Protocols for data-hiding based text document security and automatic processing

Frédéric Deguillaume Yuriy Rytsar Sviatoslav Voloshynovskiy Thierry Pun
Computer Science Department, CUI - University of Geneva,
24, rue du Général Dufour, CH-1211 Geneva 4, Switzerland
E-mail: {Frederic.Deguillaume, Yuriy.Rytsar, svolos, Thierry.Pun}@cui.unige.ch

Abstract

Text documents, in electronic and hardcopy forms, are and will probably remain the most widely used kind of content in our digital age. The goal of this paper is to overview protocols for text data-hiding based "smart documents", achieving document self-authentication, self-recovery, self-annotation and automatic processing. We argue that document security, recovery and embedded annotation are the most promising data-hiding based frameworks.

1. Introduction

Numerous watermarking or data-hiding methods have been developed for images, video, and audio, most of them targeting copyright protection and tracking applications [4, 8]. Today, protecting also textual content and hardcopy documents has clearly become an issue of the highest importance; specific schemes have been developed for text data-hiding. One popular class of text data-hiding consists of the modulation of graphical features of characters, lines, or words [1], etc. The second major class of algorithms consists of semantic or syntactic based approaches, which replace words or sentences by semantic equivalents or synonyms, or exploit punctuation ambiguities, like the scheme of Purdue university team [2]. Other approaches address mostly hardcopies by interacting with printer characteristics (like the halftoning process), or by adding features to an already printed paper in a two-stages action such as additional tiny invisible dots [6].

Beside this, there is today an increasing demand for document annotation and automatic processing, including textual content in both digital and printed forms, mostly for security related applications. Therefore, self-sufficient metadata embedded into the content as *hidden data* appears as an elegant solution for automatic document processing. The hidden data can then be resilient to format transcoding and digital/analog (D/A) conversion (oppositely to header-based annotations), allowing content-aware processing through any digital or analogue distribution channel. This leads to the concept of intelligent media or *smart document* containing all the necessary meta-data for automatic processing.

This paper discusses the protocols needed for data-hiding based text document security and automatic processing. Section 2 defines a generic text document data-hiding architecture suitable for these protocols. Then Section 3 discusses its robustness to media conversion. Section 4 presents a protocol for document identification, authentication, and integrity check with localization, which is robust against hidden-data copy; self-embedding based document recovery is also briefly introduced.

2. Text document data-hiding

Watermarking and data-hiding consist of embedding some information, called *message* and usually key-dependent, into a *host* content and in an imperceptible manner (*perceptual masking*). The described concepts below are based on *data-hiding*, meaning that message of a certain length (i.e. multibit message) is embedded, and eventually needs to be extracted and *decoded*; oppositely, the more restrictive concept of *watermarking* refers only to the *detection* of the fact that the content was marked with a given key.

2.1. Text data-hiding

We propose for the following to use graphical features modulation based text data-hiding scheme, which encodes the data into one or several features of individual characters or groups of characters, without changing the textual content itself. Examples of such features are character shifting, width, height, orientation, font, darkness, inter-character, inter-word or inter-line spacing, etc., or a combination of some of them. Modulation is kept small to ensure perceptual masking when the document is displayed or printed. If hardcopy documents are targeted, features and kind of modulation should be chosen in order to survive through printing (e.g. using inkjet or laser printers), as well as rescanning, at typical resolutions starting from 600 dpi. Although usual text documents have a two-dimensional (2D) visual structure, the hidden data consists in a one-dimensional

0-7803-9332-5/05/$20.00 ©2005 IEEE

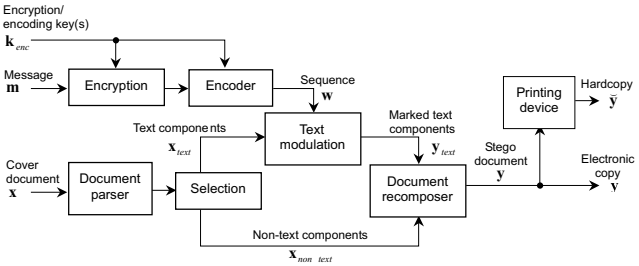

Figure 1: Protocol for document data-hiding: embedding.

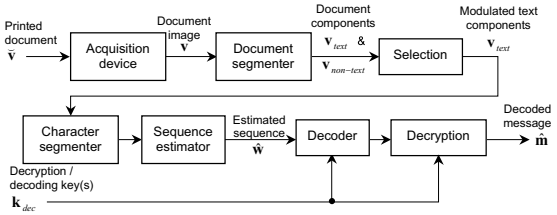

Figure 2: Protocol for document data-hiding: extraction.

