2010-01-01

E-Fraud Prevention based on the Self-Authentication of E-Documents

Jonathan Blackledge
Technological University Dublin, jonathan.blackledge@tudublin.ie

Eugene Coyle
Technological University Dublin, Eugene.Coyle@tudublin.ie

Recommended Citation

Blackledge, J., Coyle, E.:E-Fraud Prevention based on the Self-Authentication of e-Documents. The Fourth International Conference on Digital Society, St. Maarten, Netherlands Antilles, pp.329-338.

e-Fraud Prevention based on the Self-Authentication of e-Documents

J. M. Blackledge
School of Electrical Engineering Systems
Faculty of Engineering, Dublin Institute of Technology
Dublin, Ireland
Email: jonathan.blackledge@dit.ie

E. Coyle
School of Electrical Engineering Systems
Faculty of Engineering, Dublin Institute of Technology
Dublin, Ireland
Email: eugene.coyle@dit.ie

***Abstract*—We consider a method for preventing e-Fraud in which a binary image is encrypted with a floating point cipher using a convolution operation and the output quantized into a 1-bit array generating a binary image ciphertext. The output is then 'embedded' in a host image to hide the encrypted information. Embedding is undertaken either in the lowest 1-bit layer or multiple 1-bit layers. Decryption is accomplished by: (i) extracting the binary image from the host image; (ii) correlating the result with the original cipher. In principle, any cipher generator can be used for this purpose and the method has been designed to operate with 24-bit colour images. The approach has a variety of applications and in this paper, we focus on the authentication and self-authentication of e-documents (letters and certificates, for example) that are communicated over the Internet and are thereby vulnerable to e-Fraud (e.g. modification, editing, counterfeiting etc.).**

***Keywords*-Covert encryption, Steganography, Information hiding, Authentication, e-Fraud**

I. Introduction

One of the principal weaknesses of all encryption systems is that the form of the output data (the ciphertext), if intercepted, alerts the intruder to the fact that the information being transmitted may have some importance and that it is therefore worth attacking and attempting to decrypt it. This aspect of ciphertext transmission can be used to propagate disinformation, achieved by encrypting information that is specifically designed to be intercepted and decrypted. In this case, we assume that the intercept will be attacked, decrypted and the information retrieved. The 'key' to this approach is to make sure that the ciphertext is relatively strong (but not too strong!) and that the information extracted is of good quality in terms of providing the attacker with 'intelligence' that is perceived to be valuable and compatible with their expectations, i.e. information that reflects the concerns/interests of the individual(s) and/or organisation(s) that encrypted the data. This approach provides the interceptor with a 'honey pot' designed to maximize their confidence especially when they have had to put a significant amount of work in to 'extracting it'. The trick is to make sure that this process is not too hard or too easy. 'Too hard' will defeat the object of the exercise as the attacker might give up; 'too easy', and the attacker will suspect a set-up!

In addition to providing an attacker with a honey-pot for the dissemination of disinformation, it is of significant value if a method can be found that allows the real information to be transmitted by embedding it in non-sensitive information after (or otherwise) it has been encrypted, e.g. camouflaging the ciphertext. This is known as *Steganography* which is concerned with developing methods of writing hidden messages in such a way that no one, apart from the intended recipient, knows of the existence of the message in contrast to cryptography in which the existence of the message itself is not disguised but the content is scrambled [1], [2]. Steganography provides a significant advantage over cryptography alone in that messages do not attract attention to themselves, to messengers, or to recipients. No matter how well plaintext is encrypted (i.e. how unbreakable it is), by default, a ciphertext will arouse suspicion and may in itself be incriminating, as in some countries encryption is illegal.

This paper presents a method of 'hiding' encrypted information in a colour digital image. In principle, any cipher can be used to do this providing it consists of floating point (or decimal integer) numbers that are ideally, uniformly distributed. The scheme allows for the authentication and self-authentication of documents such as letters, certificates and other image based data. The encrypted watermark can be camouflaged to obfuscate its existence and the applications to which the method can be applied are numerous. For example, the self-authentication of e-documents sent as attachments over the internet provides a unique facility for many legal and financial transactions that have traditionally relied on paper based documents to secure authenticity. The method also provides a unique way of 'propagating' disinformation in the form of an encrypted document which contains hidden information.

II. Steganography

The word 'Steganography' is of Greek origin and means 'covered', or 'hidden writing'. In general, a steganographic message appears as something else known as a covertext. The conversion of a ciphertext to another plaintext form is called *Stegotext* conversion and is based on the use of *Covertext*. Some covertext must first be invented and the

ciphertext mapped on to it in some way to produce the stegotext. This can involve the use of any attribute that is readily available such as letter size, spacing, typeface, or other characteristics of a covertext, manipulated in such a way as to carry a hidden message. The basic principle is given below:

$$\begin{array}{ccccc} & & \text{Data} & \rightarrow & \text{Covertext} \\ & & & & \downarrow \\ \text{Plaintext} & \rightarrow & \text{Ciphertext} & \rightarrow & \text{Stegotext} \\ & & & & \downarrow \\ & & & & \text{Transmission} \end{array}$$

Note that this approach does not necessarily require the use of plaintext to ciphertext conversion as illustrated above and that plaintext can be converted into stegotext directly.

