ISSN 2278-3091

Volume 11, No.4, July – August 2022

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse041142022.pdf

https://doi.org/10.30534/ijatcse/2022/041142022

Electronic Document Securisation based on Document Structure

Mikaël Ange Mousse
Institut Universitaire de Technologie
Université de Parakou, Parakou, Bénin
mikael.mousse@univ-parakou.bj

Received Date : June 15, 2022 Accepted Date : July 19, 2022 Published Date : August 06, 2022

ABSTRACT

Securing electronic documents is an important task. It secures the identity of a document. In this sense, methods (mainly cryptographic methods) have been put in place to guarantee the secrecy of the meaning of messages as well as the signing of documents from certificates. The digital document is only defined by its textual content. With the rise of OCRs, the alteration and fraudulent use of documents has increased. In this work we propose an algorithm based on the use of a code book to secure the identity of the document. The codebook stores information that uniquely identifies the document. The codebook is used to make a correspondence table between the elements that we consider important to take into account in the modeling of a document and the different values that are associated with these elements. The results of our algorithm show that it is quite efficient.

Key words: computer security, document securisation, electronic document, integrity.

1. INTRODUCTION

It has been at least two decades since the expression digital document was born and integrated into IT departments. Very few of the managers were interested in this object but were rather concerned about data management. Generally speaking, information security is a process that aims to protect data or data of an organization being of fundamental importance, can be in the form of documents. The observation is that the documents related to the life of businesses and citizens are in digital form for the most part. The digital form of documents can be viewed or changed at the same time at the same time in different places. As a result, the questions that have arisen are only of a technical nature. They are linked to the probative value of digital documents, the preservation of human rights and the traceability of exchanges. They will obviously appear as an appropriate response from science and technology within a normative framework essential to ensure exchanges and accepted by all. Indeed, nowadays the falsification or modification of the digital document using software has become a reality and is becoming much more widespread. Protecting documents is therefore essential. Several techniques exist to achieve this end. These techniques are classified into three main categories:

- steganography which consists of concealing information;
- cryptography, which protects information during transmission;
- the digital signature which makes it possible to guarantee the integrity of a digital document and to authenticate its author.

All of these are essential basic tools for securing the content of documents.

Similar to the work conducted by Eskenazi et al. [10], the general objective of this work is therefore to propose a new tool allowing the authentication, the integrity of the content of a digital document whatever its form by means of the calculation of a robust and compact signature. in order to fight against fraud, falsification and malicious modification of documents. This signature will be based on the content (textual and graphic) of the document and will also take into consideration the internal structure underlying the basic elements making up this document. Thanks to a hash of the information of the document during the calculation of this signature, no information of the original document can be deduced from its sole signature. The signature can then be inserted into the document or used in content management software in the company to verify the authenticity of the document.

The history of document security is based on cryptography which has the function of hiding information and ensuring the authenticity of transmitted information. Document security algorithms can be classified into five families.

1.1 Watermarking

Early watermarking work was driven by copyright issues in an open digital environment. The duplication without loss of quality and the speed of distribution in an environment such as the Internet meant that any digital work (image, film, music, software, etc.) could be copied and distributed extremely easily without control by the rights holders. One of the first ideas to ensure the protection of works was to use cryptographic techniques: a work is offered encrypted, and users can buy a decryption key to view the original work [1]. This idea is the basis of the broadcast of encrypted channels, for example. However, this method clearly shows its limits: once the user has the work in clear, nothing prevents him from copying it and redistributing or reselling it. An intrinsic protection mechanism for the unencrypted work therefore quickly appeared essential. Watermarking makes it possible to extend the protection of works: by giving it an invisible and persistent "signature", it becomes possible to automatically trace its use in a network [1]. Alternatively, we can insert by marking, an identifier of the acquirer in order to make him responsible and to dissuade him from letting the piracy take place through negligence or with his tacit consent.