(1D) sequence accordingly to the logical sequence of characters, lines, paragraphs, and pages. Graphic features modulation is the most suitable approach for the protocols below, first for its portability to any document representation, and secondly for its preservation on the textual content.

2.2. Document data-hiding protocol

The data-hiding protocol at the document level is illustrated in block-diagrams Figures 1 (embedding) and 2 (extraction and decoding). For the embedding stage (Figure 1), the message $\mathbf{m}$ is encrypted, encoded, potentially using error correcting codes (ECC), from encryption and encoding key(s) $\mathbf{k}_{enc}$. An auxiliary known pilot or *reference* sequence can also be interleaved with the message sequence to help the compensation of distortions at the decoding stage. We then obtain the sequence $\mathbf{w}$. The source (*cover*) document is available in *structured electronic* format, that is is represented in a character-oriented encoding as generated by edition/publication tools – like Microsoft Word (DOC), Adobe Acrobat (PDF), PostScript (PS), LaTeX, Rich Text Format (RTF), Hypertext Markup Language (HTML), eXtensible Markup Language (XML), etc. It is then parsed in order to isolate its text components $\mathbf{x}_{text}$ from the other components $\mathbf{x}_{non-text}$ (like graphics, images, or even invisible elements). Individual characters are selected in $\mathbf{x}_{text}$ in their logical order and their features are modulated accordingly to $\mathbf{w}$, resulting into the marked text components $\mathbf{y}_{text}$. Then the document is recomposed from marked $\mathbf{y}_{text}$ and non modified $\mathbf{x}_{non-text}$, keeping the original document structure and resulting into the marked (*stego*) document $\mathbf{y}$ in electronic format; the latter can be either used in electronic form, or printed as $\breve{\mathbf{y}}$.

For the extraction process (Figure 2), the possibly attacked and distorted hardcopy $\breve{\mathbf{v}}$ is acquired (scanned) as the document image $\mathbf{v}$. Then it is potentially prefiltered, and a *document image segmentation* takes place in order to extract its text components $\mathbf{v}_{text}$. State-of-the-art document segmentation techniques are described in [7]. Individual characters are segmented from $\mathbf{v}_{text}$ and the sequence of modulated features is estimated as $\hat{\mathbf{w}}$. If in structured electronic form, the document $\mathbf{v}$ is parsed accordingly to its format as for the embedding process. Then, after channel state compensation and resynchronization based on the reference sequence it contains, the sequence $\hat{\mathbf{w}}$ is decoded and decrypted using decoding/decryption key(s) $\mathbf{k}_{dec}$, resulting into the estimated message $\hat{\mathbf{m}}$.

3. Robustness to media conversion

A text data-hiding scheme is resilient to media conversion if the hidden data contained in the text "follows" the document in any of its representations, including D/A conversion, various structured electronic formats, and even resists lossy compression. This implies embedding and extraction methods resilient to format conversion, and a certain level of robustness with respect to the printing/rescanning channel and other distortions.

3.1. Robustness and D/A conversion

Similarly to image data-hiding, *robustness* refers to the reliable communication of the hidden data under various kinds of intentional and non-intentional distortions or *attacks*, like signal processing attacks (filtering, noise addition, etc.), or desynchronization (geometrical distortions). For copyright protection and tracking applications, it is essential that the hidden data can be detected and decoded even after severe attacks in order to efficiently trace unauthorized copying or use. However, making features modulation based text data-hiding robust against intentional text reformatting and regeneration is still today a challenge: modulated features can easily be stripped-off from electronic documents, and an optical character recognition (OCR) tool does the same for printed documents.

On the other hand, identification, authentication and integrity check applications do not require a highly robust data-hiding algorithm, since the removal of the embedded data allows us to detect a security problem; in that case, the goal of an attacker would be to generate or re-generate some hidden data which wrongly identifies or validates a faked document. By analogy with the image data-hiding schemes, we refer to the concepts of *fragile* and *semi-fragile* data-hiding [5], in the sense that if the hidden data is altered, then an unauthorized modification can be detected. A fragile scheme, intolerant to any modification, would address electronic documents only, while a semi-fragile system is robust against some non-intentional attacks – in particular D/A conversion in order to mark hardcopy docu-

ments. Authentication and integrity check protocol is discussed in Section 4. The same considerations stand for non-security applications where the hidden data only provides some added-value to the content (e.g. for document embedded annotation or automatic processing), since the removal of hidden data is here of no interest. The needed semi-fragility can be achieved for printed documents by the use of proper ECC as well as channel state compensation and resynchronization of $\mathbf{w}$ as mentioned previously.