With the wealth of data that is generated and transmitted in todays environment and the wide variety of formats that are used means that there is much greater potential for exploiting steganographic methods than before. In other words, the e-society has generated a camouflage rich environment in which to operate and one can attempt to hide plaintext or ciphertext (or both) in a host of data types, including audio and video files and digital images. Moreover, by understanding the characteristics of a transmission environment, it is possible to conceive techniques in which information can be embedded in the transmission noise, i.e. where natural transmission noise is the covertext. There are some counter measures - steganalysis - that can be implemented in order to detect stegotext. However the technique usually requires access to the covertext which is then compared with the stegotext to see if any modifications have been introduced. The problem is to find ways of obtaining the original covertext.

A. Hiding Data in Images

The relatively large amount of data contained in digital images makes them a useful 'medium' for undertaking steganography. Consequently digital images can be used to hide messages in other images. A colour image typically has 8 bits to represent the Red, Green and Blue components. Each colour component is composed of 256 'colour values' (for a 24-bit image) and the modification of some of these values in order to hide other data is undetectable by the human eye. This modification is often undertaken by changing the least significant bit in the binary representation of a colour or grey level value (for grey level digital images). For example, for 7-bit ASCII conversion, the grey level value 100 has the binary representation 1100100 which is equivalent to character d. If we change the least significant bit to give 1100101 (which corresponds to a grey level value of 101 and character e) then the difference in the output image will not be discernable even though we have replaced the letter e with default character d. In this way, the least significant bit can be used to encode information other than pixel intensity and the larger the host image compared with the hidden message, the more difficult it is to detect the message. In this way it is possible to hide an image in another image for which there are a number of approaches available (including the application of bit modification). For example, Figure 1 shows the effect of hiding one image in another through the process of re-quantization and addition. The image to be embedded is re-quantized to just 3-bits or 8 grey levels so that it consists of an array of values between 0 to 7. The result is then added to the host image (an array of values between 0 and 255) on a pixel by pixel basis such that if the output exceeds 255 then it is truncated (i.e. set to 255). The resulting output is slightly brighter with minor distortions in some regions of the image that are homogeneous.

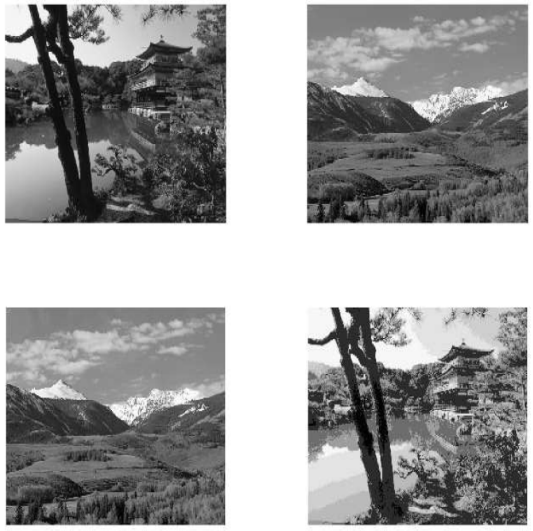

Figure 1. Illustration of 'hiding' one image (top left) in another image (top right) through simple re-quantization and addition (bottom left). By subtracting the bottom left image from the top right image and re-quantizing the output, the bottom right reconstruction is obtained.

Clearly, knowledge of the original host image allows the hidden image to be recovered (by subtraction) giving a result that is effectively completely black. However, by increasing its brightness, the hidden image can be recovered as shown in Figure 1 which, in this example, has been achieved by re-quantizing the data from 0-7 back to 0-255 grey levels. The fidelity of this reconstruction is poor compared to the original image but it still conveys the basic information, information that could be covertly transmitted through the host image as an email attachment, for example. Note that the host image represents, quite literally, the key to recovering the hidden image. The additive process that has been applied is equivalent to the 'process of confusion' that is the basis for a substitution cipher. Rather than the key being used to generate a random number stream using a pre-defined algorithm from which the stream can be re-

generated (for the same key), the digital image is, in effect, being used as the cipher. Note that the distortion generated by re-quantization means that the same method can not be used if the hidden image is encrypted. The degradation in the ciphertext will not allow an accurate decrypt to be accomplished due to loss of data.

Steganography is often used for digital watermarking. This is where the plaintext, which acts as a simple identifier containing information such as ownership, copyright and so on, is hidden in an image so that its source can be tracked or verified. The methods discussed above refer to electronic-to-electronic type communications in which there is no loss of information (assuming the image is not compressed to JPEG - Joint Photographics Expert Group - format, for example). Steganography and watermarking techniques can be developed for hardcopy data which has a range of applications. These techniques have to be robust to the significant distortions generated by the printing and/or scanning process. A simple approach is to add information to a printed page that is difficult to see. For example, some modern colour laser printers, including those manufactured by HP and Xerox, print tiny yellow dots which are added to each page. The dots are barely visible and contain encoded printer serial numbers, date and time stamps. This facility provides a useful forensics tool for tracking the origins of a printed document.

B. Disinformation

Disinformation is used to tempt the enemy into believing certain kinds of information. The information may not be true or contain aspects that are designed to cause the enemy to react in an identifiable way that provides a strategic advantage [4], [5]. Camouflage is a simple example of disinformation [6]. This includes techniques for transforming encrypted data into forms that resemble the environments through which an encrypted message is to be sent. At a more sophisticated level, disinformation can include encrypted messages that are created with the sole purpose of being broken in order to reveal information that the enemy will react to by design [7].

C. Steganographic Encryption

It is arguable that disinformation should, where possible, be used in conjunction with the exchange of encrypted information which has been camouflaged using steganographic techniques for hiding the ciphertext. For example, suppose that it had been assumed by Germany that the Enigma ciphers were being compromised by the British during the Second World War. Clearly, it would have been strategically advantageous for Germany to propagate disinformation using Enigma. If, in addition, 'real information' had been encrypted differently and the ciphertexts camouflaged using broadcasts through the German home radio service, for example, then the outcome of the war could have been very different. The use of new encryption methods coupled with camouflage and disinformation, all of which are dynamic processes, provides a model that, while not always of practical value, is strategically comprehensive and has only rarely been fully realised.

