Tamper-proofing of Electronic and Printed Text Documents via Robust Hashing and Data-Hiding

R. Villán, S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun

Computer Vision and Multimedia Laboratory - University of Geneva
24, rue du Général-Dufour - 1211 Geneva 4, Switzerland

Keywords: Authentication, Tamper-Proofing, Text Data-Hiding, Robust Text Hashing.

ABSTRACT

In this paper, we deal with the problem of authentication and tamper-proofing of text documents that can be distributed in electronic or printed forms. We advocate the combination of robust text hashing and text data-hiding technologies as an efficient solution to this problem. First, we consider the problem of text data-hiding in the scope of the Gel'fand-Pinsker data-hiding framework. For illustration, two modern text data-hiding methods, namely color index modulation (CIM) and location index modulation (LIM), are explained. Second, we study two approaches to robust text hashing that are well suited for the considered problem. In particular, both approaches are compatible with CIM and LIM. The first approach makes use of optical character recognition (OCR) and a classical cryptographic message authentication code (MAC). The second approach is new and can be used in some scenarios where OCR does not produce consistent results. The experimental work compares both approaches and shows their robustness against typical intentional/unintentional document distortions including electronic format conversion, printing, scanning, photocopying, and faxing.

1. INTRODUCTION

A text document authentication system aims at deciding whether a given text document is authentic or not. The decision about authenticity is performed at the global level, meaning that the system gives only a binary decision about the entire document: authentic or fake. On the contrary, if a system makes decisions at the local level (word-level, line-level, paragraph-level, etc.), we refer to it as a text document *tamper-proofing* system. Thus, text document authentication and tamper-proofing aim at verifying the authenticity of a text document and at indicating the local modifications, if the document is suspected to be a fake.

One possible solution to the document authentication problem consists in the generation of the document's hash based on the knowledge of a secret key K_H. This hash value is securely stored somewhere. For the authentication task, the hash value is computed again from the document under investigation and compared with the one that was stored. In the case of an authentic document, the two hash values should be identical. If not, the decision about non-authenticity of the document should be declared. Obviously, the hash function should be designed to withstand various intentional/unintentional legitimate modifications that might occur during the document's life cycle. At the same time, the hash function should be sensitive enough to various intentional malicious modifications. In image processing, the development of such hash functions is an active field of research known as *robust visual hashing*.[1] Contrarily to document authentication, where the hash is computed from the entire document, document tamper-proofing is based on the concept of local hashing. This means that a hash is computed from each local part of the document. In this manner, if the document is maliciously modified, the tamper-proofing technology is able to identify the local parts where the modifications were introduced. This characteristic is used to provide the user with some hints and evidence about the introduced modifications.

Basically, there exist three approaches to hash-based document authentication (and by extension to tamper-proofing) depending on where the hash is stored. These are: hash storage in an electronic database, hash storage onto the document itself using auxiliary special means such as 2D bar codes, special inks or crystals,

For further information contact S. Voloshynovskiy. E-mail: svolos@cui.unige.ch (http://sip.unige.ch)

magnetic stripes, memory chips, etc., and hash storage onto the document's content itself (also known as self-authentication) using data-hiding techniques. We refer to our previous publication[2] for more details about these approaches.

In this paper, we focus our attention on the self-authentication approach, which is very attractive for various reasons. First, the authentication of the document is performed directly without accessing a hash database. Second, the hash cannot be easily separated from the document like it is, if a dense 2D bar code is used for storing the hash. Finally, the self-authentication approach can be easily implemented into any modern text editing tool and the resulting document can be stored using a suitable electronic format.

The main concerns of the self-authentication approach are the limited data storage capacity offered by current text data-hiding methods, which is a direct consequence of the imposed constraints on the document's visible degradation, and the lack of reliable and secure robust text hashing functions.

Thus, the goal of this paper is two-fold. First, to address the problem of limited data storage capacity of current text data-hiding technologies. Second, to study the properties of possibly good candidates for robust text hashing. For the first goal, we consider the combination of independent text data-hiding methods. Such a strategy is a natural extension of the Gel'fand-Pinsker framework for text data-hiding.[3, 4] As for the second goal, we study two text hashing methods in order to establish their suitability for the self-authentication approach.

This paper is organized as follows. The theoretical formulation of self-authentication of documents is given in Section 2. The combination of independent text data-hiding methods to increase the data storage capacity is dealt in Section 3. Two methods for text hashing are explained in Section 4. Experimental results about the presented text hashing methods are given in Section 5. Finally, Section 6 concludes this paper and describes future research perspectives.

2. SELF-AUTHENTICATION OF DOCUMENTS

In this section, we give a formal definition of each component block of a self-authentication system based on robust hashing and data-hiding. We also indicate a sufficient condition for this system to work reliably.

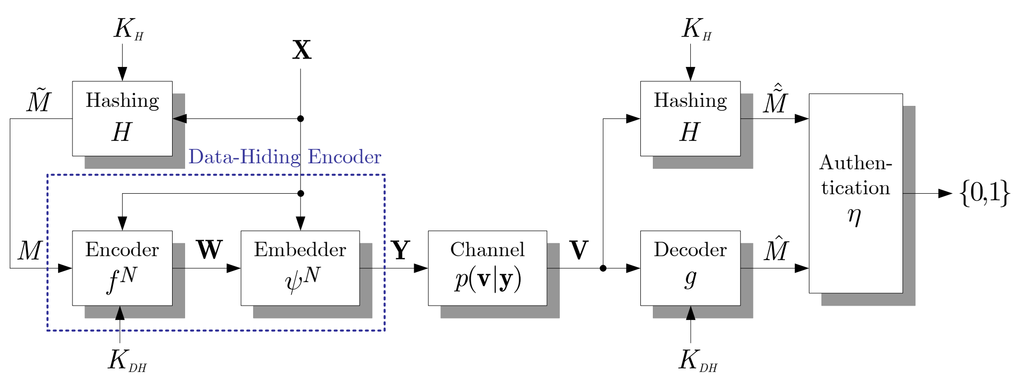


Figure 1. Document self-authentication via robust hashing and data-hiding.

