Steganographic Watermarking for Documents

Benjamín Barán, Santiago Gómez and Víctor Bogarín
Email: [bbaran, vbogarin]@cnc.una.py
Centro Nacional de Computación – CNC
Universidad Nacional de Asunción – UNA
P. O. Box: 1439 - University Campus of San Lorenzo - Paraguay
Phone: (+595-21)585550

Abstract

The present paper defines Digital Seals for Documents, their scope, application environment and their limitations. These seals can be used to insert information on documents, as it is done with watermarking, or to utilize documents as a communication channel for sending concealed messages, as it is the goal of steganography. Depending on a user needs or preferences, he or she can decide to employ one or another functionality, or a combination of them.

Towards these objectives, a system was developed which constitutes a kit with several Sealing options. The system writes either visible or invisible marks in digital documents, following different methods designed and created in this project. These marks or seals, in turn, can be viewed through a Seal Recognizer.

Implementation is done on a commercial, massively accessible format, supported by word processors commonly used in a modern office. That is the case of RTF (Rich Text Format), which in practice possesses almost all the characteristics provided by other formats such as Microsoft DOC format, and in other aspects a structure similar to that of HTML.

1. Introduction

The accelerated introduction of computerized processes of the last years have contributed to augment security requirements both at the final user level and at the enterprise level, especially since the massive utilization of personal computers, networks, and the Internet with its global availability. Throughout time, computational security needs have been focused on different features: Secret or Confidentiality, Identification, Verification, No Rejection of Authorship, Integrity Control and Availability [12]. To these, we should add the new requirements mainly aimed to: guaranteeing property of digital contents, doing a follow-up of non authorized copies, verifying object integrity and the identity of the contents' sender, providing referential data, controlling quantities of possible copies in well

defined cases, publicly and freely providing copies without a commercial value as a means of marketing to motivate purchase of the announced contents, and so on.

It is often convenient that this protection be set up in a concealed manner, so that it would only be known to the person in charge of maintaining security of the digital contents. This is done in order to avoid a degradation in visual quality of the contents and to complicate the job of those interested in unprotecting the document. This way, a follow-up of the digital document may be performed when needed, without arising suspicion.

Innovative digital techniques are being proposed within the new technological realities, oriented to the solution of one or more of the aforementioned areas and concerning security in the electronic exchange and storage of information. Some of these techniques are presented in what follows.

Cryptography is the art and science of maintaining secure messages. It has been utilized since ancient times, particularly for military purposes, and it has been developed together with technological advances. There even exist security products to which some countries apply restrictions related to the export of weapons.

Encryption, also known as *ciphering*, is the process of transforming a document so as to make it unreadable to whom does not possess its corresponding access key (*password*). The process of taking a ciphered message back to normal is known as deciphering *or de-encryption* [12].

An interesting alternative in relation to security, but within the context of protecting the intellectual property of written work, is given by a technique almost as ancient as paper manufacturing itself, known as *watermarking.* Its first uses had the purpose of registering the manufacturer brand on the product, more recently to certify paper composition, and nowadays many developed countries use it to mark their bills and stamps for more difficult faking. Utilization of the process of watermarking in the digital world has already begun, through the application of signals or patterns inserted in digital files of different formats (images, video, sound, text, etc.) to identify them in a permanent and unalterable way [14].

Instead of just assuring authenticity or integrity of a document, like in digital signatures or other similar devices, the different forms of watermarking seek to identify origin, author, owner, rights of use, distributor of contents, or authorized user of a digital document, and even determine if it has been processed and/or modified [13].

In a slightly different context, we find another ancient technique, a relative of Cryptography in the area of espionage, which has been perfected throughout time and wars. This technique is called *Steganography*, which is the art of concealing the very existence of information by inserting it in an apparently innocuous object. The word

derives from Greek and literally means "concealed writing". Steganography includes a large set of communication methods that hide the existence of a message. These methods include invisible ink, microdots, character ordering, grids covering most of the characters of a given message except for certain positions, etc. [9].

We can see how the classic concepts of signatures, seals or footprints are being extended under the necessity of applying their digital equivalents to the new computational contexts, with similar purposes of assuring authenticity, ownership and origin, among others[2].

Organization of the paper

This paper is organized in 6 sections. Section 2 presents the basic concepts on Cryptography, Watermarking and Steganography, their fields of application, their purposes, and examples of their increasing uses in the digital world. Section 3 presents the means utilized to attain the objectives of the work. Section 4 shows in detail the program that was written, its options, displays and operating modes. Section 5 presents experiments made on real world documents, practical results observed, their reach, limitations and comparisons for each type of seal. Finally, section 6 comments outstanding features and possible work to continue with the project, and offers a synthesis of the results obtained.

2. Techniques Employed

The combined use of watermarking, steganography and cryptography is convenient due to the following facts:

- if we only use cryptography to protect information, data are not readable but the existence of a secret is evidenced;
- if we only use steganography, data become invisible, but methodical analysis of all possible files, searching for hidden data, would make it possible to uncover them;
- if we only use watermarking, we would provide the information within the objectives of this technique, but it would be easy to remove, falsify or alter it.

When cryptography is applied, once a previously ciphered text has been deciphered, the text is already completely accessible and modifiable. However, the watermark stays inseparably attached to the object. This characteristic makes the combination of watermarking with cryptography more interesting than the use of cryptography alone.

2.1 Cryptography

For centuries, the kind of cryptography that has been in use is the one known as *Private Key Cryptography* or *Secret Key Cryptography*. This name comes from the fact that both sender and receiver of a communication share the same key, which has to remain secret. This type of cryptography is also known as *Symmetrical*

Cryptography, because the same key is used on both communication sides [6]. There exist several secret key algorithms. One of the best known algorithms is DES, which is still used at present in applications such as banking with automatic tellers [TAN97]; therefore it is adopted for the present work.

DES is basically a permutation, substitution and recombination of bits. Normal text is ciphered in 64-bit blocks, giving 64 bits of ciphered text. Parametrization is obtained through a 56-bit key. The process has 19 different stages. The first and last stages are key-independent transpositions. The next to last stage exchanges the 32 right bits with the left bits. The other sixteen stages are functionally identical, only parameterized by different functions of the key. Ciphering and deciphering are performed with the same key [5].

2.2 Watermarking

Some of the objectives of using this technique are: Confirmation of property, Follow up of unauthorized copies, Validation of identification and verification of integrity, Labeling, Usage control and Protection of contents [11].

At the present state of technology, *Watermarking* still presents limitations in all stages of the digital contents protection life cycle. These stages are: insertion of the watermark, distribution of contents, detection of the watermark and interpretation of the watermark. All these are seen from different points of view by the main parties involved, i.e. the contents owners (similarly to contents creators), the contents users and the contents pirates[4]. In the present work, we are interested in watermarking that can be applied to images of text. Basically three methods are well known for this purpose:

- *Line coding.* The lines of text in a document are imperceptibly displaced up or down.
- *Word space coding.* Here, the spacing is altered between words of a justified text line.
- *Character coding.* This involves minor alterations to the shape of characters.