III. Stochastic Confusion and Diffusion

In terms of plaintexts, diffusion is concerned with the issue that, at least on a statistical basis, similar plaintexts should result in completely different ciphertexts even when encrypted with the same key. This requires that any element of the input block influences every element of the output block in an irregular fashion. In terms of a key, diffusion ensures that similar keys result in completely different ciphertexts even when used for encrypting the same block of plaintext. This requires that any element of the input should influence every element of the output in an irregular way. This property must also be valid for the decryption process because otherwise an intruder may be able to recover parts of the input from an observed output by a partly correct guess of the key used for encryption. The diffusion process is a function of the sensitivity to initial conditions that all cryptographic systems must have. Further, all cryptographic systems should exhibit an inherent topological transitivity causing the plaintext to be mixed through the action of the encryption process.

The process of 'confusion' ensures that the (statistical) properties of plaintext blocks are not reflected in the corresponding ciphertext blocks. Every ciphertext must have a random appearance to any observer and be quantifiable through appropriate statistical tests. Diffusion and confusion are processes that are of fundamental importance in the design and analysis of cryptological systems, not only for the encryption of plaintexts but for data transformation in general.

A. Stochastic Confusion

The simplest approach to encrypting plaintext described by the vector $\mathbf{p}$ (e.g. consisting of 7-bit ASCII decimal integers) is to add a cipher denoted by $\mathbf{n}$ which is taken to be a stochastic function consisting of random numbers, i.e. $\mathbf{n}$ is a vector consisting of noise. The ciphertext $\mathbf{c}$ is then given by

$$\mathbf{c} = \mathbf{p} + \mathbf{n} \tag{1}$$

This is an example of a substitution cipher and relies on the use of ciphers that can be regenerated by a 'key' which is the initial condition used to 'drive' a random number generated whose algorithm is known. For a 7-bit ASCII code, if the value of any element of the array $\mathbf{c}$ exceeds 127 then the result is wrapped (e.g. 121+10=3), the output being taken to be the modulo of 127. In practice, this process is usually carried out in 'binary space' by first converting the arrays

$\mathbf{p}$ and $\mathbf{c}$ to binary form $\mathbf{p}_b$ and $\mathbf{c}_b$, respectively. The binary ciphertext $\mathbf{c}_b$ is then computed using the XOR operation, i.e.

$$\mathbf{c}_b = \mathbf{n}_b \oplus \mathbf{p}_b$$

The plaintext is given by

$$\mathbf{p}_b = \mathbf{n}_b \oplus \mathbf{c}_b$$

Irrespective of whether this simplest of encryption schemes is implemented in decimal integer or binary space, it is imperative that the cipher exhibits statistical properties such that there is no bias associated with the frequency of occurrence of any element. In other words the histrogam of $\mathbf{n}$ must be uniformly distributed in order to counteract a statistical attack based on analysing the frequency of occurrence of elements of $\mathbf{p}$ in which the space between each word is the most common. Equivalently, in binary space, it is important that there is no bias towards a 0 or 1 so that the number of bits should be the same in any binary cipher. The additive process associated with equation (1) represents a process of *confusion* based on, what is in effect, the addition of a uniformly distributed noise. It is an example of stochastic confusion upon which the majority of both substitution and transposition (or both) ciphers are based. The aim is to generate a maximum entropy cipher in such a way that there is maximum possible diffusion in terms of key dependency (i.e. that a change in any single bit of the key can effect any, and potentially, all bits of the cipher). This is the usual concept in which the term *diffusion* is used in cryptology, the aim being to maximize (in terms of the entropy of the ciphertext) the process of both diffusion and confusion.

B. Stochastic Diffusion

In this paper we consider the diffusion of plaintext in terms of the convolution of a plaintext array with the cipher, the ciphertext being given by

$$c[i] = n[i] \otimes p[i] = \sum_j n[j-i]p[j]$$

where $\otimes$ denotes the convolution sum as defined above.

It is well known that for $x \in (-\infty, \infty)$, the solution to the diffusion equation

$$(D\partial_x^2 - \partial_t)c(x,t) = 0, \quad c(x,0) = p(x)$$

where D is the *Diffusivity* and $p(x)$ is the initial condition, is given by [9]

$$c(x,t) = g(x,t) \otimes_x p(x) = \int_{-\infty}^{\infty} g(y-x,t)p(y)dy$$

where $\otimes_x$ now represents the convolution integral (over independent variable x) as defined above and

$$g(x,t) = \sqrt{\frac{1}{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right)$$

Here, the diffusion of $p(x)$ is determined by a convolution with a Gaussian function whose standard deviation is determined by the product of the Diffusivity D and the time t over which the diffusion process occurs. Stochastic diffusion is based on replacing the convolution kernel (i.e. the function g) for a stochastic function n so that

$$c(x,t) = n(x,t) \otimes_x p(x)$$

For any time t, the ciphertext is given by the convolution of the plaintext with the cipher. In this sense, any fixed value of time $t_i, i = 1, 2, 3...$ can be taken to represent the initial value used to generate $n(x)$, i.e. the key used to initiate a random number generating algorithm.

