proposed to tackle illegitimate uses of digital contents. In particular, they can be classified into three different categories: cryptography, steganography and watermarking [10]. Watermarking algorithms are the most balanced for sharing non obfuscated information combined with authorship preservation [21].

In comparison to watermarking techniques for other digital contents, text watermarking is the most difficult task. In particular, a text watermarking algorithm must work with some additional constraints, as short-length message, a limited set of transformations in order to preserve readability and a restricted number of alternative syntactic and semantic permutations.

Text watermarking algorithms can be classified as following:

- *Image-based Techniques* - Firstly the text is transformed in an image, then the watermark is embedded into the image.
- *Syntactic Techniques* - These methods transform the language depending structures in order to hide the watermark.
- *Semantic Techniques* - These methods use verbs, nouns, prepositions, but even spelling and grammar rules to permute the contents and embed the watermark.
- *Structural Techniques* - These methods exploit double letters occurrences, words-shift and lines-shift encoding and Unicode symbols to embed the watermark.

In this paper, we propose a structural text watermarking method for authorship verification. In accordance with literature [11], the proposed method is *invisible* and *detectable* and belongs to *fragile*, *blind* and *zero watermarking* classes.

Compared to other text watermarking approaches, our method leaves unaltered the readable content and preserve the length of the original text. This allows to use our text watermarking approach in several new contexts, for example short message communications and social networks posts.

The embedding process consists in two phases. First, a hash function combines the author password and the original text to generate the watermark. In the second phase, the watermark is embedded in the original text by substituting the latin symbols with homoglyph characters in the Unicode set [5]. When the text is copied and pasted elsewhere, the watermark is brought along with it, allowing authorship verification[1].

In order to establish the minimum length requirement on real text examples, we provide the results of an extensive experiments on 1.8 million of New York Times articles [18]. The results show that, on average, 101 characters are sufficient to embed the watermark preserving length and visible aspects of the original text.

The rest of the paper is organized as follows. In Section II, we review literature works related to text watermarking methods. In Section III, we describe our text watermarking method. We discuss the evaluations of our method in Section IV. Some concluding remarks are made in Section V.

II. Related works

Text watermarking approaches in literature have been classified in four categories [9], [11]: image-based, syntactic, semantic and structural.

a) Image-based method: in this widely used approach, the text is transformed into an image and then the watermark is hidden by modulating the luminance of pixels [3] or by changing the histogram [13]. However, changes in text images can be more perceptible than in pictures [6], thus most of the image-based works leverage the peculiarities of text. An effective method is the horizontal and vertical shifting of words and lines accordingly to the watermark data [4], [14]. A similar methods are to alter the inter-word spaces [8], [12]

[1] A prototype can be tested here: http://smartdata.cs.unibo.it/watermark/

978-1-5090-2857-3/16 $31.00 © 2016 IEEE
DOI 10.1109/EISIC.2016.28

or the characters' strokes and serifs [1], [4]. The image-based approaches can be considered a workaround to text watermarking and it would be unnatural and impractical in many actual scenarios, such as blogging or texting.

b) Syntactic method: in this approach the syntactic structure of natural language text is transformed with operations such as clefting or passivization [2]. Other meaning-preserving morpho-syntactic transformation can be also applied [15]. The overall text content is strongly altered and may not reflect the author's original message. Moreover, this approach may be not suitable in limited context (e.g. Twitter posts, SMS texts).

c) Semantic method: this approach, as the previous one, makes use of Natural Language Processing to embed the watermark. In [20] the similitude of terms are leveraged to substitute words accordingly to the watermark. Sometimes the semantic approach is mixed with the syntactic one and a bigger transformation space is obtained [19]. Similarly to syntactic approaches, these methods produce a visibly different document and alter the author's content.

d) Structural method: in this category the watermark is embedded using invisible Unicode symbols. The Unicode standard provides many different white spaces encoding. These different white space symbols have been used to encode long payloads in Microsoft Word documents [17]. Other than whitespaces, some totally invisible Unicode symbols, that do not fill spaces in text, have been used to embed watermark in HTML content [16]. Unlike the syntactic and semantic approaches, these works preserve the text content. However the application contexts are still restricted and the text length is not preserved, since an overhead of symbols are added in the embedding process.