Someone interested in breaking these security mechanisms could accomplish it by just (uniformly or randomly) respacing lines, respacing words or reshaping characters. Marks inserted in a text by any of these techniques can always be removed by rewriting the document, and even this effort can be considerably reduced using scanners with OCR (optical character recognition) [10].

These techniques are applied to image representations of documents, that describe each page of a document as an array of pixels, and are also applied to formatted document files. These are digital files describing the contents of the document and the disposition of pages through a standard format description language such as Postscript, Tex, Troff, etc. [3]. For text prepared under widely available word processors (like Microsoft Word

versions 7.0 and later), watermarks can be inserted in files, but only with aesthetic purposes; that is, a graphic or text object can be printed as background for the document, offering the look of water-marked paper, but in these cases, the mark is easily removable.

2.3 Concealed Writing (Steganography)

Throughout history, people have been hiding information by employing several methods. Putting aside the examples from the world of espionage, steganography can be very useful in real world applications. One of these could be the transmission of credit card numbers through the Internet, where ciphered data act like a magnet for hackers. With steganography, the existence of sensitive information is not even noticed.

Another scenario for its utilization is when it is desired to maintain certain information hidden within an organization, avoiding the risk of being taken by disgruntled employees preparing to start their own business; or when it is desired to manage, copy or send confidential information without knowledge of the secretary or the assistant, because the information affects them; or when it is desired to send a boss, a colleague or a subordinate some information "just for your eyes".

Some programs exist (*steganos*) that apply steganography on text documents under formats TXT, RTF and HTML, by adding tabs and spaces at line ends. This way, a byte per text line is hidden (8 bits can be represented as a combination of 8 tab and space characters). They can be easily detected and even the creators of these programs recommend not to use them, since they prefer the hiding done on images which provides a much greater hiding space.

The main difference between *watermarking* and *steganography* basically lies on the intention. Traditionally, the latter hides information, whereas watermarking extends the information and becomes an attribute of the sealed document. In steganography, the object of communication is the hidden message, and the "packaging" is only a means of sending it. In watermarking, the object of communication is the packaging and the hidden message only references that packaging.

3. The Proposal

This paper proposes the creation of a *"sealer"* that inserts into a commonly formatted document (RTF in our developed prototype), some marks or digital patterns that cannot be perceived when the document is visualized under a regular word processor or editor. Depending on the method employed, ciphering of the seal can be made through a secret key – that way only the ones having the key will be able to view the seal. In other cases, the seal can be directly visualized, i.e. without applying any additional security mechanisms.

The sealed document may be recognized as long as a *seal recognizing program* is available. This program

plays the role of the ultra-violet light that reveals the mark inserted on hundred-dollar bills, and it has the capacity to recognize the seal and any other additional information that the owner could have included with it, in the case of invisible or concealed seals. Visible seals are directly seen by any user of the document.

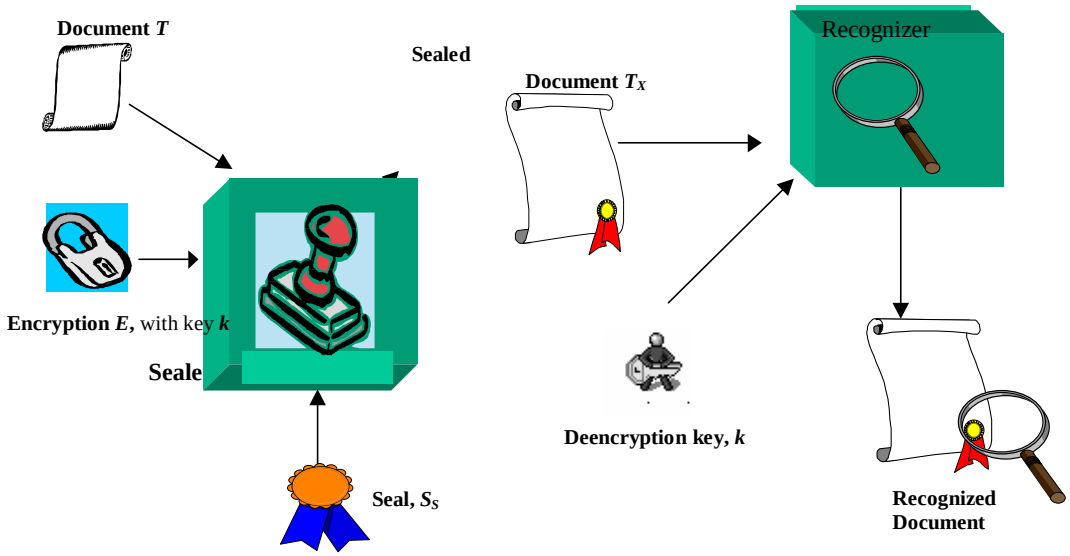

Figure 3.1 **Outline of the implemented prototype.**

3.1 Formalization

Application of a seal S_x to a document T yields as result a sealed document T_x, so that:

$$T_x = T + S_x \quad (1)$$

In order to implement the present proposal, a prototype was developed that performs three varieties of sealing with differentiated features and functions, as explained in this section.

- ***Seals Applied to Format.*** Here, characters in a document T are condensed or expanded in an almost unnoticeable way. This function is represented as the G operator in (2). This way, the condensed/expanded characters make up a pattern, corresponding to a substitution alphabet denoted by A, whose later reading enables the recuperation of the concealed message M. This message may be in turn ciphered, $E(M)$, with a secret key k and personalized for user u, that would use, in this case, a personalized alphabet A_u.

In case someone without the access key k wants to read the document searching for hidden contents, he/she would not be able to infer its existence, nor access its contents. Besides, codification base64, $b64$, is employed in order to work with ASCII characters instead of binary characters. The resulting string is preceded by the user key, k, used later to detect that the message M in fact belongs to this user [7]. Hence, the seal to the format S_f of document T can be expressed as:

$$T_f = T + S_f \text{, where } S_f = G(A_u, k+(b64(E_k(M)))) \quad (2)$$

The number of characters $N(T)$ of the original document T should be larger than the length $L(M)$ of the watermarking message M. Preferably, make $N(T)>>L(M)$, so that M would go unnoticed.