Referring to Figure 1, the document protector has access to the host data $\mathbf{X}$ and to the uniquely assigned secret keys K_H and K_{DH}, which are uniformly distributed over the sets $\mathcal{K}_H = \{1, 2, \ldots, |\mathcal{K}_H|\}$ and $\mathcal{K}_{DH} = \{1, 2, \ldots, |\mathcal{K}_{DH}|\}$, respectively. We assume that $\mathbf{X} \sim p_{\mathbf{X}}(\cdot)$. The secret key K_H and the host data $\mathbf{X}$ are used to generate a hash message M that is encoded into the watermark $\mathbf{W}$ based on $\mathbf{X}$ and the secret key K_{DH}. The watermark $\mathbf{W}$ is embedded into the host data $\mathbf{X}$, resulting in the watermarked data $\mathbf{Y}$. The watermarked data $\mathbf{Y}$ is communicated through the attacking channel $p(\mathbf{v}|\mathbf{y})$ which introduces some legitimate distortions. For authenticating the document, the decoder outputs $\hat{M}$, an estimate of M, based on the attacked data $\mathbf{V}$ and K_{DH}. Additionally, the hash $\hat{\tilde{M}}$ is computed from $\mathbf{V}$ and K_H. Finally, the decision about the authenticity of $\mathbf{V}$ is made based on the comparison of $\hat{M}$ and $\hat{\tilde{M}}$. We assume that the message $M \in \mathcal{M}$ and the hash $\tilde{M} \in \tilde{\mathcal{M}}$

are uniformly distributed over $\mathcal{M} = \{1, 2, \ldots, |\mathcal{M}|\}$ and $\tilde{\mathcal{M}} = \{1, 2, \ldots, \left|\tilde{\mathcal{M}}\right|\}$, respectively. We also assume that $|\mathcal{M}| = 2^{NR_{DH}}$ and $|\tilde{\mathcal{M}}| = 2^{NR_H}$, where R_{DH} is the data-hiding rate, R_H is the hashing rate, and N is the length of all the involved vectors $\mathbf{X}$, $\mathbf{W}$, $\mathbf{Y}$ and $\mathbf{V}$.

The distortion function is defined as $d(\mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} d(x_i, y_i)$, where $d(x_i, y_i) : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}^+$ denotes an element-wise distortion metric between x_i and y_i.

Definition 1: A *discrete memoryless legitimate data-hiding channel* consists of four alphabets $\mathcal{X}$, $\mathcal{W}$, $\mathcal{Y}$, $\mathcal{V}$ and a probability transition matrix $p(\mathbf{v}|\mathbf{w}, \mathbf{x})$ that corresponds to the covert channel communication of the watermark $\mathbf{W}$ through the host image $\mathbf{X}$ (channel $p(\mathbf{y}|\mathbf{w}, \mathbf{x})$) and the attacking channel $p(\mathbf{v}|\mathbf{y})$ such that $p(\mathbf{v}|\mathbf{w}, \mathbf{x}) = \Sigma_{\mathbf{y}\in\mathcal{Y}^N} p(\mathbf{y}|\mathbf{w}, \mathbf{x})p(\mathbf{v}|\mathbf{y})$. The attacking channel is subject to the distortion constraint D^A:

$$\sum_{\mathbf{y}\in\mathcal{Y}^N} \sum_{\mathbf{v}\in\mathcal{V}^N} d(\mathbf{y}, \mathbf{v})p(\mathbf{v}|\mathbf{y})p(\mathbf{y}) \leq D^A, \tag{1}$$

where $p(\mathbf{v}|\mathbf{y}) = \prod_{i=1}^{N} p_{V|Y}(v_i|y_i)$.

Definition 2: A $(2^{NR_{DH}}, N)$ code for the data-hiding channel consists of a *message set* $\mathcal{M} = \{1, 2, ..., 2^{NR_{DH}}\}$, an *encoding function*:

$$f^N : \mathcal{M} \times \mathcal{X}^N \times \mathcal{K}_{DH} \to \mathcal{W}^N, \tag{2}$$

an *embedding function*:

$$\psi^N : \mathcal{W}^N \times \mathcal{X}^N \to \mathcal{Y}^N, \tag{3}$$

subject to the embedding distortion constraint D^E:

$$\frac{1}{|\mathcal{K}_{DH}||\mathcal{M}|} \sum_{k_{DH}\in\mathcal{K}_{DH}} \sum_{m\in\mathcal{M}} \sum_{\mathbf{x}\in\mathcal{X}^N} d(\mathbf{x}, \psi^N(f^N(m, \mathbf{x}, k_{DH}), \mathbf{x}))p_{\mathbf{X}}(\mathbf{x}) \leq D^E, \tag{4}$$

and a *decoding function*:

$$g : \mathcal{V}^N \times \mathcal{K}_{DH} \to \mathcal{M}. \tag{5}$$

We define the *average probability of error* for a $(2^{NR_{DH}}, N)$ code as:

$$P_e^{(N)} = \frac{1}{|\mathcal{K}_{DH}||\mathcal{M}|} \sum_{k_{DH}\in\mathcal{K}_{DH}} \sum_{m\in\mathcal{M}} \Pr\{g(\mathbf{V}, k_{DH}) \neq m | K_{DH} = k_{DH}, M = m\}. \tag{6}$$

Definition 3: A rate $R_{DH} = \frac{1}{N}\log_2|\mathcal{M}|$ is achievable for distortions (D^E, D^A), if there exists a sequence of $(2^{NR_{DH}}, N)$ codes with $P_e^{(N)} \to 0$ as $N \to \infty$.

Definition 4: The capacity of the data-hiding channel is the supremum of all achievable rates for distortions (D^E, D^A).

Theorem 1 (data-hiding capacity for a fixed channel)[5]: A rate R_{DH} is achievable for the distortion D^E and the fixed attacking channel $p(v|y)$ with bounded distortion D^A, iff $R_{DH} < C$, where:

$$C = \max_{p(u,w|x)} [I(U;V) - I(U;X)], \tag{7}$$

and U is an auxiliary random variable distributed over the set $\mathcal{U}$, with $|\mathcal{U}| \leq |\mathcal{X}||\mathcal{W}| + 1$.