III. Our approach

In this paper we propose a text watermarking method that preserves both the visual appearance and the length of the original text. Our approach replaces spaces and characters with visually indistinguishable Unicode symbols. The watermark can be only detected and extracted using a predefined symbols mapping. Nonetheless, the watermark is verifiable, meaning that the authorship can be only proved by the owner of the password.

A. Unicode confusables

Among the 112 thousand of Unicode symbols currently used, some symbols are totally or partially indistinguishable from others. However, these similar symbols (homoglyphs) have a different numerical representation. Recently, homoglyph symbols have been used also for steganography tasks [7]. The Unicode Consortium is aware of the security problems arising from the similarity between different symbols [5] and maintains a list of confusable symbols[2]. Among the confusable set we identified the most visually similar symbols for white space and latin letters. We tested the homoglyphs under the most used font families in modern desktop and web applications. Then, we selected those symbols that have obtained the most imperceptible differences. That gives rise to a set of 8 white space similar symbols, Table I, and a set of 16 letters and punctuations with one almost or totally indistinguishable associated symbol, Table II.

TABLE I
The white space selected symbols. Eight different white spaces are used to encode a sequence of three bits.

White space	Bits	Unicode
Space	000	0x0020
En Quad	001	0x2000
Three-per-em Space	010	0x2004
Four-per-em Space	011	0x2005
Punctuation Space	100	0x2008
Thin Space	101	0x2009
Narrow No-break Space	110	0x202f
Medium Mathematical Space	111	0x205f

TABLE II
The subset of letters and punctuations selected symbols. Original Unicode and Duplicate Unicode are used to encode bit 0 and bit 1, respectively.

	Bit 0	Bit 1
Symbol	Original Unicode	Duplicate Unicode
-	0x002d	0x2010
;	0x003b	0x037e
C	0x0043	0x216d
D	0x0044	0x216e
K	0x004b	0x212a
L	0x004c	0x216c
M	0x004d	0x216f
V	0x0056	0x2164
X	0x0058	0x2169
c	0x0063	0x217d
d	0x0064	0x217e
i	0x0069	0x2170
j	0x006a	0x0458
l	0x006c	0x217c
v	0x0076	0x2174
x	0x0078	0x2179

B. Password based watermark generation

Our watermarking method is mainly conceived for authorship verification of text. In order to provide secure authorship verification we rely on a cryptographic hash function, that is a one-way function that takes a text t, a password k and generates a fixed size string, namely the hash value. The robustness of the algorithm and the size of the produced hash value represent two important properties that drives the choice of the hash function. In particular, in order to be capable of watermarking also short texts we need a small payload (hash value). Nevertheless, a smaller hash value can negatively affects the robustness of the function.

C. Unicode watermark embedding

The watermark w, generated using the hash function, is embedded in the original text t, replacing original symbols with their homoglyphs in a controlled way.
In Figure 1 is shown how the embedding algorithm scans each character in the original text looking for a confusable symbol.

[2] http://www.unicode.org/Public/security/8.0.0/confusables.txt

When a confusable symbol is found, it can be a white space or a symbol (letter or punctuation). In the first case, the white space is used to encode three bits using Table I: if the three bits are equal to 000 than the original white space is left, otherwise it is replaced by the correspondent Unicode space symbol in Table I. In the second case, when a letter or punctuation is found, the symbol is used to encode one bit using Table II. More precisely, if this bit is 0 the original symbol is left, otherwise it is replaced with its homoglyph symbol in Table II.

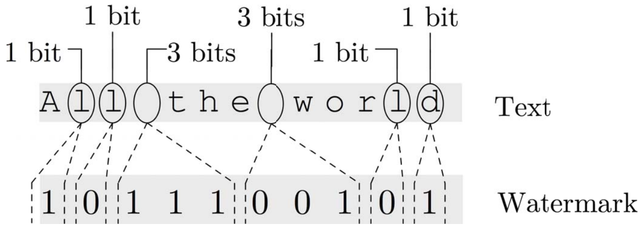


Fig. 1. The spaces in Table I and the symbols with a *duplicate* in Table II are used to embed the watermark w.

D. Unicode watermark extraction

The extraction algorithm is rather straightforward. Each confusable symbol found in the watermarked text that belongs to Table I or Table II is extracted. If it is in standard format then the hash bit is "000" for white spaces or "0" for letters and punctuations, otherwise it is one of the 7 non-zero bits in Table I for white spaces or "1" for letters and punctuations in Table II. Thus, the hash value w is reconstructed from the most significant bit to the least significant bit.