3.2. Resilience to format conversion

Features, which are common to most existing electronic text document formats, should be choosen allowing the hidden data to be attached to the document when stored in the electronic form. This is usually not a problem for features (characters fonts, shifts, etc.) for many usual formats (DOC, PS, PDF, etc.) mentioned in Section 2 [1]. Since it is not practical to design a document parser for every different format, a solution is to implement a parser for one or two "generic" formats (such as DOC or PS), and to rely on the widely available format conversion facilities for other formats. Concerning hardcopies, one of the numerous existing document image segmentation algorithms can be used depending on the targeted application and on the apriori knowledge of the document layout [7]. The correct segmentation of individual characters is essential in order to estimate their features; however the recognition of characters themselves are not necessary, then no OCR is needed – an advantage in term of computational complexity.

3.3. Hidden data resynchronization

A first source of desynchronization, similarly to the image data-hiding, consists of geometrical distortions which typically occure for printed document when rescanned: slight rotation, or different resolutions resulting into different document image scales. Electronic versions however are not concerned by such geometrical distortions. A second source of desynchronization, more specific to text documents, can result from document or characters miss-segmentation: missed text areas, two characters wrongly taken as a single one, skipped characters, or inserted outliers. The result at the decoder consists of symbol deletions or insertions giving 1D shifts of parts of sequence $\hat{\mathbf{w}}$. Electronic versions can also be affected by insertion or deletion if modified using a text edition/publication tool. A two-levels resynchronization can compensate for these distortions at the decoder side: first, it is essential that both electronic format parsing and printed version segmentation result into the same components in the same logical order; orientation compensation (document *deskewing*) can be done using the natural properties of structured text with parallel lines, based on Fourier transform magnitude or Hough transform. Secondly, sequence $\hat{\mathbf{w}}$ resynchronization and reordering can be made using cross-correlation with the embedded reference sequence.

[1] Note that simple electronic ASCII text is more difficult to modulate, since it offers almost no manipulable graphic features – except the addition of extra space characters between words or lines.

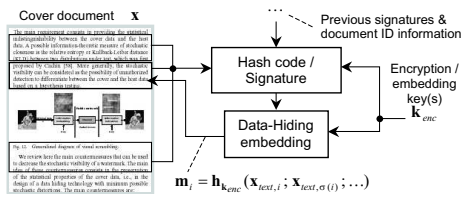

Figure 3: Authentication and tamper proofing: embedding.

4. Identification, authentication and tamper proofing

4.1. Document identification

Here is proposed a protocol for integrated document *identification*, *authentication*, and integrity check with localization or *tamper proofing*. Document identification (ID) can be achieved by embedding unique data which identifies the nature and the origin of the document, such as: author name, system where it was created, authority identifier, date and time of creation (timestamp), etc. The environment where the document is created or modified is considered here to be a trusted environment.

4.2. Authentication and tamper proofing

However embedding such document ID information is unsecure by itself. First, the hidden data is vulnerable to the *copy attack*, a protocol attack which estimates the modulated features from a protected document (without knowing the key) and remodulates another document identically – thus copying the document ID information into another document, thus creating an ambiguity. Therefore the document should be *authenticated* at the same time using content-dependent data, such as a *hash code* or a *digital signature* of the textual content and of the ID information. Secondly, tamper proofing (with localization capability) can be achieved by authenticating the text by parts or *blocks*. This means that a local modification of the textual content can be detected with the resolution of one block, i.e. one word, one or several line(s), one paragraph, etc., depending on the embedding rate of the data-hiding algorithm.

Typically (Figure 3) in the i-th block $\mathbf{x}_{text,i}$ is embedded the signature $\mathbf{h}_i = \mathbf{h}_{\mathbf{k}_{enc}}(\mathbf{x}_{text,i}; \mathbf{x}_{text,\sigma(i)}; ...)$ using key $\mathbf{k}_{enc}$, where $\sigma(i)$ represents the neighbouring blocks (e.g. $\sigma(i) = \{i-1, i+1\}$), and the "..." additional data such as document ID information and previous signatures. For the verification (Figure 4), recomputed signatures from the possibly distorted text block $\tilde{\mathbf{h}}_i = \mathbf{h}_{\mathbf{k}_{dec}}(\mathbf{v}_{text,i}; \mathbf{v}_{text,\sigma(i)}; ...)$

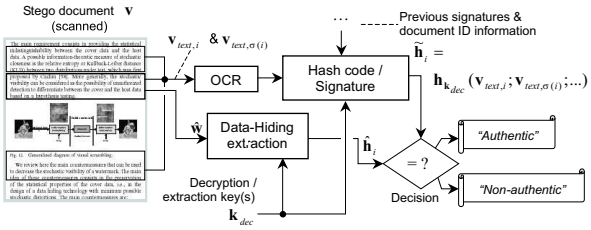

Figure 4: Authentication and tamper proofing: verification.