Definition 5: A hash consists of a *hash set* $\tilde{\mathcal{M}} = \{1, 2, ..., 2^{NR_H}\}$ and a *hash function*:

$$H : \mathcal{K}_H \times \mathcal{X}^N \to \tilde{\mathcal{M}}. \tag{8}$$

The construction of a hash should satisfy several conflicting requirements. To analyze these constraints we assume that $\mathbf{X} = (X_1, X_2, ..., X_N)$ is a discrete memoryless source (DMS). The hash function produces the secure hash index $\tilde{M} \in \tilde{\mathcal{M}}$, i.e. a hash value, given K_H and $\mathbf{X}$. Contrarily to classical hashing,[6] where two vectors that differ in only a single bit have independent hash values, we require that two vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ that are perceived (respectively, understood) by the observer (respectively, by the reader) to be similar in some sense have the same hash value, even if $\mathbf{X}_1$ and $\mathbf{X}_2$ have small bit-level discrepancies. In practice, it also means that if a vector $\mathbf{X}_2$ is obtained via a mapping $p(\mathbf{x}_2|\mathbf{x}_1)$ of $\mathbf{X}_1$, where $E[d(\mathbf{X}_1, \mathbf{X}_2)] \leq D^A$, i.e. the difference between the two vectors is defined by some value of legitimate variation D^A, one should expect $H(K_H, \mathbf{X}_1) = H(K_H, \mathbf{X}_2)$. Additionally, the hash should be secure in the sense that having the host data $\mathbf{X}$, the attacker cannot generate a hash without the knowledge of the secure key K_H.

Definition 6: An authenticator is defined as a binary decision $\{0, 1\}$ based on the mapping:

$$\eta : \tilde{\mathcal{M}} \times \tilde{\mathcal{M}} \rightarrow \{0, 1\}. \tag{9}$$

The authentication amounts to select one of the two hypothesis $\{H_0, H_1\}$ based on the binary representations of the hash computed from the observed data $\mathbf{V}$, namely $\hat{\hat{M}} \equiv \hat{\hat{\mathbf{B}}}$, and the decoded message $\hat{M} \equiv \hat{\mathbf{B}}$. The binary decision $\{0, 1\}$ is taken by comparison of the number of different bits with respect to a predefined threshold.

If $\mathbf{X}$ is a finite alphabet stochastic process that satisfies the asymptotic equipartition property (AEP),[7] then there is a hashing-data-hiding code with specified probability of authentication error, if the rate of the hashing code R_H satisfies $R_H \leq R_{DH} < C$.

3. GEL'FAND-PINSKER TEXT DATA-HIDING

Four major groups of methods for text data-hiding have appeared in literature in the last 15 years: *syntactic methods*,[8, 9] where the diction or structure of sentences is transformed without significantly altering their meaning; *semantic methods*,[8, 9] where words are replaced by their synonyms and/or sentences are transformed via suppression or inclusion of noun phrase coreferences; *open space methods*,[8, 10] where either the inter-line space, the inter-word space or the inter-character space is modulated; and *character feature methods*,[10–12] where features such as shape, size or location are manipulated.

Syntactic and semantic methods are not suitable for all types of documents (e.g. contracts, identity documents, literary texts) and need, usually, human supervision. Some open space methods such as inter-line space modulation and inter-word space modulation can be automated, are robust against printing and scanning, but have low data storage capacity. On the other hand, inter-character space modulation and existing character feature methods have higher data storage capacities, but are less or not robust at all against printing and scanning. In this paper, we will not consider syntactic or semantic methods.

We explain now how the text data-hiding problem can be considered as a particular instance of the Gel'fand-Pinsker problem.[3, 4] The text, where some message m is to be hidden, is represented by $\mathbf{X}$ and called cover text. Each component X_n, $n = 1, 2, \ldots, N$, of $\mathbf{X}$ represents one character from this text. Here, a character is defined as an element from a given alphabet (for instance, the latin alphabet {A, B, ..., Z}). This alphabet can also contain punctuation characters as well as other special characters. To be more precise, each character X_n should be conceived as a data structure consisting of multiple *quantifiable* component fields (features): *shape* (geometric definition), *location*, *orientation*, *size*, *color*, etc.

We illustrate this approach by considering the family of quantization based methods, namely Scalar Costa Scheme[13] (SCS) and Quantization Index Modulation[14] (QIM).

In SCS the auxiliary random variable U used in (7) is approximated by:

$$U = W + \alpha' X = \alpha' Q_m(X),$$

where $Q_m(\cdot)$ is a scalar quantizer for the message $m \in \mathcal{M}$ and α' is a compensation parameter. This amounts to define the watermark as $W = U - \alpha' X = \alpha' Q_m(X) - \alpha' X$. The resulting stego text is obtained as:

$$Y = W + X = \alpha' Q_m(X) + (1 - \alpha')X. \tag{10}$$

We show in Figure 2 the corresponding SCS codebook and an illustration of how to use it for text data-hiding.

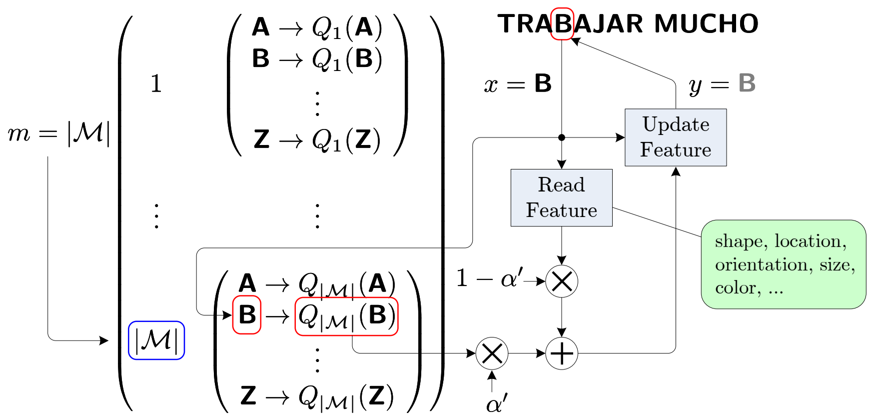


Figure 2. SCS codebook for text data-hiding ($N = 1$).