The inverse problem associated with stochastic diffusion as defined above is not as simple as applying an XOR operation to the ciphertext in binary space. In this case we are required to apply a suitable deconvolution process, i.e. to solve the problem: Given c and n, compute p. We can deconvolve by using the convolution theorem giving

$$p(x) = \mathcal{F}_1^{-1}\left[\frac{C(k)N^*(k)}{\mid N(k) \mid^2}\right]$$

where N is the Fourier transform of n, C is the Fourier transform of c, k is the spatial frequency and $\mathcal{F}_1^{-1}$ denotes the (one-dimensional) inverse Fourier transform, i.e. for a piecewise continuous function $f(x)$ with spectrum denoted by $F(k)$,

$$\mathcal{F}_1^{-1}[F(k)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(k)\exp(ikx)dx = f(x)$$

This approach requires regularization in order to eliminate any singularities when $\mid N \mid^2 \rightarrow 0$ through application of a constrained deconvolution filter [10], [11]. It is not appropriate for encryption because the inverse process is fundamentally ill-conditioned. Instead we consider a method which allows the inverse problem to be solved directly by correlation. To do this the cipher needs to be 'pre-conditioned'. We let

$$m(x) = \mathcal{F}_1^{-1}[M(k)]$$

where $\forall k$

$$M(k) = \begin{cases} \frac{N^*(k)}{|N(k)|^2}, & \mid N(k) \mid \neq 0; \\ N^*(k), & \mid N(k) \mid = 0. \end{cases}$$

The ciphertext is then given by

$$c(x) = m(x) \otimes_x p(x)$$

This result allows us to solve the inverse problem by correlating (denoted by $\odot_x$) c with n. This is based on the following analysis: Using the convolution theorem

$$m(x) \otimes_x p(x) \leftrightarrow M(k)P(k)$$

and, using the correlation theorem,

$$n(x) \odot_x c(x) \leftrightarrow N^*(k)C(k)$$

where $\leftrightarrow$ denotes transformation into Fourier space. Thus

$$N^*(k)C(k) = N^*(k)M(k)P(k) = N^*(k)\frac{N^*(k)}{\mid N(k) \mid^2} = P(k)$$

so that

$$p(x) = n(x) \odot_x c(x)$$

The pre-conditioning of a cipher so that decryption can be undertaken using correlation provides a simple solution for utilizing the process of stochastic diffusion to encrypt data. In this paper, the process is applied in 'image space' to watermark a digital image.

IV. Digital Image Watermarking

In 'image space', we consider the plaintext to be an image $p(x, y)$ of compact support $x \in [-X, X]$; $y \in [-Y, Y]$. Stochastic diffusion is then based on the following results:

Encryption

$$c(x, y) = m(x, y) \otimes_x \otimes_y p(x, y)$$

where

$$m(x, y) = \mathcal{F}_2^{-1}\left[M(k_x, k_y)\right]$$

and $\forall k_x, k_y$

$$M(k_x, k_y) = \begin{cases} \frac{N^*(k_x,k_y)}{|N(k_x,k_y)|^2}, & \mid N(k_x, k_y) \mid \neq 0; \\ N^*(k_x, k_y), & \mid N(k_x, k_y) \mid = 0. \end{cases}$$

Decryption

$$p(x, y) = n(x, y) \odot_x \odot_y c(x, y)$$

Here, k_x and k_y are the spatial frequencies and $\mathcal{F}_2^{-1}$ denotes the two-dimensional inverse Fourier transform. For digital image watermarking, we consider a discrete array $p_{ij}, i = 1, 2, ..., I; j = 1, 2, ..., J$ of size $I \times J$ and discrete versions of the operators involved, i.e. application of a discrete Fourier transform and discrete convolution and correlation sums.

If we consider a host image denoted by $h(x, y)$, then we consider a watermarking method based on the equation

$$c(x, y) = Rm(x, y) \otimes_x \otimes_y p(x, y) + h(x, y)$$

where

$$\|m(x, y) \otimes_x \otimes_y p(x, y)\|_\infty = 1$$

and

$$\|h(x, y)\|_\infty = 1$$

By normalising the terms in this way, the coefficient $0 \leq R \leq 1$ can be used to adjust the relative magnitudes of the terms such that the diffused image $m(x, y) \otimes_x \otimes_y p(x, y)$ becomes a perturbation of the 'host image' (covertext) $h(x, y)$. This provides us with a way of digital watermarking [12] one image with another, R being referred to as the 'watermarking ratio', a term that is equivalent, in this application, to the standard term 'Signal-to-Noise' or SNR as used in signal and image analysis. For colour images, the method can be applied by decomposing the image into its constituent Red, Green and Blue components. Stochastic diffusion is then applied to each component separately and the result combined to produce an colour composite image.

For applications in image watermarking, stochastic diffusion has two principal advantages:

- a stochastic field provides uniform diffusion;
- stochastic fields can be computed using random number generators that depend on a single initial value or seed (i.e. a private key).

A. Binary Image Watermarks

Watermarking a full grey level or colour image in another grey or colour image, respectively, using stochastic diffusion leads to two problems: (i) it can yield a degradation in the quality of the reconstruction especially when R is set to a low value which is required when the host image has regions that are homogeneous; (ii) the host image can be corrupted by the watermark leading to distortions that are visually apparent. Points (i) and (ii) lead to an optimisation problem with regard to the fidelity of the watermark and host images in respect of the value of the watermark ratio that can be applied which limits the type of host images that can be used and the fidelity of the 'decrypts'. However, if we consider the plaintext image $p(x, y)$ to be of binary form, then the output of stochastic diffusion can be binarized to give a binary ciphertext. The rationale for imposing this condition is based on considering a system in which a user is interested in covertly communicating documents such as confidential letters and certificates, for example.