- ***Seals Applied to the Document Encoding***. In this case, we apply marks hidden in predetermined places of the file. The marks are not affected by the document contents or formatting; this function is represented in (3) by operator g. This mark is an identifier key I to a database W where the registered messages can be found. A secret key k is available, but only known by the person possessing the program, and utilized to cipher the registered messages $E(W)$ [1]. The applicable notation is:

$$T_c = T + S_c \quad \text{where} \quad S_c = g(I{:}E_k(W)) \qquad (3)$$

- ***Visible Seals***. In this case, an object O (text, graphic or image), is inserted as background in all pages of the document. The developed prototype enables having a library with n graphic seals to choose from, to be applied to any document T. These watermarks are visible to the naked eye and they fulfill their aesthetic goals if they are given the additional security feature PD to "*protect document*" through a secret key, a feature offered by some word processors. This feature PD makes the document "read only", that is, viewable but not modifiable. This can be expressed as:

$$T_v = T + S_v \quad \text{where} \quad S_v = O_n, \qquad (4)$$

and optionally, we use the feature $PD(T_v)$.

3.2 The Program

In essence, the program is a kit of seals with a variety of features and utilities, focused on different aspects of the *Watermarking* and *Steganography* techniques. Figure 3.2 shows a scheme of the seal types implemented, with their component modules and the options that each one of them provides.

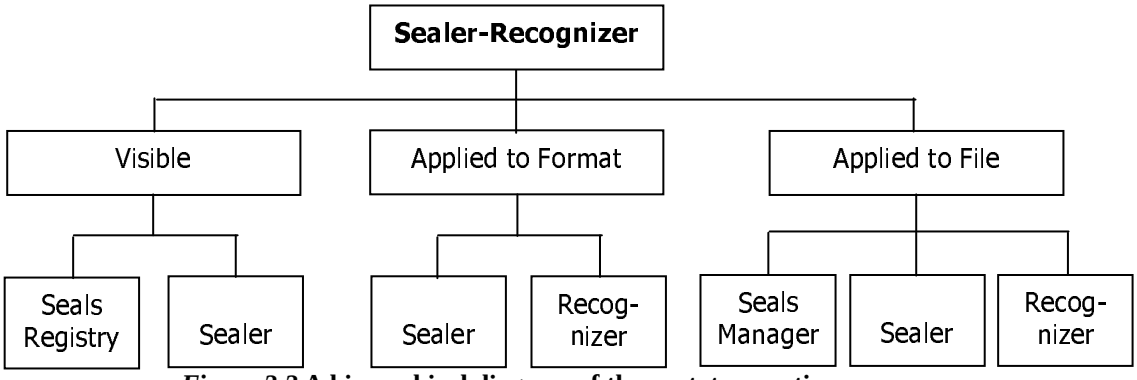

Figure 3.2 **A hierarchical diagram of the prototype options.**

The current version was coded in Visual Basic 5.0 and it has no limitations regarding size of documents to be sealed. The entire program was developed with only one language, including the interfaces, ciphering and

deciphering routines, conversion to/from base64, management of the database, and handling of the document files.

In what follows, we show brief *pseucodes* of the sealer subprogram logic, from where the logic of the recognizer programs can be intuitively understood.

According to what was defined in (1) and (2), the *Seal Applied to Format* algorithm performs the following steps:

Write message M to be inserted.
Choose document T to be sealed.
  Verify that the document length is not shorter than the number of characters in message M, i.e. $L(T) > N(M)$
  Cipher, M, using DES with $k \rightarrow E_k(M)$
  Code $E_k(M)$ with base64 $\rightarrow b64(E_k(M))$
Traverse the internal coding of T until the first available place is found for the seal to be inserted.
If there are already previous messages, append after last message,
    While there are letters in $k+M$
      Take letter of $k+M$
      Search the corresponding formatting commands for that letter according to substitution alphabet A_u
      Insert those commands before the paragraph of T that will be modified
      Go to following character of paragraph T
    End while
Save document $T_f = T + S_f$ with the included formatting codes.

Pseudocode 1. **Seals applied to format S_f.**

Following what was defined in (1) and (3), the *Seal Applied to Document Encoding* algorithm does these actions:

If seal was not registered yet
    Cipher the new seal W utilizing user k's secret key $\rightarrow E_k(W)$
      Add new seal or message W to seals database
End if
Choose seal $E_k(W)$ to be inserted, by viewing W from screen
  Choose document T to be sealed
Search the formatting commands among which access identifier I will be inserted to seals database
Insert I
Save document $T_c = T + S_c$.

Pseudocode 2. **Seals applied to file S_c.**

Should the author of a document T want to apply another seal to it, for example for distribution of multiple copies, identifying the authorized receiver in each seal, he/she could first do the necessary copies and then apply the differentiated seal to each copy.

According to the definitions in (1) and (4) , the logic of *Visible Sealing* is as follows:

If seal was not registered yet
    Indicate which document *T* possesses the seal
    Take seal *O* from such document and store it in sealer library
End if
Go to visible sealing option
Indicate document *T* to be sealed
Choose visible seal O_n to be applied
Insert seal O_n
Save document $T_v = T + S_V$

Pseudocode 3. **Visible Seals S_v.**

4. The Sealer-Recognizer

The program can be operated either from its pull-down menu, or from the tool bar through the corresponding icons, which give access to the following functions.

4.1 Seals Applied to Format: S_f

Their work is based upon application of a seal that is not perceptible by the human eye. Seeking it through an exhaustive search on a normal word processor is an extremely laborious process.

Figure 4.1 shows the previous (normal) visualization of a document without alteration. In figure 4.2, we can see the screen that is utilized to type the message *M* to be inserted and the windows where the user selects the unit, folder and file *T* that will be used as vehicle for the message. Once the text was writen and the file selected, the user double clicks on the file to insert *M* into the document.