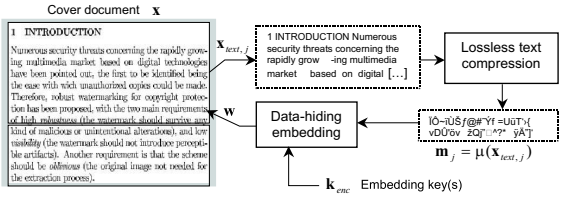

Figure 5: Text document self-embedding.

is compared with the extracted one $\hat{\mathbf{h}}_i$ in order to detect block modification [2]. Signing key $\mathbf{k}_{enc}$ should be known only by the trusted authority allowed to produce "valid" documents. Verification $\mathbf{k}_{dec}$ with $\mathbf{k}_{dec} = \mathbf{k}_{enc}$ is also secret in symmetric protocols; but the most useful protocol is the asymmetric one with $\mathbf{k}_{dec} \neq \mathbf{k}_{enc}$, private $\mathbf{k}_{enc}$ and public $\mathbf{k}_{dec}$, allowing everyone to control the document authenticity. Moreover, it is shown that some measures are mandatory to make tamper proofing secure against malicious partial document substitution, composition, and cryptographic attacks [3]. Main measures consist of: including a unique ID or timestamp into the payload – here the document ID information; chaining neighbouring blocks and previous signatures as mentioned; and using *undeterministic* signatures.

4.3. Data-hiding based content recovery

If the data-hiding scheme achieves sufficient rate, document content recovery is possible based on *self-embedding*. It is illustrated in Figure 5: a losslessly compressed version of the j-th text content part $\mathbf{x}_j$ (potentially different from blocks used for tamper proofing), $\mu(\mathbf{x}_j)$, is embedded in the same part resulting into $\mathbf{y}_j$. Lossless text compression with high compression factors can be used, such as the Lempel-Ziv (LZ) algorithm. When used jointly with authentication and tamper proofing, the protocol can not only detect local alteration, but also restore the original parts.

5. Conclusions

We presented generic text data-hiding based protocols for documents, in both electronic format and hardcopy form. The presented protocols are suitable for document authentication and tamper proofing, content self-recovery, and automatic document processing. Applications can be among others authentication documents (such as passports and ID cards), payment documents, contracts, letters, and technical reports. They provide cheap and convenient solutions for many practical scenarios, requiring only standard printers and scanners. Moreover these frameworks can be integrated directly into common text document editing/publication tools.

[2] Note that for automatic verification of printed documents, an OCR is required at the decoder for recomputing text signatures.

Acknowledgments

This work was partially supported by the Swiss NCCR IM2 project - Interactive Multimedia Information Management, the Swiss SNF grant No. 21-064837.01 and SNF Professorship grant No. PP002-68653/1 projects, as well as by the European IST FP6-507609 SIMILAR and IST-2002-507932 ECRYPT Networks of Excellence. The authors are also thankful to Oleksiy Koval, Renato Villán, Emre Topak and Sergei Startchik for many helpful and interesting discussions.

References

[1] A. M. Alattar and O. M. Alattar. Watermarking electronic text documents containing margin justified paragraph and irregular line spacing. In *IS&T/SPIE Electronic Imaging 2004*, volume 5306, pages 685–695, San Jose, CA, USA, January 2004.

[2] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. Natural language watermarking and tamperproofing. In *5th Information Hiding Workshop (IHW) 2002*, volume LNCS 2578, Noordwijkerhout, The Netherlands, October 2002.

[3] P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen. Toward a secure public-key blockwise fragile authentication watermarking. In *IEEE ICIP2001*, pages 494–497, Thessaloniki, Greece, October 2001.

[4] I. Cox, M. L. Miller, and J. A. Bloomr. *Digital Watermarking*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, October 2001.

[5] F. Deguillaume, S. Voloshynovskiy, and T. Pun. Secure hybrid robust watermarking resistant against tampering and copy attack. *Signal Processing*, 83(10):2133–2170, October 2003.

[6] F. Jordan, R. Meylan, and M. Kutter. Method for preventing counterfeiting or alteration of a printed or engraved surface. US Patent 10/380,914, European Patent PCT/CH01/00560, September 3 2000.

[7] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(1):38–62, January 2000.

[8] C. I. Podilchuk and E. J. Delp. Digital watermarking: Algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–46, July 2001.