If we consider a plaintext image $p(x, y)$ which is a binary array, then stochastic diffusion using a pre-conditioned cipher $0 \leq m(x, y) \leq 1$ consisting of an array of floating point numbers will generate a floating point output. The Shannon Information Entropy of any array $A(x_i, y_i)$ with Probability Mass Function (PMF) $p(z_i)$ is given by

$$I = -\sum_{i=1} p(z_i) \log_2 p(z_i)$$

The information entropy of a binary plaintext image (with PMF consisting of two components whose sum is 1) is therefore significantly less than the information entropy of the ciphertext image. In other words, for a binary plaintext and a non-binary cipher, the ciphertext is data redundant. This provides us with the opportunity of binarizing the ciphertext by applying a threshold T, i.e. if $c_b(x, y)$ is the binary ciphertext, then

$$c_b(x, y) = \begin{cases} 1, & c(x, y) > T \\ 0, & c(x, y) \leq T \end{cases} \tag{2}$$

where $0 \leq c(x,y) \leq 1 \forall x,y$. A digital binary ciphertext image $c_b(x_i, y_j)$ where

$$c_b(x_i, y_i) = \begin{cases} 1, & \text{or} \\ 0, & \text{for any } x_i, y_j \end{cases}$$

can then be used to watermark an 8-bit host image $h(x,y), h \in [0,255]$ by replacing the lowest 1-bit layer with $c_b(x_i, x_j)$. To recover this information, the 1-bit layer is extracted from the image and the result correlated with the digital cipher $n(x_i, y_j)$. Note that the original floating point cipher n is required to recover the plaintext image and that the binary watermark can not therefore be attacked on an exhaustive XOR basis using trial binary ciphers, i.e. binarization of a stochastically diffused data field is entirely irreversible.

B. Statistical Analysis

The expected statistical distribution associated with stochastic diffusion is Gaussian. This can be shown if we consider a binary plaintext image $p_b(x,y)$ to be described by a sum of N delta functions where each delta function describes the location of a non-zero bit at coordinates (x_i, y_j). Thus if

$$p_b(x,y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(x - x_i)\delta(y - y_j)$$

then

$$c(x,y) = m(x,y) \otimes_x \otimes_y p(x,y)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} m(x - x_i, y - y_j).$$

Each function $m(x - x_i, y - y_j)$ is just $m(x,y)$ shifted by x_i, y_j and will thus be identically distributed. Hence, from the Central Limit Theorem

$$\Pr[c(x,y)] = \Pr\left[\sum_{i=1}^{N} \sum_{j=1}^{N} m(x - x_i, y - y_j)\right] =$$

$$\prod_{i=1}^{N}{}_{\otimes} \Pr[m(x,y)] \equiv \Pr[m(x,y)] \otimes_x \otimes_y \Pr[m(x,y)] \otimes_x \otimes_y \ldots$$

$$\sim \text{Gaussian}(z), \quad N \to \infty$$

where Pr denotes the Probability Density Function. We can thus expect $\Pr[c(x,y)]$ to be normally distributed and for $m(x,y) \in [0,1] \forall x,y$ the mode of the distribution will be of the order of 0.5. This result provides a value for the threshold T in equation (2) which for $0 \leq c(x,y) \leq 1$ is 0.5 (theoretically). Note that if $n(x,y)$ is uniformly distributed and thereby represents δ-uncorrelated noise then both the complex spectrum N^* and power spectrum $\mid N \mid^2$ will also be δ-uncorrelated and since

$$m(x,y) = \mathcal{F}_2^{-1}\left[\frac{N^*(k_x, k_y)}{\mid N(k_x, k_y) \mid^2}\right]$$

$\Pr[m(x,y)]$ will be uniformly distributed. Also note that the application of a threshold which is given by the mode of the Gaussian distribution, guarantees that there is no statistical bias associated with any bit in the binary output, at least, on a theoretical basis. On a practical basis, the threshold needs to be computed directly by calculating the mode from the histogram of the cipher and that bit equalization can not be guaranteed as it will depend on: (i) the size of the images used; (ii) the number of bins used to compute the histogram.

C. Principal Algorithms

The principal algorithms associated with the application of stochastic diffusion for binary image watermarking are as follows:

Algorithm I: Encryption and Watermarking Algorithm

Step 1: Read the binary plaintext image from a file and compute the size $I \times J$ of the image.

Step 2: Compute a cipher of size $I \times J$ using a private key and pre-condition the result.

Step 3: Convolve the binary plaintext image with the pre-conditioned cipher and normalise the output.

Step 4: Binarize the output obtained in Step 3 using a threshold based on computing the mode of the Gaussian distributed ciphertext.

Step 5: Insert the binary output obtained in Step 4 into the lowest 1-bit layer of the host image and write the result to a file.

The following points should be noted:

(i) The host image is taken to be an 8-bit or higher grey level image which should ideally be the same size as the plaintext image or else resized accordingly. However, in resizing the host image, its proportions should be the same so that the stegotext image does not appear to be a distorted version of the covertext image. For this purpose, a library of host images should be developed whose dimensions are set according to a predetermined application where the dimensions of the plaintext image are known.

(ii) Pre-conditioning the cipher and the convolution processes are undertaken using a Discrete Fourier Transform (DFT).

(iii) The output given in Step 3 will include negative floating point numbers upon taking the real component of a complex array. The array must be rectified by adding the largest

negative value in the output array to the same array before normalisation.