Distortion-Compensated QIM (DC-QIM) can be seen as a generalization of SCS to the vector case, where N-dimensional vector quantizers are used instead of scalar ones. In this scheme, which is also well suited for text data-hiding, we have:

$$\mathbf{U} = \mathbf{W} + \alpha'\mathbf{X} = \alpha'\mathbf{Q}_m(\mathbf{X}),$$

and the stego text is given by:

$$\mathbf{Y} = \mathbf{W} + \mathbf{X} = \alpha'\mathbf{Q}_m(\mathbf{X}) + (1-\alpha')\mathbf{X}. \tag{11}$$

In this work, we fix the decoder to be the minimum Euclidean distance decoder, defined by:

$$\hat{m}(\mathbf{V}) = \arg\min_{m\in\mathcal{M}} \|\mathbf{V} - \mathbf{Q}_m(\mathbf{V})\|,$$

where $\mathbf{V}$ is the noisy stego text.

In subsections 3.1 and 3.2, we give concrete examples on how to apply these methods to the problem of text data-hiding. In particular, we show that all previously proposed character feature methods,[10–12] including inter-character space modulation, can be considered as particular cases of the scheme described by (10), and that all previously proposed open space methods[8, 10] modifying either inter-word or inter-line space are all particular cases of the more general scheme described by (11), where the exploited character feature is *location*.

3.1. Color Index Modulation

In this method,[4] the stego text is obtained via (10), for $\alpha' = 1$ and defining the character feature X to be *color*. The main idea of this method is to quantize the color of each character in such a manner that the human visual system is not able to distinguish between the original and quantized characters, but it is still possible for a specialized reader, e.g. a high dynamic range scanner in the case of printed documents. It should also be mentioned that when halftoning is used as printing technology, halftone patterns are used to represent colors. Since the choice of these halftone patterns is not unique, i.e. there are various halftone patterns reproducing the same color, it is possible to exploit this characteristic to increase the data embedding rate.

An example illustrating this method is shown in Fig 3. Therein, dark characters encode a 0, whereas light ones encode a 1. Thus, a binary sequence can be sequentially embedded into the cover text.

Figure 3. Color Index Modulation: (a) original text; (b) marked text.

3.2. Location Index Modulation

In this method, the stego text is obtained via (10), for $\alpha' = 1$ and defining the character feature X to be *location*. The spatial location of each character is defined with respect to a two-dimensional (2D) orthogonal coordinate system, i.e. $X = (X^h, X^v)$. Here, we assume that the same coordinate system can be used for locating the original character, the stego character, and the noisy stego character.

In its simplest form, this method quantizes either the horizontal coordinate X^h (also known as inter-character space modulation[15]) or the vertical coordinate X^v. However, it is also possible to quantize both coordinates at the same time. In this case, one should think of $Q_m(\cdot)$ in (10) as a 2D vector quantizer acting on both horizontal and vertical coordinates. We show in Figure 4 some examples of quantizers for these scalar quantization schemes.

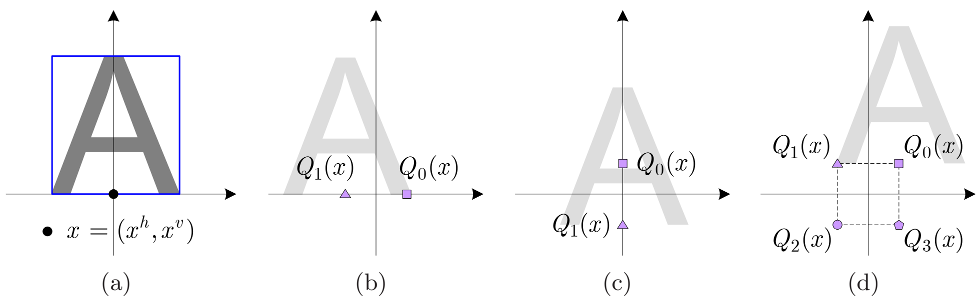


Figure 4. Location Index Modulation ($N = 1$): (a) original character $x = \mathsf{A} = (x^h, x^v)$; (b) marked character $Q_1(x)$ by horizontal shifting; (c) marked character $Q_1(x)$ by vertical shifting; (d) marked character $Q_0(x)$ by combined horizontal and vertical shifting.

It is possible to generalize this method to consider groups of characters, such as words or lines of text, instead of individual characters. Such a generalization is based on (11), where we take $\alpha' = 1$ and the feature vector $\mathbf{X}$ is obtained by concatenation of the individual component character locations $X_n = (X_n^h, X_n^v)$, $n = 1, 2, \ldots, N$. In this case, one has many degrees of freedom for designing the vector quantizer $\mathbf{Q}_m(\cdot)$. For example, if $\mathbf{X}$ represents a word, then one could design $\mathbf{Q}_m(\cdot)$ in such a way that all characters of $\mathbf{X}$ are shifted horizontally by the same amount and in the same direction. This technique is also known as word-shift coding.[10] Similarly, another well-known technique, namely line-shift coding,[10] is a special case of the described vector quantization method, where $\mathbf{X}$ represents a line of text and $\mathbf{Q}_m(\cdot)$ is such that all characters of $\mathbf{X}$ are shifted vertically by the same amount and in the same direction. We show in Figure 5 a 2D representation* of possible quantizers for these vector quantization schemes.

3.3. Hybrid Schemes

A natural extension of the schemes described by (10) and (3) is to consider simultaneously *multiple* character features instead of a single one. In fact, this idea was already introduced in the previous subsection while describing LIM. Indeed, one can consider the horizontal coordinate of a character as its first feature and the vertical coordinate as its second feature. Notice that these two features are independent from each other. Clearly, the main advantage of combining multiple independent features is the higher data storage capacity of the resulting scheme. For example, schemes (b) and (c) in Figure 4 have a maximum data embedding rate of 1 bit/character, whereas scheme (d) in the same figure has a maximum data embedding rate of 2 bits/character. Another example of a (true) hybrid scheme with even higher data storage capacity (maximum 3 bits/character) is the one that combines the CIM scheme shown in Figure 3 with the LIM scheme shown in Figure 4(d).