A distinction is made between watermarking adapted to the content and digital watermarking. Content-aware watermarking methods use specific characteristics of the document. These are still extractable after attack, since this must not affect the semantics of the document. Digital watermarking, also called Watermarking, was one of the solutions to reinforce the security of multimedia documents. The main idea is to hide subliminal information in a document to provide a security or informational service. The particularity of this technique compared to a simple storage of information in the header of the file is that the brand is intimately linked and resistant to the data. Thus, the watermark is independent of the file format and it can be detected or extracted even if the document has undergone modifications or if it is incomplete [1].

1.2 Digital Rights Management

It is a technique is a technical protection measure to control access to digital works (music, video/film, book, video game, software in general, etc.). In the past, digital rights management algorithms have been used in the sale and distribution of music [2]. These methods are based on a principle of cryptography in order to protect any digital content from unauthorized use. The work is encrypted to make it unreadable. Obtaining a decryption key then allows full access to readable content [2]. This process involves 3 actors:

1. The distributor who provides encrypted content;
2. The license server;
3. The user's reading tools (reader type medium, tablet, and its associated reading application).

1.3 Streaming

Streaming allows content to be protected because it is based on the idea that the user cannot copy a work if he does not have it in his possession. And without a personal copy, no illegal distribution possible, the reasoning is simple. The security provided therefore seems quite high since the user does not have his own copy of the book, he only accesses a server to view it and he has no means of downloading the content of the file [3]. The limit of this type of service lies in the constant need for a connection to access online content. In this case, with the offers as we currently know them, it is not possible to read content in a disconnected context [3].

1.4 Steganography

Steganography is a technique that comes to the rescue of tattooing. This technique aims to conceal data in documents [1]. The main purpose of this technique is to allow the signing of documents (to be able to put copyrights on them) but it can be used for other things (more or less legal). One of the proven methods is invisible ink. It is heard of in the Arabic scriptures and was widely used by students in the Middle Ages. This ink is then made from onion juice and ammonia chloride. The writing is then made visible thanks to a source of heat (like a flame for example). It relies on the fact that the information to be hidden is mixed with banal information in such a way as to go unnoticed. Information is hidden in text or image files. One of the methods is the substitution of the least significant bits. In this method, support data with high redundancy, for example image or sound, are used. This type of signal is made up of a set of samples (pixels or audio samples), each sample representing the amplitude of the signal at a given time or place. A small variation of the signal being generally imperceptible, the hidden message can therefore be transmitted by slightly modifying the amplitude of the samples [1]. In practice, the least significant bit of each sample is replaced by one of the bits of the message to be transmitted.

1.5 Hash functions

The principle is that a clear message of any length must be transformed into a fixed message of reduced length called hashed or condensed, less than the initial one. The interest is to use this hash as a fingerprint of the original message so that the latter is uniquely identified [4]. The hash does not contain enough information on its own to allow reconstruction of the original text. The objective is to be representative of a particular and well-defined piece of data (in this case the message). Hash functions have many properties [4]:

- they can apply to any message length M;
- they produce a result of constant length;
- it must be easy to calculate h = H(M) for any message M;
- for a given h, it is impossible to find x such that H(x) = h.
- for a given x, it is impossible to find y such that H(y) = H(x)
- it is impossible to find x, y such that H(y) = H(x).

Recently, several approaches used the extraction of specific features to characterize a document. Then Mikkilineni et al. [11,12] used a gray level co-occurrence matrix and Shang et al. [13] used noise energy, contour roughness and average gradient of character edges to characterize printing document. These approaches cannot be used for document authentication. Tkachenko et al. [14] introduced a two-level QR code for private message sharing and document authentication. This code is added to the document and it is used to detect unauthorized duplication of the document. Another research works introduce some approaches which consist to separate the document into primary elements such as images [15], text [16], layout [10], table [17]. These elements are used to check the integrity pf the document. Editing the document (paying attention to the specified elements for integrity checking) represents a major shortcoming of its approaches.