(iv) For colour host images, the binary ciphertext can be inserted in to one or all of the RGB components. This provides the facility for watermarking the host image with three binary ciphertexts (obtained from three separate binary documents, for example) into a full colour image. In each case, a different key can be used.

(v) The binary plaintext image should have homogeneous margins in order to minimise the effects of ringing due to 'edge-effects' when processing the data in the spectral domain.

Algorithm II: Decryption Algorithm

Step 1: Read the watermarked image from a file and extract the lowest 1-bit layer from the image.

Step 2: Regenerate the (non-preconditioned) cipher using the same key used in Algorithm I.

Step 3: Correlate the cipher with the input obtained in Step 1 and normalise the result.

Step 4: Quantize and format the output from Step 3 and write to a file.

The following points should be noted:

(i) The correlation operation should be undertaken using a DFT.

(ii) For colour images, the data is decomposed into each RGB component and each 1-bit layer is extracted and correlated with the appropriate cipher, i.e. the same cipher or three ciphers relating to three private keys respectively.

(iii) The output obtained in Step 3 has a low dynamic range and therefore requires to be quantized into an 8-bit image based on floating point numbers within the range max(array)-min(array).

Figure 2 shows the result of applying Algorithm I and Algorithm II for a full colour 24-bit image. This example illustrates the embedding of a binary ciphertext into the lowest 1-bit level of a host image which provides a facility to 'hide' information in a host image without impeding its fidelity. In this example, the binary cipher has been inserted into each RGB component. This improves the fidelity of the decrypt which becomes the result of contributions from all three colour channels rather than one channel in the case of a grey level image.

Figure 2. Example of stochastic diffusion applied to a 24-bit colour image (top-left) used to hide an encrypted image of a plaintext image with different font sizes (top-right). The fidelity of the decrypt (bottom-left) is enhanced through the use of a colour host image and watermarking all three colour components with the same watermark. The result can then be post-processed to generate a high quality reconstruction of the original plaintext (bottom-right) which in this case has been generated using a Gaussian lowpass filter with a radius of 1 pixel followed by binarization using a user defined threshold adjusted to give a visually optimal result.

D. 2-bit Randomization

A principal weakness in the algorithms proposed is that the binary watermark is inserted into the lowest 1-bit layer of the host image. The ciphertext can therefore easily be removed and cryptanalysed. A simple solution to this problem is to randomize the watermark over the lowest 2-bit layers of the host image. Thus if $c_b(x_i, x_j)$ is the binary watermark of size $X \times Y$ and $0 \leq r(x_i, y_j) \leq 1$ is an array of uniformly distributed floating point numbers (also of size $X \times Y$) we apply the quantization

$$\forall x_i, y_j \in [X \times Y]$$

$$R(x_i, y_j) = \begin{cases} 0, & 0 \leq r(x_i, y_j) < 0.333; \\ 1, & 0.333 \leq r(x_i, y_j) < 0.666; \\ 2, & 0.666 \leq r(x_i, y_j) < 1; \end{cases}$$

The watermark is then 2-bit randomized by simple array addition to give

$$C(x_i, x_j) = c_b(x_i, x_j) + R(x_i, y_j)$$

The 2-bit array C then replaces the lowest 2-bit layer of the host image. The 1-bit watermark is recovered using array subtraction, i.e.

$$c_b(x_i, x_j) = C(x_i, x_j) - R(x_i, y_j)$$

which requires that r can be reproduced (using a known cipher generating algorithm with a known 'key').

V. StegoCrypt

StegoCrypt has been designed using MATLAB to examine the applications to which stochastic diffusion can be used. It has been designed with a simple Graphical User Interface as shown in Figure 3 whose use is summarised in the following table:

Encryption Mode	**Decryption Mode**
Inputs: Plaintext image Covertext image Private Key (PIN)	*Inputs:* Stegotext image Private key (PIN)
Output: Watermarked Covertext image	*Output:* Decrypted watermark
Operation: Encrypt by clicking on button E (for Encrypt)	*Operation:* Decrypt by clicking on button D (for Dycrypt)

The PIN (Personal Identity Number) is an alpha-numerical

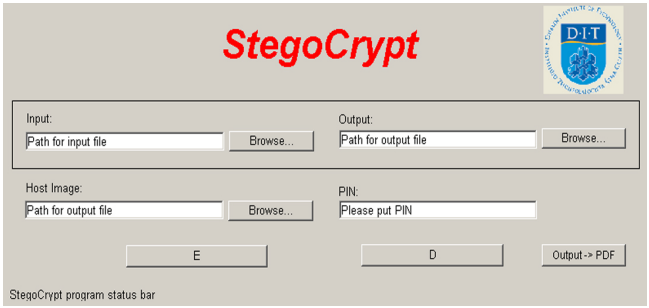

Figure 3. Graphical User Interface for *StegoCrypt* software system.

string with upto 16 elements, generated automatically upon encryption, by default. In principal, any existing encryption algorithm, application or system can be used to generate the cipher required by *StegoCrypt* by encrypting an image composed of random noise. The output then needs to be converted into a decimal integer array and the result normalised as required, i.e. depending on the format of the output that is produced by a given system. In this way, *StegoCrypt* can be used in conjunction with any existing encryption standard.