Hybrid schemes may also have other advantages due to the fact that different features have different properties. For example, it is known that CIM is less robust to photocopying than LIM; therefore, one can build a hybrid scheme capable of authenticating the contents of a text document (based on LIM and robust text hashing), and capable of discerning the original text document from its copies (based on CIM).

*An exact graphical representation is not possible since one would need to represent N-dimensional vectors $\mathbf{X} = (X_1, X_2, \ldots, X_N)$, where each component is a complex number (the real part representing the horizontal coordinate and the imaginary part representing the vertical coordinate).

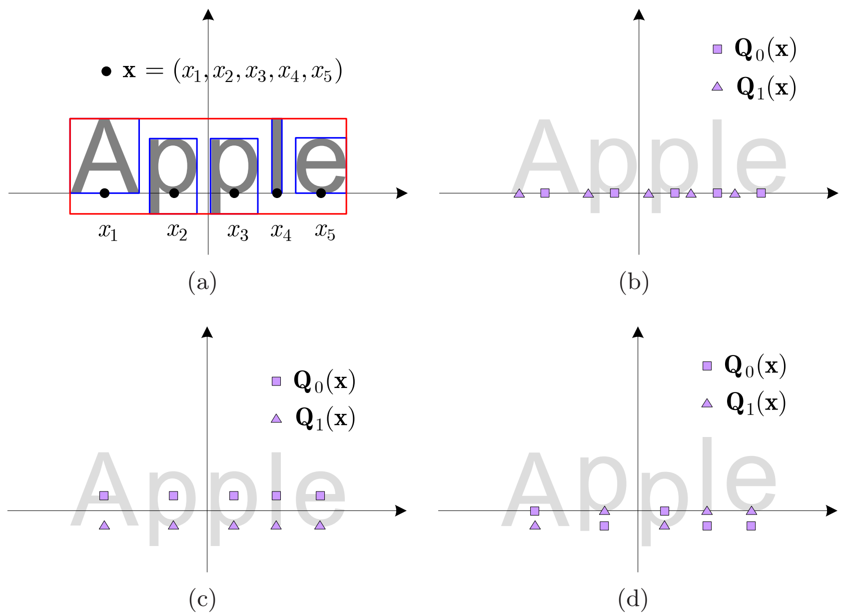


Figure 5. Location Index Modulation ($N = 5$): (a) original word $\mathbf{x} = (\mathsf{A}, \mathsf{p}, \mathsf{p}, \mathsf{l}, \mathsf{e}) = (x_1, x_2, x_3, x_4, x_5)$, where each $x_n = (x_n^h, x_n^v)$; (b) marked word $\mathbf{Q}_0(\mathbf{x})$ by horizontal shifting (a.k.a. word-shift coding); (c) marked word $\mathbf{Q}_1(\mathbf{x})$ by vertical shifting; (d) marked word $\mathbf{Q}_1(\mathbf{x})$ by mixed horizontal and vertical character shifting.

3.4. Error Control Coding (ECC) for Print-and-Scan Channels

The quantization-based methods explained above, which are by themselves channel codes, may not be completely robust to one-time or many-times printing and scanning, i.e. the decoder's output may contain some errors. In fact, it can be verified by experimentation that there is a trade-off between the invisibility of the watermark and the decoding accuracy. Typical examples of this problem relate to the use of printers, scanners, copy machines, fax machines, etc.

In order to reduce the error rate to an acceptable level, an outer layer of coding can be used. The correct design of such an outer layer takes into account the *equivalent channel* formed by the quantization encoder, the concatenation of the employed print-and-scan channels, and the quantization decoder. Moreover, the operation of the overall decoding machinery may require some modifications in order to get full benefit of soft-decision decoding techniques. For example, the quantization decoder may be modified so that it outputs soft estimates rather than hard estimates. Notice that the equivalent channel is, in general, different according to the used quantization technique (e.g. CIM, LIM, etc.). Accurate modeling of this channel for a particular quantization technique is challenging and there exist only few works[16–18] tackling this problem.

Finally, notice that the use of an ECC scheme together with a text data-hiding scheme decreases the data storage rate of the overall scheme. For example, if the text data-hiding rate is 2 bits/character and the rate of the ECC scheme is 1/2, then the effective data embedding rate is 1 bit/character.

4. ROBUST HASHING OF TEXT DOCUMENTS

A robust text hashing function H takes as input a secret key K_H and a text object $\mathbf{X}$ to give the hash value $\mathbf{H} = H(k_H, \mathbf{x})$. The text object could be either a character, a word, a sentence, a paragraph, a line of text, a text fragment, or even the whole text document. The hash value $\mathbf{H}$ is required to be invariant under unintentional/intentional legitimate modifications of the text document such as conversion between electronic formats, data-hiding, and typical handling operations that include printing, scanning, photocopying, faxing, etc.

We will consider two types of text hashing techniques. One attractive feature of these techniques is that they are compatible with character feature text data-hiding methods such as CIM and LIM.

4.1. OCR + MAC Text Hashing

This text hashing technique is based on OCR and a classical cryptographic MAC. The main idea is to apply OCR to the text document in order to obtain its ASCII representation; and then, using the secret key K_H, to compute the MAC of this representation in order to obtain the desired hash value. As it will be shown in Section 5, the use of OCR provides good robustness against legitimate modifications. However, since this technique highly relies on the accuracy of the employed OCR tool, it completely fails when OCR makes a mistake (this is because classical cryptographic MACs for similar inputs are generally quite dissimilar).

4.2. Random Tiling Text Hashing

Inspired by original work for images,[19] we describe in the following paragraphs a new robust text hashing algorithm. Let $\mathbf{x} = (x_1, x_2, \ldots, x_L)$ represent an input text object and k_H be a secret key. We suppose that $\mathbf{x}$ comes either from an electronic support (vector graphics representation of a text document) or from a scanned image of a printed text document. In our algorithm, we use k_H as seed of the random number generator used in all steps requiring random quantities. The following algorithm produces the hash value for a single text object.