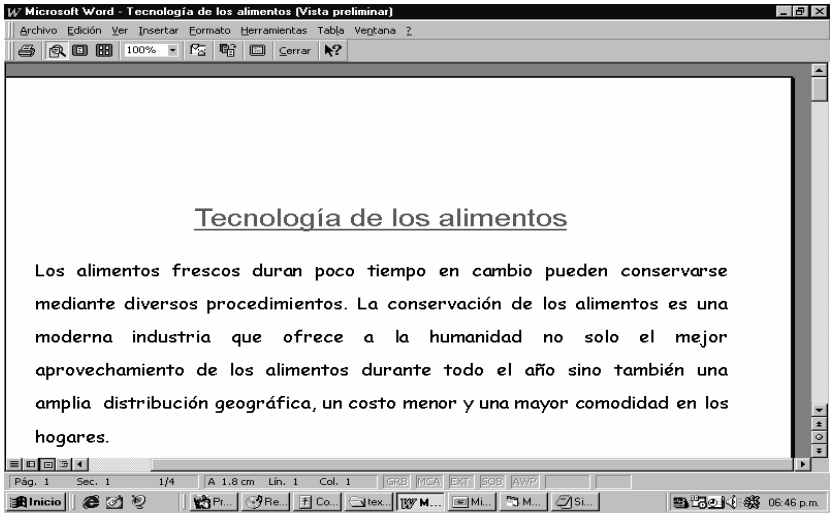

Figure 4.1 **Document *T* before any type of seal.**

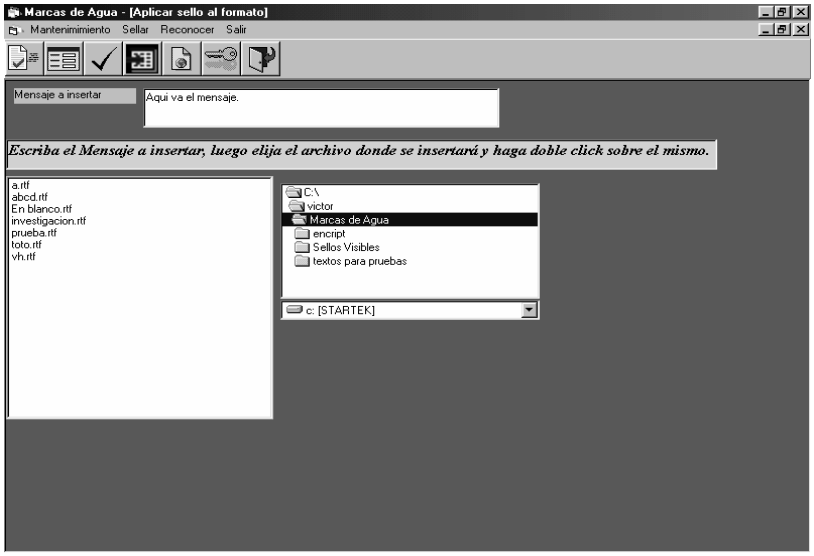

Figure 4.2 **Insertion of seals into the document format.**

In the following pre-visualization of the document (Figure 4.3), it can be seen that the text has not suffered visible changes, since compression and expansion of characters is performed in such a small percentage that is not noticeable to the eye. Modified letters have been colored in blue only for explanation purposes.

Upon an individual inspection of the formatting for each letter in the document, it can be noticed that there are alterations in their expansions, but that does not give any information about their contents, since formatting codes are different from one user to another, and furthermore, message *M* is ciphered and coded according to (2).

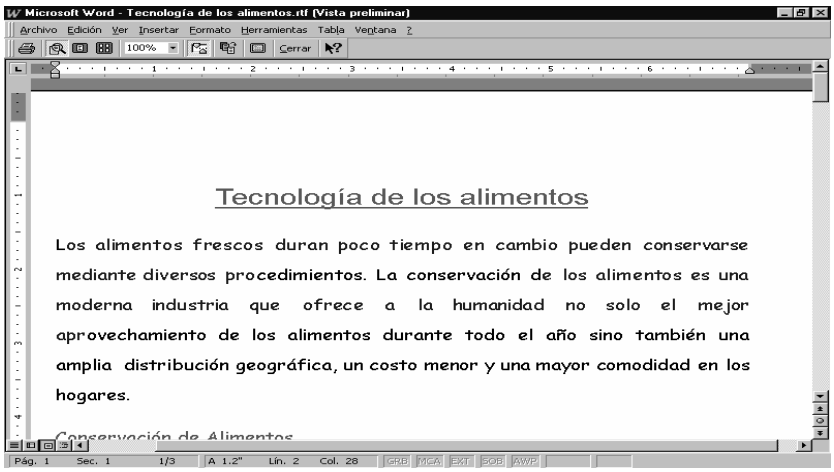

Figure 4.3 **Pre-visualization of document T_x after applying seal to format S_f.**

Figure 4.4 presents the "*Recognizer*" program screen, which has two methods for searching hidden information:

- **Direct method.** By double clicking on a particular file, we verify whether it contains hidden information that belongs to the user, which is determined by the inclusion of the secret ciphered key preceding the message. If such a key is not found and positively recognized, the Recognizer warns that no hidden message was found. That is, if there exists a hidden message inserted by someone else, it will not be identified nor understood by a Recognizer different from the one in use by the person who inserted the seal.

- **Complete method.** This method performs a complete analysis of the selected folder. In this case the search for hidden messages or seals is applied to all files, displaying each found message and the file that stores it, and so continuing until inspection of all files in the directory is complete.

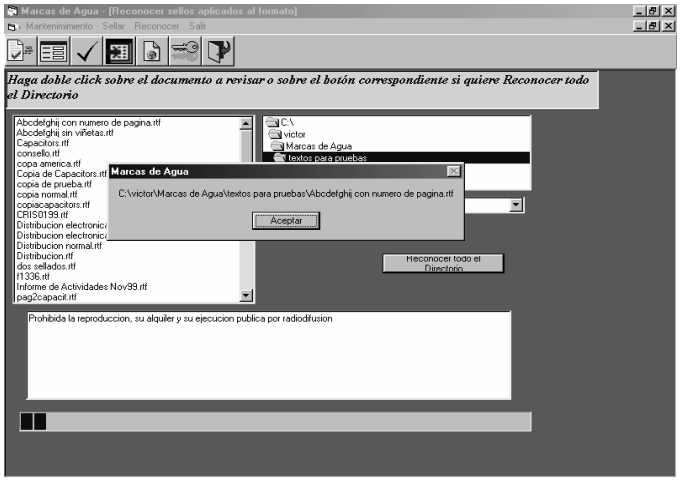

Figure 4.4 **The Recognizer screen for seals applied to format.**

4.2 Seals Applied to the Document Encoding: S_c

Text *W* associated to each document *T* is defined by user *u*. Normally, for each copy we would include at least the user's name and additional information as the authorized receiver's name (if the user wants to individualize the receiver); the rights of use and any other additional information.

To seal a document, we go to the corresponding screen, either from the pull-down menu or via the appropriate icon. Once there, all previously registered marks or seals *W* are exhibited by the program. These marks are stored in a local database, but ciphered (*E*) with the user's secret key *k*, so that if they fall in some unauthorized third party's hands, they will be of no use. To be displayed on the screen, these seals get deciphered, $D_k(E_k(W))$, so that the user can choose the seal to be inserted along with the unit, directory and document where it will be inserted.