The embedding and extraction algorithms cost is constant in the best case, depending only on the number of bits in the hash, while linear on the number of characters in the text in the worst case scenario.

E. Authorship verification

The authorship verification process allows the original text owner, who generated the watermark, to prove his authorship. The process requires to be robust, that is no attacker with limited computational resources should be able to success in the process. Non-blind watermarking methods ensure this robustness through the impossibility to extract the watermark without the original text. Our method is a blind watermarking and an attacker can reconstruct the watermark using our Unicode mapping. As a result, the robustness is provided through the owner password and the hash function. Therefore, the authorship verification depends on the reproducibility of the extracted watermark, which is possible for the password owner only.

In Figure 2 the authorship verification process is described in the 3 following steps:

1) Given the watermarked text, both the author and the attacker can extract the original text t and the watermark w using the mapping tables.

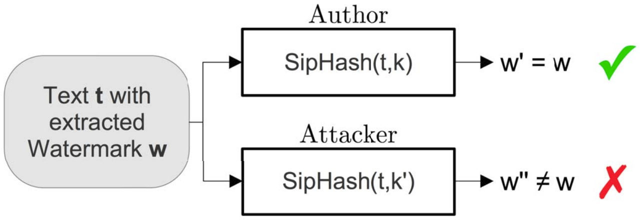


Fig. 2. The password k is a proof of authorship. Once the watermark w is extracted, only the author with the original password k is able to reproduce it.

2) The author of the watermark apply the SipHash function to the text t using his password k, thus obtaining a watermark w'. The attacker apply the SipHash function on t as well, however using a wrong password k', obtaining a watermark w''.
3) The reproduced watermarks w' and w'' are compared to the original watermark w, proving the authenticity of the key k and as a result the authorship of its owner.

While the chosen SipHash MAC hash function is considered to be secure in most contexts, given enough time or computational power for the attacker or new developments in cryptographic algorithms could raise the hash function requirements. For this reason we show in Table III alternative hash functions of different sizes and the related boundaries and averages for the text length requirement.

IV. Evaluation

The number of characters required to embed a watermark strictly depends on how many confusable symbols are among those characters. More confusable symbols in a text means less characters needed to embed the payload (especially if the confusables are white spaces, accounting for more embedding capacity).

In order to investigate the text length requirements in real world texts, we take in consideration the New York Times Corpus [18], a collection of 1.8 million articles from the New York Times newspaper. To stress the fact that this watermark approach can be successfully applied to small texts, we extracted the lead paragraph from all articles and experimented on these portions of text.

We tested the embedding algorithm using cryptographic hash functions of different hash length, from 64 bits (SipHash) to 224 bits (SHA-2), in order to statistically find out the probability of successfully embed a watermark. In Table III we show that using our approach it is possible to embed a SipHash watermark in the 94.2% of the lead paragraphs. Considering the other hash sizes, the percentage of successfully watermarked articles drop to 84.5% using a 128bit hash, 77.1% using a 160bit hash and 62.3% with a 224bit hash.

Apart from the percentage of watermarkable paragraphs, Table III shows that the minimum number of characters needed to encode the watermark is of only 46 characters (white space included), while on average the needed characters are less than

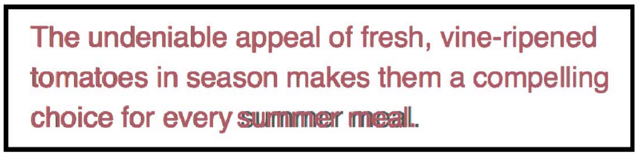
TABLE III

MINIMUM TEXT LENGTH FOR LENGTH-PRESERVING WATERMARKING ON THE NEW YORK TIMES CORPUS WITH DIFFERENT HASH LENGTHS.

	SipHash (64bit)	**MD5 (128bit)**	**SHA-1 (160bit)**	**SHA-2 (224bit)**
Success % on lead paragraphs (average length: 526 symbols)	94.2%	84.5%	77.1%	62.3%
Minimum number of characters needed	46	93	116	163
Maximum number of characters needed	101.3	197.7	246.5	344

101 (white space included). Doubling the hash size, on average the minimum length required is 198 characters to embed the 128 bits of an MD5 hash. In the most robust setting (SHA-2 with 224 bits) the minimum character needed are 344 on average.