1. Preprocess the image containing the text object so as to correct a possible skew.
2. Segment and convert the image into a bitmap composed of only black and white pixels. We suppose that the text object $\mathbf{x}$ belongs to a well-defined region $\mathcal{R}$ in this image. For simplicity, we fix the shape of $\mathcal{R}$ to be a rectangle (see Figure 6).
3. Generate at random P rectangles $\mathcal{R}_p = \{(i,j) : 1 \le i \le I_p, 1 \le j \le J_p\}$, where I_p and J_p are, respectively, the height and width in pixels of the p-th rectangle $\mathcal{R}_p$, $p = 1, 2, \ldots, P$. We assume that each rectangle $\mathcal{R}_p$ is randomly positioned inside $\mathcal{R}$, i.e. $\mathcal{R}_p \subset \mathcal{R}$, and that $\bigcup_{p=1}^{P} \mathcal{R}_p = \mathcal{R}$. In Figure 6, we schematize the generated random rectangles for two kinds of text objects.
4. Compute
$$\mu_p = \frac{1}{|\mathcal{R}_p| \sum_{r \in \mathcal{R}_p} \lambda_{k_H}(r)} \sum_{r \in \mathcal{R}_p} l(r) \lambda_{k_H}(r),$$
for $p = 1, 2, \ldots, P$, where $l(r)$ is the luminance value (0 or 1) of the pixel located at $r = (r^h, r^v)$ and $\lambda_{k_H}(r) \in \{0, 1\}$ is a key-dependent weight for the same pixel. If $\lambda_{k_H}(r) = 1$ for all $r \in R_p$, then μ_p is simply the sample mean.
5. Compute the intermediate hash $\tilde{\mathbf{h}} = (\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_P)$ by randomly picking P thresholds $T_p(k) \in [0,1], p = 1, 2, \ldots, P$ and defining $\tilde{h}_p$ as:
$$\tilde{h}_p = \begin{cases} 0 & \text{if } \mu_p < T_p(k), \\ 1 & \text{if } \mu_p \ge T_p(k). \end{cases}$$
6. Produce the final hash value $\mathbf{h} = H(k_H, \mathbf{x})$ by randomly choosing an index set $\{p_1, p_2, \ldots, p_Q\} \subset \{1, 2, \ldots, P\}$, $Q \le P$, and letting $\mathbf{h} = (\tilde{h}_{p_1}, \ldots, \tilde{h}_{p_Q})$.

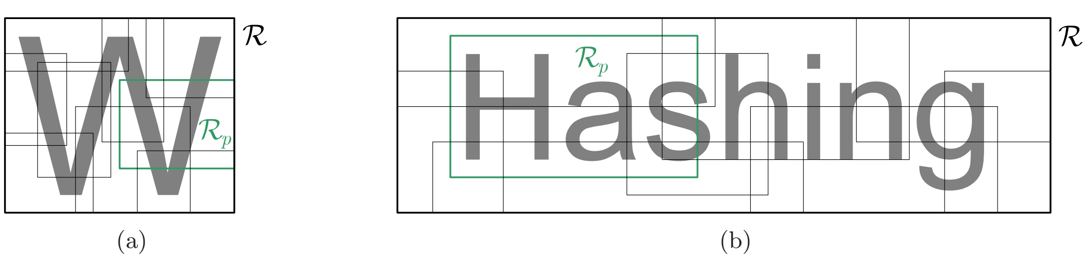


Figure 6. Random tiling text hashing: (a) character-based, $x = \mathsf{W}$; (b) word-based, $\mathbf{x} = (\mathsf{H}, \mathsf{a}, \mathsf{s}, \mathsf{h}, \mathsf{i}, \mathsf{n}, \mathsf{g})$.

5. EXPERIMENTAL RESULTS

In this part we present the experimental results for the two text hashing methods described in Section 4. We used standard office equipment (printer, scanner, copy machine, fax machine) to perform our tests. All of the digital images containing text objects were created/processed at 600 ppi.

For the implementation of OCR + MAC text hashing we used ABBYY FineReader as OCR tool and HMAC SHA-1 truncated to 80 bits as MAC. Assuming that a text line of 80 characters can reliably store 80 hash bits (see Section 3.4), this implementation takes into account the rate requirement of the text data-hiding part of the self-authentication system, namely that $R_H \leq R_{DH}$.

The following parameters were used for the implementation of random tiling text hashing: $P = 1024$, $\lambda_{k_H}(r) = 1$ for all $r \in \mathcal{R}_p$, and $Q = P$ (in fact Step 6 in Section 4.2 is replaced by $\mathbf{h} = \tilde{\mathbf{h}}$). The width and height (in pixels) of a random rectangle $\mathcal{R}_p$ were drawn uniformly at random from the intervals [5,10], [5,50], respectively. We do not claim, however, any optimality of the used parameters (e.g. the length of the hash is still relatively long in the current implementation).

The considered text objects were text lines (Arial font, 10 pt). A sample of five text objects is shown in Figure 7.

1 A text document authentication system aims at deciding whether a given text document is
2 authentic or not. The decision about authenticity is performed at the global level, meaning
3 that the system gives only a binary decision about the entire document: authentic or fake.
4 On the contrary, if a system makes decisions at the local level, we refer to it as a text
5 document tamper-proofing system. Thus, text document authentication and tamper-proofing

Figure 7. Sample text lines

Two classes of modifications were considered in the scope of this work: legitimate and illegitimate modifications. The legitimate modifications include electronic format conversion (Word $\leftrightarrow$ PostScript $\leftrightarrow$ PDF), printing and scanning, photocopying, and faxing. The illegitimate modifications (see Figure 8) include tampering of text lines by adding one new character, by suppressing one character, by replacing one character with a visually different one, and by replacing one character with a visually similar one.

Concerning the legitimate distortions, we show in Figures 9 and 10, the obtained results for OCR + MAC text hashing and random tiling text hashing, respectively. These figures should be interpreted as follows. Rows (or columns) from 1 to 5 represent the sample text lines of Figure 7. Rows (or columns) from 6 to 10, in this order, correspond also to the sample text lines 1 to 5 of Figure 7 but *after* a legitimate distortion has been introduced. We use the relative Hamming distance $d_H(\mathbf{h}_1, \mathbf{h}_2)$ to compare any two hash values $\mathbf{h}_1$ and $\mathbf{h}_2$. Recall that the desired properties for a robust text hashing method should be the following:

1. $d_H(\mathbf{h}_1, \mathbf{h}_2) \approx 0$ for any two similar text objects whose hashes are $\mathbf{h}_1$ and $\mathbf{h}_2$,
2. $d_H(\mathbf{h}_1, \mathbf{h}_2) \approx 0.5$ for any two dissimilar text objects whose hashes are $\mathbf{h}_1$ and $\mathbf{h}_2$.