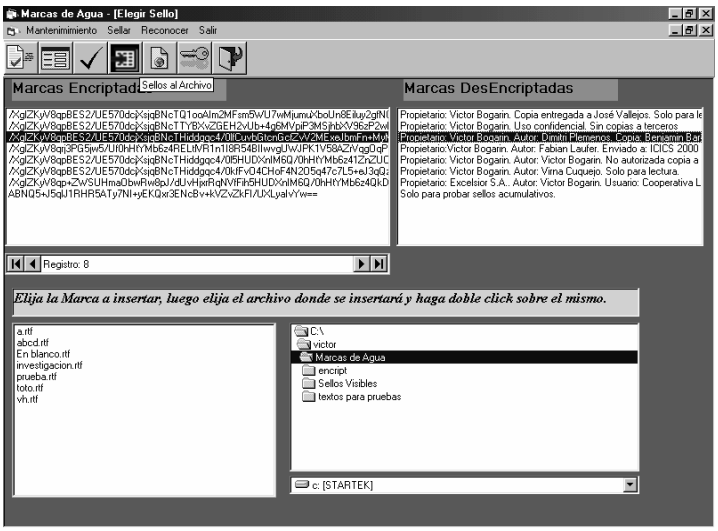

Figure 4.5 **Screen for seals applied to the file.**

The Recognizer program is applied to the entire selected directory, showing in a window those documents that are found with marks, according to the method described in (3).

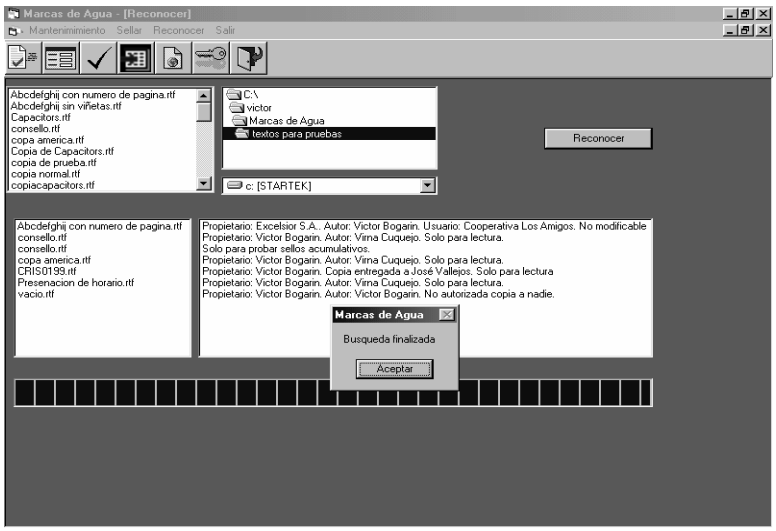

Figure 4.6 **Screen for seals applied to the file.**

Also, messages that correspond to those marks, W, are shown after accessing the database through the marks and after deciphering on screen, $I{:}E_k(W)$, to be viewed by the user. A progress bar indicates the advance in the revision of files or other documents.

Cumulative seals are possible, for the same user or among different users, but retaining capacity to determine

the order of creation of the seals should it become necessary.

$$T_c = T + S_{c1} + S_{c2} + ... \quad (5)$$

Furthermore, a seal maintenance program is provided to add, modify or delete seals besides traversing their database. In this case, the code and the message description are assigned (both of them ciphered (E_k)) before the seal is inserted to the database. This ciphered code or identifier, $I{:}E_k(W)$, is the mark inserted in the file whose detection and retrieval yields to the text W of the applied seal.

As it can be noticed, the secret key k is initially taken at user's entry into the sealing system, and from there it is transparently employed in all called programs, with the user almost not perceiving it, since all this data manipulation is done internally – even without regard to the type of seal in use.

Figure 4.7 **Screen for maintenance of seals applied to file.**

4.3 Visible Seals: S_v

For a visible seal, an object O (text or graphic) is inserted as background for all pages of the document. This can be also accomplished through basic functions offered by the latest versions of some word processors; but here this technique is included in the prototype for a possible combined application of visible seals with other sealing methods, as it will be explained later. Figure 4.8 presents the pre-visualization of a document, without any alteration or a visible seal applied to it.

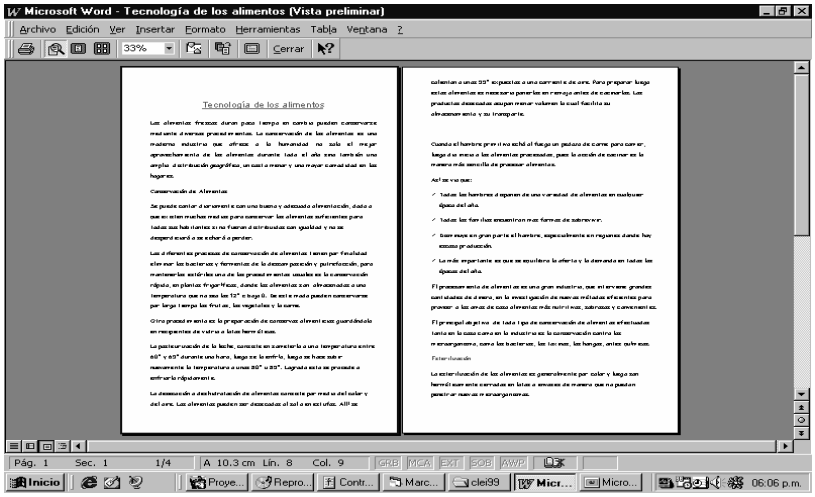

Figure 4.8 **Pre-visualization of a document without a visible seal.**

Figure 4.9 shows the components for the program that inserts the visible seals. The unit, directory and document where the insertion will occur are selected, as well as the seal itself, from the list of registered seals that are property of the user. When one of the registered seals is selected, it is displayed on the corresponding screen. Once all desired elements have been selected, touching the "Seal" button executes the process.

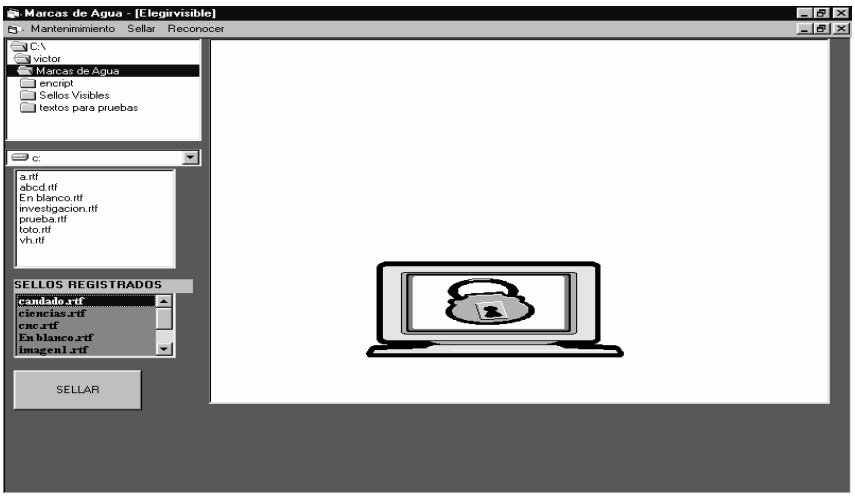

Figura 4.9 **Screen for application of visible seals.**

Figure 4.10 displays the pre-visualization of the document once it was sealed with a visible seal. These seals can be accumulated, that is, it is possible to add another visible seal to the same document and both of them will be inserted and they will overlap when displayed.