VI. Example Application of e-Fraud Prevention

The principal aim of the approach described in this paper is to encrypt an image and transform the ciphertext into a binary array which is then used to watermark a host image. This provides a general method for hiding encrypted information in 'image-space'. In this sense, we have developed a covert encryption method for Electronic Data Interchange (EDI)

A. e-Fraud Prevention of e-Certificates

Electronic or e-documents consisting of letters and certificates, for example, are routinely used in EDI. EDI refers to the structured transmission of data between organizations by electronic means. It is used to transfer electronic documents from one computer system to another; from one trading partner to another trading partner, for example [13], [14]. The USA National Institute of Standards and Technology defines EDI as *the computer-to-computer interchange of strictly formatted messages that represent documents other than monetary instruments* [15]. EDI remains the data format used by the vast majority of electronic transactions in the world and EDI documents generally contain the same information that would normally be found in a paper document used for the same organizational function.

In terms of day-to-day applications, EDI relates to the use of transferring documents between two parties in terms of an attachment. For hardcopies, the attachment is typically the result of scanning the document and generating an image which is formatted as a JPEG or PDF (Print Device File) file, for example. This file is then sent as an attachment to an email which typically refers to the attachment, i.e. the email acts as a covering memorandum to the information contained in the attachment. However, a more common approach is to print a document directly to PDF file, for example. Thus, letters written in Microsoft word, for example, can be routinely printed to a PDF file for which there are a variety of systems available, e.g. PDF suite http://pdf-format.com/suite/.

For letters and other documents that contain confidential information, encryption systems are often used to secure the document before it is attached to an email and sent. The method discussed in this paper provides a way of encrypting a document using stochastic diffusion and then hiding the output in an image, thus providing a covert method of transmitting encrypted information. However, the approach can also be used to authenticate a document by using the original document as a 'host image'. In terms of the *StegoCrypt* GUI shown in Figure 3, this involves using the same file for the *Input* and *Host Image*. An example of this is shown in Figure 4 where a hardcopy issue of a certificate has been scanned into electronic form and the result printed to a PDF file. The properties of the image are as follows: File size=3.31Mb; Pixel Dimensions - Width=884 pixels, Height =1312 pixels; Document Size - Width=39.5 cm, Height=46.28cm; Resolution=28 pixels/cm. The result has been encrypted and binarised using stochastic diffusion and the output used to watermark the original document. The fidelity of the decrypt is perfectly adequate to authenticate aspects of the certificate such as the name and qualifications of the holder, the date and signatures, for example. Figure 5 shows the 'Coat of Arms' and the signatures associated with the decrypt given in Figure 4. These results illustrate that the decrypt is adequately resolved for the authentication of the document as a whole. It also illustrates the ability for the decrypt to retain the colour of the original plaintext image.

Figure 4. Certificate with binary watermark (left) and decrypt (right).

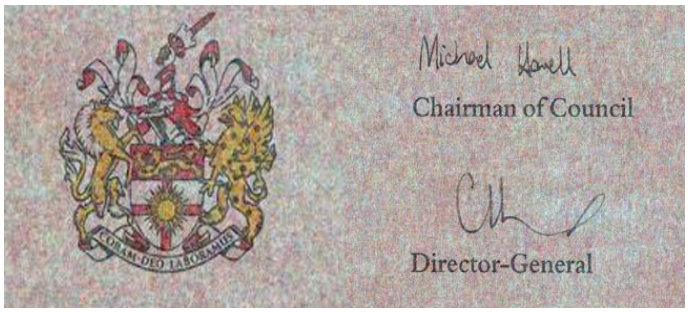

Figure 5. 'Coat of Arms' (left) and signatures (right) of decrypt given in Figure 4.

B. Authentication of Electronic Letters

When a document is scanned, noise is generated and becomes an inherent feature of the image even though it may not be visually intrusive, e.g. the image of the scanned certificate given in Figure 4. Low level scan noise in a digital image of a scanned document with a uniform (typically white) background is of value in camouflaging the existence of a 1-bit or randomized 2-bit based watermark. However, when an image of a letter is generated directly (i.e. printed to a PDF file from Microsoft Word, for example), no noise is generated. This provides a method of revealing the existence, or otherwise, of the watermark. The solution is to add low level noise to the image before inserting the watermark so that it is not clear as to whether the information revealed by statistical inspection of the image is due to the presence of a watermark or background noise. For users of Microsoft word, the solution is to use a texture (*Format→Background→Fill Effect...*) to generate an inhomogeneous background or to include a *Picture Watermark* (*Format→Background→Printed Watermark...*) with *Washout*. By using 'printed watermarks' that have been generated using *texture codes*, an additional level of authentication can be used even if the document is printed [19], [20].

One of the weaknesses of watermarking a letter with itself (a form of self-authentication) is the range of Cribs that are available to an attacker who has extracted the watermark and undertaking cryptanalysis. Thus, ideally, features such as the letter headings and date, for example, should be eliminated from the host image before encryption is undertaken, leaving just the text and important features such as the signature, for example, as agreed by the user(s).

C. Plausible Deniability

Another reason for encrypting plaintext before generating stegotext is to provide a solution that allows a sender to decrypt the data to provide plausible information if forced to do so. With regard to this application, it is possible to develop different encrypted information which is embedded into the host image at additional layers. Thus, in addition to replacing the lowest 1-bit layer of the host image with the binary ciphertext, the next lowest 1-bit layer is replaced with another binary ciphertext. If $c_{b,1}[i][j]$ and $c_{b,2}[i][j]$ are two distinct binary cipheretext images, both consisting of elements that are either 0 or 1, then the first and second 1-bit layers of the covertext image are replaced with $c_{b,1}[i][j]$ and $2 + c_{b,2}[i][j]$ respectively. This process can be repeated for further layers depending on the characteristics of the host image, i.e. the redundancy of pixel values in each 1-bit layer with regard to the fidelity of the stegotext image.