This work introduces a robust strategy for document integrity checking. We propose a codebook that integrates a lot of document-specific information.

2. PROPOSED APPROACH

To obtain a robust signature, we propose a system that have three modules.

1. Character recognition
2. Features extraction
3. Implementation of the signature

The first step is used to convert a document to text. This first step is important since it allows the computer to understand the content of the document. The second step is the extraction of features which will be used to performed the signature of the document. These features include information that can uniquely identify a document. After the extraction of the features, we suggest to build a codebook. This codebook is used to prove the integrity of the document.

2.1 Character recognition

The first module, called character recognition, is a document-to-text conversion module. This module is produced using an optical character reader. Document segmentation leads to the decomposition of the document into structural units such as textual regions or graphics. A bad application of the segmentation method leads to errors [5]. OCR performance measurement can continue on the quality of the physical segmentation of the document into regions. This operation is in fact decisive for converting the text back, essentially in the case where the structure of the document is multi-column, includes tables and graphics. An error in the segmentation of the regions can distort the order of their reading [5].

In our work we used TESSERACT. Tesseract is an open-source OCR engine that was developed at HP [8]. It is equipped with a convolutional neural network (algorithm based on the LSTM) which allows it to perform the operation for which it is proposed. To perform character recognition, the TESSERACT algorithm searches for lines. The line search algorithm is designed so that the page can be recognized without having to straighten the characters. This saves time and memory. This strategy also helps to combat the loss of image quality. The key elements of the process are blob filtering and line building [9]. Assuming the layout analysis has already provided text regions of approximately uniform text size, a filter removes drop caps and vertically touching characters. The median height approximates the size of the text in the region, so it's safe to filter out blobs that are smaller than a fraction of the median height, most likely being punctuation, diacritics, and noise. Filtered blobs are more likely to match a pattern of non-overlapping, parallel, but angled lines. Sorting and processing blobs by abscissa makes it possible to assign blobs to a single line of text. Once the filtered blobs are obtained, a least median of squares adjustment is used to estimate the baselines, and the filtered blobs are reinserted into the appropriate rows. The last step in the line creation process merges the drops that overlap by at least half horizontally, putting diacritics with the correct base and correctly matching parts of certain broken characters.

2.2 Features extraction

Once we have tools allowing us to segment the work and read the content of the scanned document, we must find information that will allow us to uniquely characterize the document. To do this we decide to establish a code book. The code book makes it possible to make a correspondence table between the elements that we will consider important to take into account in the framework of the modeling of a document and the different values which are associated with these elements. In our work we decided to extract two groups of characteristics. The characteristics of the first group will allow us to describe the document globally, while the characteristics of the second group will place much more emphasis on the elements present on the pages of the document.

General characteristics

By considering each document, we proposed to extract elements whose descriptions are given by Table 1.

Table 1 : General characteristics

Features	Explication
d_nbr_pag	number of pages in the document
d_nbr_fig	number of images in the document
d_nbr_tab	Number of tables in the document
d_not_bpg	Footnote
d_nbr_pag	Number of paragraphs in the document
d_nbr_ped	list of pages on which the signature is based

We have used these elements to show the importance of these descriptions in a document. They have the greatest interest in document analysis and recognition. After the identification of the pages, we suggest the extraction of some pages-based features.

Pages features

As at the global level, it is important to characterize the pages of the document. For this, we propose the extraction of the following characteristics.

1. P_nbr_mot: number of words on the page;
2. P_nbr_lig: Number of lines on the page;
3. P_nbr_npg: page number;
4. P_nbr_car: Number of characters on the page
5. P_not_nbp: footnote of the page
6. P_nbr_par: Number of paragraphs on the page.

These elements show the role played by the selected pages. The set of these elements made up is the list of pages selected in the document. Which leads us to say that document recognition only uses visual information.