We observe from Figures 9 and 10 that both text hashing methods show good effectiveness for the tested legitimate distortions. However, from Figure 9(d) we observe that when the OCR tool makes a mistake, e.g. a punctuation mistake, recognition of two spaces instead of one, etc., then the hash value of a text line is completely different from the one obtained from a similar text line.

Concerning the illegitimate distortions, we show in Figures 11 and 12, the obtained results for OCR + MAC text hashing and random tiling text hashing, respectively. These figures should be interpreted as follows. Rows (or columns) from 1 to 5 represent the sample text lines of Figure 7. Rows (or columns) from 6 to 10 correspond to the tampered text lines shown in Figure 8. We observe from Figures 11(a),(b),(c) and 12(a),(b),(c) that both text hashing methods show good effectiveness for the tested illegitimate distortions. From Figures 11(d) and 12(d), we observe that only OCR + MAC text hashing is able to handle the corresponding illegitimate distortion. Random tiling text hashing completely fails in this case. However, notice again, from Figure 11(d), that OCR + MAC text hashing can also fail if the OCR tool makes a mistake. In the case at hand, the OCR tool recognized the word aufhentication as authentication.

6 A text document authentication systemi aims at deciding whether a given text document is
7 authentic or not. The decision about authenticity is performed at the global levels, meaning
8 that then system gives only a binary decision about the entire document: authentic or fake.
9 On the contrary, if a system makes decisions at the local level, we refer to it as a texto
10 document tamper-proofing system. Thusj, text document authentication and tamper-proofing

(a)

6 A text document autentication system aims at deciding whether a given text document is
7 authentic or not. The decision about authenticity s performed at the global level, meaning
8 that the system gives only a binary decision about the entire document: authentic o fake.
9 On the contary, if a system makes decisions at the local level, we refer to it as a text
10 document tamper-proofing system. Thus, text document authentication nd tamper-proofing

(b)

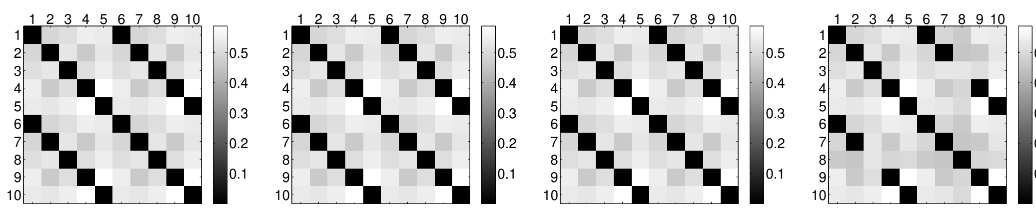
6 A text document authentication sysoem aims at deciding whether a given text document is
7 authentic or not. The decision about authenticity as performed at the global level, meaning
8 that the sistem gives only a binary decision about the entire document: authentic or fake.
9 On the contrary, if a system makes decisions at the vocal level, we refer to it as a text
10 document tamper-proofing system. Thus, text document authentication and tanper-proofing

(c)

6 A text document authentication system atms at deciding whether a given text document is
7 authentic or not. The docision about authenticity is performed at the global level, meaning
8 that the system gives only a binary decision about the entire dooument: authentic or fake.
9 On the cohtrary, if a system makes decisions at the local level, we refer to it as a text
10 document tamper-proofing system. Thus, text document aufhentication and tamper-proofing

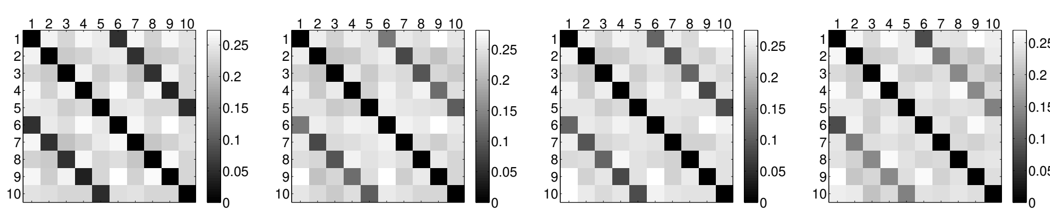
(d)

Figure 8. Sample of tampered text lines: (a) by adding one new character; (c) by suppressing one character; (c) by replacing one character with a visually different one; (d) by replacing one character with a visually similar one.



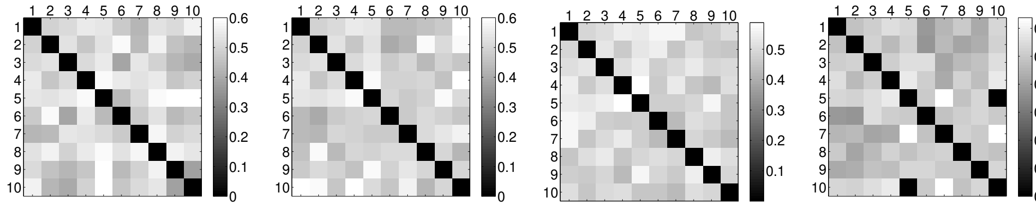
(a) (b) (c) (d)

Figure 9. Legitimate distortions for OCR + MAC text hashing: (a) electronic format conversion; (b) printing and scanning; (c) photocopying; (d) faxing.



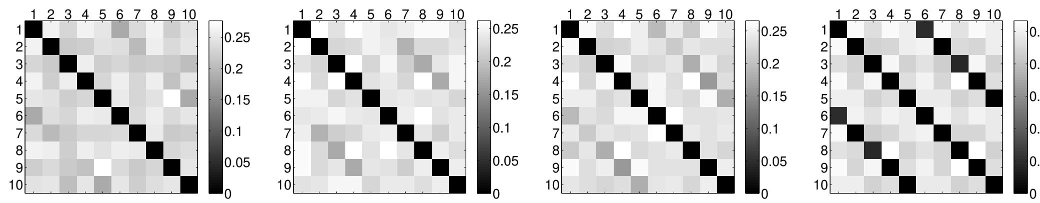
(a) (b) (c) (d)

Figure 10. Legitimate distortions for random tiling text hashing: (a) electronic format conversion; (b) printing and scanning; (c) photocopying; (d) faxing.