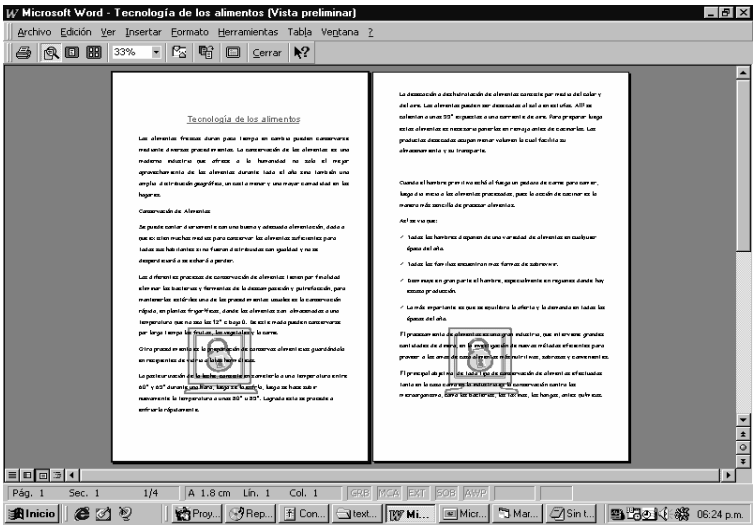

Figure 4.10 **Pre-visualization of a document with a visible seal.**

Figure 4.11 shows the method for "Registering Visible Seals", which remain available for their insertion afterwards. To this end, we choose the unit, directory and file that contains a visible seal or watermark. Then, a name is chosen to describe it in the library of seals. Once a validation processes are completed and the "Extract Seal" button was clicked, a new seal is obtained from that document and stored for later use.

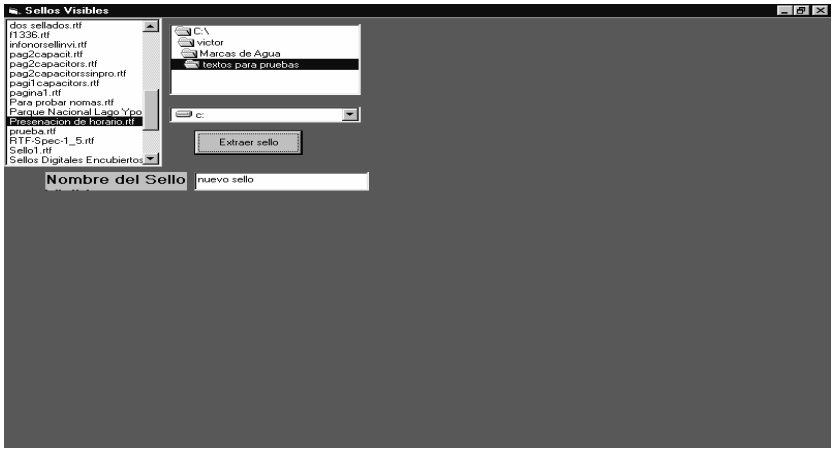

Figure 4.11 **Obtaining visible seals from previously sealed documents.**

A comparable result can be obtained from some word processors, with the corresponding options and the copy and paste functions from a document with a watermark or through the insertion of objects. However, here we store visual objects independent of the original document, forming a library of objects, thereby saving us all the steps required by a commercial word processor.

5. Experimental Results

This section presents experiences obtained with real world documents, practical results observed, the scope and limitations of each type of seal, and comparisons between them.

5.1 Sealing to the Format: S_f

Once a Seal to Format $\boldsymbol{S_f}$ was applied through the implemented kit, the resulting document, maintains the seal even if:

- the entire document is copied onto an empty or non-empty document;
- a sealed paragraph is copied to a sealed or unsealed document;
- fonts are changed in style, size, color, etc., in the sealed paragraph;
- new blank spaces, characters and text are inserted within sealed text;
- it is sealed several times, either with the same seal or with other seals, that is, seals can be accumulated;
- if sealed paragraphs from two different documents are copied to a third document; both seals will be retained.

From here, it can be noted that Seals to Format are quite resistant and dependable, and that the only way to remove them without affecting document contents is by applying (uniform or arbitrary) character spacing. This can be performed through the word processors' own functions, or by editing the document with a plain editor and removing all spaces that do not correspond to the default. The latter alternative involves too much work and an advanced knowledge of the coding in use.

The overhead implied by this type of seal is about 20 bytes per inserted character, so that the file grows in proportion to the length of the inserted message or the redundancy that the user may want to add, for instance, having all or most paragraphs sealed.

Assuming the existence of the seal is unknown to any person other than the author, it is possible to do a follow-up of paragraphs or entire documents copied from another one. Usefulness of this feature lies not only in the inclusion of watermarking with data like theme, author, property, rights of use, or any other information that could be included in a document, but also in the possibility of employing it in Steganography. That is, the document could serve as a hidden communication channel without having to cipher a message to make it unreadable and thereby producing evidence that a document is being sent with confidential, informative, alert or other kind of contents.

The ciphered text of the hidden message could be obtained if the characters upon which the seal was applied are located and the employed compression-expansion rate (which becomes a substitution alphabet) is known. Only the owner of the Sealer-Recognizer program possesses this information, since the program is personalized not only in its secret key but also in that the alphabet

employed is made unique for each user. This is accomplished by randomly generating the compression-expansion pattern at program installation time, causing the characters in the alphabet (in this version, 72 characters including upper and lower case letters, numbers and punctuation marks) to be differently combined in each installation. For instance, codes affecting format and meaning of "A" for a user, are not the same for another user. On the next line we show examples of the letter "A" with different compression rates (0.1 expansion, 0.1 compression, 0.2 expansion and 0.2 compression respectively), which could correspond to different users.

A, A, A, A

Another important feature of this alternative is that it enables for the message to be self-contained in the document, that is, no auxiliary database is needed to recognize the message.

5.2 Sealing the File Encoding

For the implementation of this option, a signal is included in the body of the document, identifying the author plus other additional information, with the following characteristics:

- it cannot be detected by simple document inspection when viewed under word processing software;
- it is not easily removable with minor changes to the document (obviously, if the entire document is rewritten, the new version will not be sealed);
- data added that make up the "seal" do not cause an excessive growth compared to original file size.

A Seal Applied to File Encoding requires the program user to have files where his/her messages are stored, since only an access key (a short string of characters) to the actual message database is inserted in the document.