2.3 Implementation of the signature

From the global characteristics and the specific characteristics of the page obtained from section 3.2, we obtain a signature which is represented as follows: <cd_nbr_pag, cd_nbr_fig, cd_nbr_tab, cd_not_bpg, cd_nbr_pag, cd_nbr_ped>. In this signature, the values are obtained as follows:

- **cd_nbr_pag = d_nbr_pag;**
- **cd_nbr_fig = d_nbr_fig;**
- **cd_nbr_tab = d_nbr_tab;**
- **cd_not_bpg = d_not_bpg;**
- **cd_nbr_pag = d_nbr_pag** ;
- **cd_nbr_ped** corresponds to a list of n elements in the form of a code list of the characteristics extracted per page. The value n represents the number of pages used to build the signature. Each element is represented as <**P_nbr_mot, P_nbr_lig, P_nbr_npg, P_nbr_car, P_not_nbp, P_nbr_par**>.

Once the signature obtained, we also defined a method for the comparison of the signatures. First, the characteristic attributes of the document must be compared. When making this comparison, you must first base yourself on the overall characteristics. It is necessary to make the difference between the two (02) values by taking into account only the first 4 values. If the distance is greater than a value ß then we conclude that the two documents are different. Otherwise, the comparison is extended to the characteristics per page. If the difference between the two is greater than the defined threshold α, then the two documents are different. We compute an ecludiean distance in order to compare the key.

In the implementation, three parameters are important. The number of pages used to generate the signature, the α and ß values. These values are very important because they make it possible to make the proposed security algorithm much more efficient.

3. EXPERIMENTAL RESULTS AND DISCUSSION

3.1 Experimental setup

In order to validate our proposal, we carried out tests. This subsection describes the experimental protocol. The experiments were conducted in two aspects. This is due to the fact that the robustness of the system will depend on the performance of each of the two modules that make up our system (see Figure I). The first aspect concerns the character recognition module. For this, 150 documents of 225 pages each were digitized. These documents have the following characteristics:

- the pages of one (or more) document(s) contain text in French in a single column, the text comes from a typewriter and corresponds to a 12-point Courier font.
- the pages of one or more documents contain French text in two columns in Times 12-point font.
- the pages of a document(s) contain two-column English text in Times 12-point font.
- the pages of a document(s) contain text in two columns, one in English and the other in French, written in Times 10- and 12-point fonts.

The pages contain images, tables, graphs, but in places the characters are poorly printed. All these specifications are due to the fact that we want to ensure the robustness of the recognition system.

The system is deployed by using a laptop which have the following characteristics:

- Intel Core i7-5500U CPU 2.40 GHz;
- RAM memory: 16 GB;
- Hard disk: 1 TB;
- processor: i7 CPU;
- Frequency: 2.4GHz.

The scanning of documents is carried out using a scanner which has the following characteristics:

- pixel resolution: 512 dpi (middle x, y above scanning area);
- scan data: 8-bit grayscale (256 levels of gray);
- scan input area: 14.6 millimeters (nom. width at center) 18.1 millimeters (nom. length);
- compatibility: Microsoft Windows Seven, Linux.

The implementation is done using MATLAB. MATLAB is a numerical computing and programming platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models [7]. It is a fourth-generation programming language emulated by a development environment of the same name called MATLAB (matrix laboratory) developed by MathWorks. MATLAB can manipulate matrices, display curves and data, implement algorithms, create user interfaces, and can communicate with other languages such as C, C++, Java, and Fortran. MATLAB users come from very different backgrounds such as engineering, science and economics in both industrial and research contexts. Matlab can be used alone or with toolkits [7].

3.2 Results and discussion

The following section presents the results obtained by our system. These results are presented under two categories. First, we tested our implementation of the automatic character reading algorithm. The percentages of success are recorded in Table 2.