In the watermark embedding process some of the original text characters are replaced with other symbols. It is crucial to show the watermarked text is indistinguishable from the original one. As shown in Figure 3, the differences between the original and the watermarked paragraphs are hardly noticeable by the human eye. In fact, the most perceptible difference regards the horizontal spacing. The reason is that the 8 different white spaces encoding differ in horizontal spacing and slightly affect the characters position.



The undeniable appeal of fresh, vine-ripened tomatoes in season makes them a compelling choice for every summer meal.

Fig. 3. A lead paragraph from a New York Times article: comparison of the original and the watermarked text by overlapping.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented a method for authorship verification of text through watermarking. In particular, the novel approach allows to embed a password based watermark in texts based on latin alphabet. The embedding process exploits homoglyph Unicode symbols to ensures two important features: visual-indistinguishability and length-preservation. We shown that we can embed a 64bit watermark with only a 46 characters long text.

These features make possible to apply the proposed method in several contexts, as: short message chats, blogging and social network posts. For this reason, we plan to extensively evaluate the method on different social networks and cross-platform instant messaging application.

VI. ACKNOWLEDGMENTS

This work has been supported in part by the Italian MIUR Ministry, OPLON and IMPACT projects (respectively SCN_00176, RBFR107725).

REFERENCES

[1] T. Amano and D. Misaki. A feature calibration method for watermarking of document images. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 91–94. IEEE, 1999.

[2] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding*, pages 185–200. Springer, 2001.

[3] A. K. Bhattacharjya and H. Ancin. Data embedding in text for a copier system. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 2, pages 245–249. IEEE, 1999.

[4] J. T. Brassil, S. Low, N. F. Maxemchuk, and L. O. Gorman. Electronic marking and identification techniques to discourage document copying. *Selected Areas in Communications, IEEE Journal on*, 13(8):1495–1504, 1995.

[5] M. Davis and M. Suignard. Unicode security mechanisms. Unicode technical standard #39, Unicode. http://www.unicode.org/reports/tr39/.

[6] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, 1999.

[7] S. Hosmani, H. R. Bhat, and K. Chandrasekaran. Dual stage text steganography using unicode homoglyphs. In *Security in Computing and Communications*, pages 265–276. Springer, 2015.

[8] D. Huang and H. Yan. Interword distance changes represented by sine waves for watermarking text images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(12):1237–1245, 2001.

[9] Z. Jalil and A. M. Mirza. A review of digital watermarking techniques for text documents. In *Information and Multimedia Technology, 2009. ICIMT'09. International Conference on*, pages 230–234. IEEE, 2009.

[10] S. Katzenbeisser and F. Petitcolas. *Information hiding techniques for steganography and digital watermarking*. Artech house, 2000.

[11] M. Kaur and K. Mahajan. An existential review on text watermarking techniques. *International Journal of Computer Applications*, 120(18), 2015.

[12] Y.-W. Kim, K.-A. Moon, and I.-S. Oh. A text watermarking algorithm based on word classification and inter-word space statistics. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 775–779. IEEE, 2003.

[13] Y.-W. Kim and I.-S. Oh. Watermarking text document images using edge direction histograms. *Pattern Recognition Letters*, 25(11):1243–1251, 2004.

[14] S. H. Low, N. F. Maxemchuk, and A. M. Lapone. Document identification for copyright protection using centroid detection. *Communications, IEEE Transactions on*, 46(3):372–383, 1998.

[15] H. M. Meral, B. Sankur, A. S. Özsoy, T. Güngör, and E. Sevinç. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, 23(1):107–125, 2009.

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

[17] L. Y. Por, K. Wong, and K. O. Chee. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082, 2012.

[18] E. Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[19] M. Topkara, C. M. Taskiran, and E. J. Delp III. Natural language watermarking. In *Electronic Imaging 2005*, pages 441–452. International Society for Optics and Photonics, 2005.

[20] 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 8th workshop on Multimedia and security*, pages 164–174. ACM, 2006.

[21] X. Zhou, W. Zhao, Z. Wang, and L. Pan. Security theory and attack analysis for text watermarking. In *E-Business and Information System Security, 2009. EBISS'09. International Conference on*, pages 1–6. IEEE, 2009.