As an example, consider the size of office memoranda or notes, which are normally around a few dozen Kbytes, whereas in turn, code added as part of the seals would be around a few dozen bytes. This means that file growth in this case is less than 0.1%. The larger the files, the smaller will be the ratio of additional data to original data (overhead), because the seals overhead have a constant size.

The method of Seal Applied to File Encoding has certain aspects that can be regarded as limitations, or at least, points to be remembered. For example, if someone knew the existence of the seal in the document, he/she could edit it, modifying its contents or rewriting another text. The problem here is that a seal belonging to the original author would be obtained by someone who is not its true owner. It is not very clear what benefit would be accomplished in this way by the intruder, since he/she would only be assigning a third party's seal to a newly written document. Given that the author is the only person capable of detecting that a document is his/her property, this maneuver does not seem to make much sense.

A way of detecting the seal would be to leave the document blank, with the seal alone, and to create another blank document without a seal, and view both with a standard editor. The differences found thorough detailed inspection would give the seal. Another characteristic is that when copying one or several paragraphs, the seal is not transported.

5.3 Visible Seals

Visible seals have a clear purpose and usefulness. The inserted information can be the company logo, a slogan, a Web or electronic address, etc. A characteristic is that this information is inserted only once, but it is visible and accessible from any page on the sealed document.

This type of seal can be truly useful and effective by making the document not editable nor modifiable, that is, only viewable, which can be done by protecting the document so it can be read only. This functionality is provided by some word processors and it permits freezing the contents of a document, which is in fact a desired characteristic in electronic publications, technical reports, memoranda, etc.

However, because of some functional weaknesses in commercial word processors, related to re-saving, format conversion and unprotection, there exist procedures that can be performed to eliminate this read only protection, as it will be explained.

To get a higher security level, document modification could be avoided via the “Protect Document” standard functionality that some word processors come equipped with. This way, the document cannot be altered, edited, erased, etc., and even after resaving a read-only file with a different name it keeps the protection characteristics of the original. Therefore, it would not be possible to modify any protected document if the password is unknown. However, in some commercial word processors the weakness is given by the fact that when saving the new file its format can be selected. For instance, the RTF format could be used, and it will maintain most original characteristics, except for security. When the document is reopened, it can be unprotected without requiring the password – hence word processors ought to have a security option to prevent re-saving a protected file.

All this causes protection against writing to be almost nonexistent in most commercially available word processors, with a password for file opening the only way that really works, but at the cost of turning the document inaccessible to someone who does not know the secret password. This is not useful when the goal is to publish text that has to remain unchanged, as it is the case of an electronically distributed publication, or a publication on the Internet.

Even more, the utilization of a password for file opening used to be a fragile option until recent versions of Microsoft Word, which can be pointed as an example of little concern about security in certain commercial products. There were even Internet sites offering

programs for obtaining the secret key of a given document (until version 7 of M.S. Word) or simply decipher them. From version 97 on, security in this product is more reliable and programs sold for deciphering now only use a "brute force" method, that is, they try every possible key [8].

The possibility of "document protection" is particularly important when Visible Seals are employed, since in this mode the existence of information inserted by the owner or author is evident to the eye. In other types of seals it is not so important since the read-only status could rise suspicion about the document contents.

Emphasis of this work is put on *enough security* of documents of the "office type", based upon simple, easy to use techniques that are sufficiently resistant to attacks in such environment.

5.4 Comparisons between the implemented seals

A scheme is presented, to compare the main characteristics of each type of seal in order to distinguish their outstanding differences and similarities.

Concept	**Seal to Format: S_f**	**Seal to File Coding: S_c**	**Visible Seal: S_v**
Overhead	Proporcional to the message *M*	Constant and small, given by *I* (identifier of the Registered Message database, *W*)	Proporcional to the size of *O*, the visual object.
Location of *M*	Self-contained in document *T*	It is in the database, accessed through *I*	As background of document in *T*
Additional requeriments	Substitution alphabet of format codes, *A*	Database	Seal library
Process	$T_f=T+G(A_u,,k+(b64(E_k(M))))$	$T_c=T+g(I:E_k(W))$	$PD(T_v) = PD(T+O)$
Transportability (if sealed document text is copied, is the seal transported?)	Yes	No	No
Affected by document formats?	Only affected by changes in the expansion of characters.	No	No. Only affected by manipulation of the background.
Ciphering	Yes	Yes	No
Hiding	Yes	Yes	No

A limitation that was found with these methods as well as with other published or commercialized ones about watermarking or steganography, is that all marks are removed when files are converted to text format (TXT). But in this case all format codes, font styles, paragraph layouts, etc. that make up the document itself, are lost.

6. Conclusions

Finally, let us summarize the novel features of the project, as well as proposals for future work that would give continuity to all what was developed and would enhance the functionality of the prototype, turning it

conceptually more robust and even turning it applicable within the enterprise, government and scientific world. In this manner, its usefulness for real world applications can make marketing viable.

6.1 Outstanding Features

Among the main features of the techniques presented here, the following aspects stand out as conception and implementation novelties:

- The best known application that implements steganography on text, hides one byte per text line and it can be easily detected. In this project, one byte is hidden per character, allowing a significantly larger space for hiding purposes. Also, the detection of hidden information involves a lot of work, with even more margin remaining to increase robustness.
- As mentioned in section 2.1 on watermarking, this is generally applied to text images, with three existing methods: coding of the line, coding of word spacing and coding of characters. This work defines and implements other variants: character compression inserted in the internal code of the file, and visible sealing.
- Most published methods to date require the unmarked version of the document in order to identify the mark. Concealed information stored is detected through a comparison with the unmarked document. In this proposal it is not necessary the original document.
- Generally, watermarking on text is based upon the processing of text images, resulting either from conversion of the original or scanned from printed copies. In this work, files are processed in their original formats, that is, as text and not as image files.
- Some work has been published where watermarking techniques are applied to Postscript formats or to text images. However, in this project the format utilized is RTF, which is recognized by most word processors (as M.S. Word), i.e. it is more widely used, especially in the commercial or enterprise environments.
- Commonly existing watermarking includes some steganography component. The techniques presented here are applicable to both pure watermarking and steganography, as well as to a combination of the two.

6.2 Proposals for Future Work

From the experience gained in the study of all these techniques, the state of the art, the possible applications and the perceived weaknesses, several roads appear for continuity of the work. They can be grouped into the following:

6.2.1 Inclusion as Part of a Word Processor

The developed functionalities can be inserted into a commercially used word processor such as Microsoft Word, in the form of a macro, programmed in Visual Basic. The component would be available either as a menu option or as additional buttons in some of the standard dialog boxes, so that the Sealer-Recognizer program would become integrated to the word processor,

even though it always may be possible to execute it separately.