Table 2 : Character recognition rate

Class	Capture	Omnipage	TestBridge	TypeReader	Tesseract
ASCII number	91,40%	93,44%	96,61%	95,93%	**98,42%**
ASCII lowercase letters	98,07%	98,82%	99,15%	94,90%	**99,54%**
ASCII spaces	99,56%	99,46%	99,49%	97,34%	**99,59%**
ASCII special symbols	97,41%	97,08%	96,10%	89,93%	**95,48%**
ASCII uppercase letters	87,71%	95,73%	95,73%	93,70%	**98,82%**
Latin1 lowercase letters	0,00%	96,72%	97,54%	57,41%	**99,33%**

In Table 2, in addition to the good recognition rate obtained by the TESSERACT algorithm, we have also provided the good recognition rate values obtained by other optical character recognition applications. These are Capture, Omnipage, TestBridge and TypeReader software. These softwares are the most used in the character recognition community [6]. The good recognition value is obtained by dividing the number of characters well detected and the total number of characters submitted to the system. This table shows the competitiveness of the character recognition algorithm used. This is crucial since the robustness of the proposed system strongly depends on this module. Once past the recognition stage, we did the signature robustness test.

To perform this test, we proceeded as follows. Firstly, we extracted pages to obtain the key. These pages were chosen randomly. Once the keys have been extracted, we have modified a page of the document. The purpose of the manipulation is to see if the key allows the unique identification of the document. The page to be modified is generated by a random algorithm. For the same document, the experimentation process is repeated 500 times. Thus, for each document, 500 experiments were carried out with modifications on different pages. These iterations were performed with the optimal values for α and ß. Indeed, experiments have been conducted to identify the optimal values for each type of document. The values obtained are represented by the graph presented in Fig 1. These values represent the mean values obtained in each case.

According to Fig. II, our experiments started with at least 20% of the document page count. Below 20%, the values obtained are practically nil. From 50% of the total number of pages of the document, we start with satisfactory results. Beyond 80% of the total number of pages of the document, the percentage of good identification goes above 90%. Depending on the sensitivity of the document and the desired security rate, we choose the ideal number of pages to use in the design of the document signature. The last aspect on which we conducted our experiments was the key extraction time. Once the optical reading process has been completed, the key extraction algorithm needed an average of 0.387 seconds per page to extract the information to be used for the design of the signature.

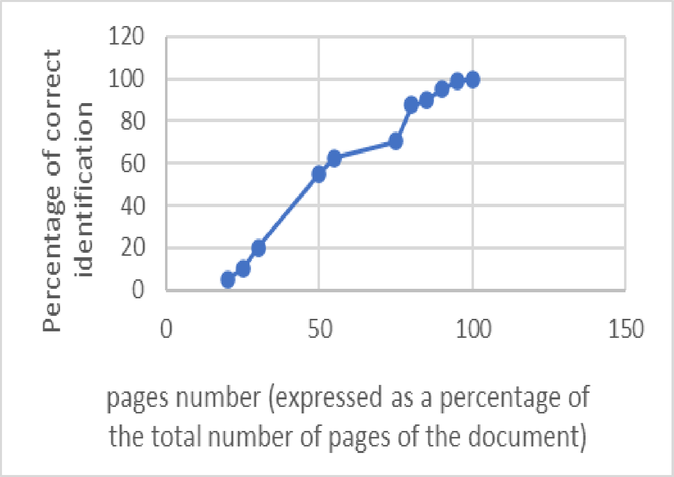

Figure 1: Percentage of correct identification

4. Conclusion

In this work, we have proposed an algorithm for identifying an electronic document. The goal is to ensure the integrity of the document. This algorithm uses elements obtained from the document. These elements are grouped into two categories. To obtain these elements, the proposed system uses two modules. The first module allows the reading of the characters of the electronic document. Once the reading is complete, the signature is obtained using an algorithm based on a codebook. The experiments carried out allowed us to demonstrate the effectiveness of the proposed strategy. These experiments allowed us to justify that the rate of good identification depends on the number of pages used. The results also allowed us to demonstrate that the solution does not need huge hardware resources before obtaining the results.