(a) (b) (c) (d)

Figure 11. OCR + MAC text hashing: (a) addition of one new character; (c) suppression of one character; (c) replacement of one character by a visually different one; (d) replacement of one character by a visually similar one.



(a) (b) (c) (d)

Figure 12. Random tiling text hashing: (a) addition of one new character; (c) suppression of one character; (c) replacement of one character by a visually different one; (d) replacement of one character by a visually similar one.

6. CONCLUSIONS

In this paper, we dealt with the problem of authentication and tamper-proofing of electronic and printed text documents by considering the combination of robust text hashing and text data-hiding technologies. Firstly, we addressed the problem of limited data storage capacity of current text data-hiding methods by considering their combination in the scope of the Gel'fand-Pinsker text data-hiding framework. Secondly, we studied two text hashing algorithms, namely OCR + MAC text hashing and random tiling text hashing, that are particularly well suited for the considered problem. In particular, we showed by experimentation that OCR + MAC text hashing shows better applicability than random tiling text hashing. However, we have also observed that the OCR + MAC text hashing method highly relies on the accuracy of the OCR tool. Moreover, the experimental work also confirms that both text hashing algorithms are robust against typical legitimate document distortions that include electronic format conversion, printing, scanning, photocopying, and faxing. Countermeasures for the weaknesses and a security analysis of both text hashing methods are left for future research work.

ACKNOWLEDGMENTS

This paper was partially supported by the Swiss National Science Foundation (SNF) professorship grant no. PP002-68653/1, the SNF project 200021-111643/1, the Interactive Multimodal Information Management (IM2) project, and the European Commission through the IST program under contract IST-2002-507932 ECRYPT and the sixth framework program under the number FP6-507609 SIMILAR. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

1. J. Fridrich, "Visual Hash For Oblivious Watermarking," in *Proceedings of SPIE Photonic West Electronic Imaging 2000, Security and Watermarking of Multimedia Contents*, **3971**, pp. 286–294, (San Jose, USA), Jan. 24-26 2000.
2. S. Voloshynovskiy, O. Koval, R. Villán, E. Topak, J. E. Vila-Forcén, F. Deguillaume, Y. Rytsar, and T. Pun, "Information-Theoretic Analysis of Electronic and Printed Document Authentication," in *Proceedings of SPIE-IS&T Electronic Imaging 2006, Security, Steganography, and Watermarking of Multimedia Contents VIII*, (San Jose, USA), Jan. 15–19 2006.

3. R. Villán, S. Voloshynovskiy, F. Deguillaume, Y. Rytsar, O. Koval, E. Topak, E. Rivera, and T. Pun, "A Theoretical Framework for Data-Hiding in Digital and Printed Text Documents," in *Proceedings of 9th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security*, **LNCS 3677**, pp. 280–281, (Salzburg, Austria), Sep. 19–21 2005.
4. R. Villán, S. Voloshynovskiy, O. Koval, J. E. Vila-Forcén, E. Topak, F. Deguillaume, Y. Rytsar, and T. Pun, "Text Data-Hiding for Digital and Printed Documents: Theoretical and Practical Considerations," in *Proceedings of SPIE-IS&T Electronic Imaging 2006, Security, Steganography, and Watermarking of Multimedia Contents VIII*, (San Jose, USA), Jan. 15–19 2006.
5. S. Gel'fand and M. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory* **9**(1), pp. 19–31, 1980.
6. D. R. Stinson, *Cryptography, Theory and Practice*, CRC, 2 ed., 2002.
7. T. Cover and J. Thomas, *Elements of Information Theory*, Wiley and Sons, New York, 1991.
8. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal* **35**(Nos 3&4), pp. 313–336, 1996.
9. M. Topkara, C. Taskiran, and E. J. Delp, "Natural Language Watermarking," in *Proceedings of SPIE-IS&T Electronic Imaging 2005, Security, Steganography, and Watermarking of Multimedia Contents VII*, (San Jose, USA), Jan. 17–21 2005.
10. J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for electronic distribution of text documents," *Proceedings of the IEEE (USA)* **87**(7), pp. 1181–1196, 1999.
11. A. K. Bhattacharjya and H. Ancin, "Data Embedding in Text for a Copier System," in *Proceedings of the ICIP*, **2**, pp. 245–249, 1999.
12. Q. Mei, E. K. Wong, and N. Memon, "Data hiding in binary text documents," in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, **4314**, pp. 369–375, Aug. 2001.
13. J. J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. on Signal Processing* **51**, pp. 1003–1019, Apr. 2003.
14. B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory* **47**, pp. 1423–1443, 2001.
15. N. Chotikakamthorn, "Electronic Document Data Hiding Technique Using Inter-Character Space," in *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 419–422, Nov. 24–27 1998.
16. R. Villán, S. Voloshynovskiy, O. Koval, and T. Pun, "Multilevel 2D Bar Codes: Towards High Capacity Storage Modules for Multimedia Security and Management," in *Proceedings of SPIE-IS&T Electronic Imaging 2005, Security, Steganography, and Watermarking of Multimedia Contents VII*, **5681**, pp. 453–464, (San Jose, USA), Jan. 16–20 2005.
17. S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document Identification for Copyright Protection Using Centroid Detection," *IEEE Transactions on Communications* **46**(3), pp. 372–383, 1998.
18. A. Malvido, F. Pérez-González, and A. Cousi, "A Novel Model for the Print-and-Capture Channel in 2D Bar Codes," in *International Workshop on Multimedia Content Representation, Classiffication and Security*, **LNCS 4105**, pp. 627–634, Springer-Verlag Heidelberg, September 2006.
19. M. K. Mihcak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," in *DRM '01: Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management*, pp. 13–21, Springer-Verlag, (London, UK), 2002.