In this way, seals could be generalized for other formats employed by word processors, because if format is changed from RTF to one such as DOC, and then back into RTF, the seals remain in the document. Therefore, a practical way of taking advantage of the existing kit for any format would be: initially to save the document as RTF, seal it and then save it again in the desired format (say, DOC). For seal recognition, convert all selected files to RTF and then apply the Recognizer. These steps can of course be automated. Another format considered for future work is HTML, widely used in the Internet, because its structure and format are similar to the ones employed in this project.

6.2.2 Augmented Robustness

An important job will be to improve the resistance of seals. Different options will have to be studied to attain that objective. One of them may be to create a format that could be called *Secure RTF*, which would remain ciphered after sealing. The word processor would be enhanced with the ability to recognize this type of file and to transparently decipher it with the user key.

Hiding of the watermarking string (the access key I to the seals database) could also be made more robust, by splitting up the string characters into different locations of the coding and following different patterns, from values randomly obtained for each user.

This would yield $S_c=G_u(I:E_k(W))$, where G_u indicates the use of steganographic hiding methods, and not only the simple insertion as it is the case of the present prototype, which stores the entire string together and at the same location for all users.

Another functionality that can also be added is the distribution of the hidden message in the formatting, by dispersing the message throughout the entire document, depending on the ratio of the quantity of characters between the document utilized as channel and the message that we want to communicate. This way, the search for out of normal formatting codes that could rise a reviewer's suspicion, would be hindered. Currently the message characters are inserted one after another, beginning at the first available place for insertion.

Techniques for error detection/correction can be used to fix "slight" modifications to the document. At the same time, the Sealer-Recognizer can be continuously upgraded turning it as sophisticated as desired, through the addition of a Message Digest, Private and Public Keys, Digital Signatures, and so on, on top of the main techniques proposed here: watermarking, steganography and cryptography, of which the latter is the one with more alternatives, applicable in accordance with the primary purpose in hand. Even more, the Recognizer may be implement as an agent moving from one computer to another, reporting found documents to the owner.

The techniques that comprise traditional watermarking, previously mentioned, can also be added to the program with illustrative purposes, or they can be put to work together with the ones proposed in this paper in order to increase robustness and extend the supported character set.

6.2.3 Applications

Here are some possible further applications of the prototype:

- **Concealed Auditing**. User data such as date and time of each modification could be inserted in a transparent and automatical fashion by the word processor. This function is similar to the auditing systems employed in transaction-oriented applications.
- **A document serving as channel for another document**. With purely steganographic purposes, it is possible to use a text document *T* as a communication channel by inserting concealed TXT or RTF files into that document. This is feasible as long as the holding document provides enough hiding space given by the length of the text *T* compared to the length of the steganographic message that we want to send.
- **Document serving as channel for any other file**. It would also be possible to generalize Seals to Format for any type of file. This can be accomplished, for instance, by converting the object to RTF and then inserting the resulting ASCII file into the document *T* employed as channel.
- **Visible Seals inserted in the format**. What we so far have had as Visible Seals, could be inserted in a concealed way and then the Recognizer could reproduce the corresponding image or graphic. This way, we would have that $S_v = G(A_u(O_n))$ where G is the compression-expansion operator, A_u is the substitution alphabet for user u and O_n is the inserted object, equivalent to the object placed as background in the Visible Seals case.
- **Combined techniques**. A combination of the presented techniques can also be put to use, in order to make them more secure. For example, insert the watermarking database code (currently inserted in the file coding), into the format. Since the code is a short string, this offers the advantage of easier hiding, easier dispersion and the possibility of redundancy. It would permit to obtain $S_{fc} = G(A_u(\mathrm{I}{:}E_k(W)))$ where $S_{fc} = S_f + S_c$.

6.3 Results

The main results obtained with the implemented prototype are:

- to recognize own documents by an author;
- to enable follow-up and identification of unauthorized copies made from sealed documents, provided there is access to the folder where such copies are stored;
- to detect that a file contains parts of a sealed document that belongs to an author;
- to use documents as a means to communicate concealed messages;

There exists a great potential for applications in many present day automated offices, where computational files rather than the traditional printed paper files are handled, and in electronic communications via Internet.

We presented the implementation of a prototype that is applicable to day-to-day situations in which "sufficient security" satisfies user needs because it discourages undue utilization of non-authorized documents, covering an area with a large number of users that are not based on documental databases, such as Lotes Notes and others.

Although the initial objective in this work was the use of watermarking to preserve author copyright and several other known functionalities of this technique, the developed prototype goes further, enabling a more flexible use that even includes steganography, also satisfying this discipline's coverage.

It can be noted that this development has an important practical value and that it is feasible to market its utilization to the general public of word processor users, because it provides functions that are attractive for some, very useful for others and indispensable to the rest. We cannot forget to point out that with the proposed tools it is possible to help protect authors' rights and property. These considerations support the value of the proposal to motivate protection of intellectual and ownership rights for the novel type of published works and goods that make part of the "new" digital world.

References

[1] Barán, B.; Gomez, S., and Bogarín, V.; "Sellos Digitales Encubiertos para Documentos", *IX Panel de Informática* – Asunción, Paraguay, Oct-98.

[2] Berghel, H. "Watermarking Cyberspace", *ACM Communications*, Nov-97.

[3] Brassil, J.; Low, S.; Maxemchuk, N., and O'gorman, L. "Electronic Marking and Identification Techniques to Discourage Document Copying", *IEEE J. Selec. Areas in Comm.*, Oct-95.

[4] Craver, S.; Yeo, L., and Yeung M.; "Technicals Trials and Legal Tribulations", *ACM Communications*, Jul-98.

[5] Garfinkel, S., and Spafford, G. "Practical Unix Security", O'Reilly & Associates, 1994.

[6] Garfinkel, S. "PGP Pretty Good Privacy", O'Reilly & Associates, 1995.

[7] Gomez, S.; Bogarín, V., and Barán, B. "Sellos Digitales para Documentos", *XXV Conferencia Latinoamericana de Informática*–Asunción, Paraguay, Sep-99.

[8] Hernández, J. "Soluciones criptográficas en aplicaciones no dedicadas", *Byte España,* Apr-99.

[9] Johnson, N., and Jajodia, S. "Exploring Steganography: Seeing the Unseen", *IEEE Computer*, Feb-98.

[10] Low, S., and Maxemchuk, N. "Performance Comparison of Two Text Marking Methods", *IEEE Journal on Selected Areas in Communications*, May-98.

[11] Memon, N., and Wong, P.; "Protecting Digital Media Content", *ACM Communications*, Jul-98.

[12] Schneier, B. "Applied Cryptography", Wiley & Sons - 1996.

[13] Zhao, J. "Look, It's Not There", *Byte*, Jan-97.

[14] Berghel, H., and O'Gorman, L. "Protecting ownership rights through digital water-marking", *IEEE Computer* , Jul-96.