Our prospects are the implementation of a solution for securing multimedia content. This step will allow us to generalize our approach and find elements that can be adapted to the type of elements to be secured.

REFERENCES

1. P. Nguyen, S. Baudry, **Le tatouage de données audiovisuelles**, *Les Cahiers du numérique*, vol. 4, no. 3, pp. 135-165, 2003.
2. R. Stallman, "**S'opposer à la mégestion numérique des droits**", sur *gnu.org*.
3. M. Fansi, V. Lalanne, A. Gabillon, **Vers l'interopérabilité des Systèmes de DRM (Digital Rights Management)**, 2007.
4. R. Dumont , **Cryptographie et Sécurité informatique**, *Faculté des Sciences Appliquées*, Université de Liege, 2010.
5. A. Belaïd, L. Pierron, L. Najman, D. Reyren, **La numérisation de documents : principes et évaluation des performances**, *Bernard Hidoine; Jean-Claude Le Moal. Bibliothèques numériques*, La Bresse, ADBS, 35, 2000.
6. A. Belaïd, H. Cecotti, **Reconnaissance de caractères : évaluation des performances**, *Mullot, Rémy. Les documents écrits : de la numérisation à l'indexation par le contenu*, HERMES, 2006, Traité IC2, série informatique et systèmes d'information, 2006.
7. **Matlab description**, https://fr.mathworks.com/products/matlab.html
8. I. Marosi, **Industrial OCR approaches: architecture, algorithms and adaptation techniques**, *Document Recognition and Retrieval XIV*, SPIE, 6500-01, 2007.
9. R. Smith, **An overview of the Tesseract OCR Engine, Ninth international conference on document analysis and recognition** (*ICDAR 2007*). IEEE. p. 629-633, 2007.
10. S. Eskenazi, P. Gomez-Krämer, et J-M. Ogier, **The Delaunay document layout descriptor**, *in Proc. of the 15th ACM SIGWEB International Symposium on Document Engineering*, 2015, pp. 1–10, 2015.
11. A.K. Mikkilineni, N. Khanna, E.J. Delp, **Texture based attacks on intrinsic signature-based printer identification**. *In: Media Forensics and Security*. vol. 7541, p.75410T. International Society for Optics and Photonics, 2010.
12. A.K. Mikkilineni, N. Khanna, E.J. Delp, **Forensic printer detection using intrinsic signatures.** *In: Media Watermarking, Security, and Forensics*. vol. 7880, p. 78800R. International Society for Optics and Photonics, 2011.
13. S. Shang, N. Memon, X. Kong, **Detecting documents forged by printing and copying**. *EURASIP Journal on Advances in Signal Processing 2014*(1), 1–13, 2014
14. I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J.M. Gaudin, C. Guichard, **Two-level QR code for private message sharing and document authentication**. *IEEE Transactions on Information Forensics and Security* 11(3), 571–583, 2016.
15. S. Eskenazi, B. Bodin, P. Gomez-Krämer, J.M. Ogier, **A perceptual image hashing algorithm for hybrid document security**. *In: IAPR International Conference on Document Analysis and Recognition (ICDAR).* vol. 1, pp. 741–746. IEEE, 2017.
16. S. Eskenazi, P. Gomez-Krämer, J.M. Ogier, **A study of the factors influencing OCR stability for hybrid security**. In: *IAPR International Conference on Document Analysis and Recognition (ICDAR).* vol. 9, pp. 3–8. IEEE, 2017.
17. H. Alhéritière, F. Cloppet, C. Kurtz, J.M. Ogier, N. Vincent, **A document straight line based segmentation for complex layout extraction**. In: *IAPR International Conference on Document Analysis and Recognition (ICDAR).* vol. 1, pp. 1126–1131. IEEE, 2017.