D. Key-Exchange

In order to use *StegoCrypt* effectively the system must be designed with: (i) a cryptographically secure cipher generator; (ii) a key exchange algorithm. With regard to point (ii) there are a range of key exchange algorithms available that can be implemented [16]. A common solution is to utilise the RSA algorithm, not to encrypt plaintext, but to encrypt and transmit the keys used to drive a symmetric encryption system of which *StegoCrypt* is a typical example. RSA based products are available commercially from a range of providers, e.g. http://www.rsa.com/.

VII. Discussion

The use of the internet to transfer documents as image attachments has and continues to grow rapidly as part of a global EDI infrastructure. It is for this 'market' that the approach reported in this paper has been developed. Inserting a binary watermark into a host image obtained by binarizing a floating point ciphertext of a document provides a cryptographically secure solution. Although the watermark can be easily removed from the covertext image - unless 2-bit randomization is implemented as discussed in Section IV(D) - it can not be decrypted without the recipient having access to the correct cryptographically secure algorithm and key. The encrypted watermark is not 'suspicious' especially

when a document has background scan noise, for example, or a background texture has been introduced as discussed in Section VI(B).

The key-exchange approach briefly discussed in Section VII(D) is typical of an infrastructure based on a single sender-receiver scenario in which the use of a system such as *StegoCrypt* is downloaded and installed by both parties or is accessible on-line. In this case, the algorithm used to generate the cipher is the same and the security is based on the PIN which is exchanged using a Public Key Infrastructure. For example, many institutes such as universities still issue 'paper certificates' to their graduates. These certificates are then scanned and sent as attachments along with a CV and covering letter when applying for a job. It is at this point that the certificate may be counterfeited and, for this reason, some establishments still demand originals to be submitted. *StegoCrypt* provides the facility to issue electronic certificates (in addition or in substitution to a hardcopy) which can then be authenticated as discussed in Section VI(A). By including a serial number on each certificate (a Certificate Identity Number) which represents a 'public key', the document can be submitted to the authority that issued the certificate for authentication, for which an online service can be established as required subject to any regulation of investigatory powers e.g. [25].

ACKNOWLEDGMENTS

The authors are grateful for the support of the Science Foundation Ireland and Dublin Institute of Technology.

REFERENCES

[1] S. Katzenbeisser and F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, 2000.

[2] N. F. Johnson, Z. Duric and S. Jajodia, *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academicf Publishers, 2001.

[3] P. R. Marie, *Fractal-Based Models for Internet Traffic and Their Application to Secure Data Transmission*, PhD Thesis, Loughborough University, 2007.

[4] W. R. Harwood, *The Disinformation Cycle: Hoaxes, Delusions, Security Beliefs, and Compulsory Mediocrity*, Xlibris Corporation, 2002.

[5] R. Miniter, *Disinformation*, Regnery Publishing, 2005.

[6] T. Newark and J. F. Borsarello, *Book of Camouflage* Brassey's, 2002.

[7] H. Gerrad and P. D. Antill, *Crete 1941: Germany's Lightning Airborne Assault*, Osprey Publishing, 2005.

[8] R. Canetti, C. Dwork, M. Naor and R. Ostrovsky. *Deniable Encryption*, Theory of Cryptography Library, ePrint Archive 1996.
http://eprint.iacr.org/1996/002

[9] G. Evans, J. M. Blackledge and P. Yardley, *Analytical Methods for Partial Differential Equations*, Springer Undergraduate Mathematics Series, Springer-Verlag, 2000.

[10] M. Bertero and B. Boccacci, *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, 1998.

[11] J. M. Blackledge, *Digital Signal Processing*, Second Edition, Horwood Publishing, 2006.

[12] I. J. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2002.

[13] J. M. Blackledge and K. W. Mahmoud, *Printed Document Authentication using Texture Coding*, ISAST Journal on Electronics and Signal Processing, To be Published, 2009.

[14] Federal Information Processing Standards Publication 161-2, *Electronic Data Interchange (EDI)*, 1996.
http://www.itl.nist.gov/fipspubs/fip161-2.htm

[15] M. Kantor and J. H. Burrows (1996-04-29). Electronic Data Interchange', National Institute of Standards and Technology, 1996 http://www.itl.nist.gov/fipspubs/fip161-2.htm.

[16] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley, 1996, ISBN: 0471117099.

[17] J. M. Blackledge, *Digital Image Processing*, Horwood Publishing, 2005, ISBN 1-898563-49-7, http://eleceng.dit.ie/papers/103.pdf.

[18] J. M. Blackledge, *Multi-algorithmic Cryptography using Deterministic Chaos with Applications to Mobile Communications*, ISAST Transactions on Electronics and Signal Processing, **2**(1), 23 - 64, 2008, ISSN 1797-2329.

[19] J. M. Blackledge, *Information Hiding by Stochastic Diffusion and its Application to Printed Document Authentication*, Irish Signals and Systems Conference, University College Dublin, June 10-11th 2009; http://eleceng.dit.ie/papers/116.pdf

[20] J. M. Blackledge and K. W. Mahmoud, *Printed Document Authentication using Texture Coding*, ISAST Transaction on Electronics and Signal Processing (To be Published), 2009; http://eleceng.dit.ie/papers/135.pdf

[21] D. Knuth, *The Art of Computer Programming: Volume 2, Seminumerical Algorithms*, Second Edition, Addison-Wesley, 1981.

[22] N. Hahnfield, *Cryptography Tutorial: RSA*, 2001;
http://www.antilles.k12.vi.us/math/cryptotut/rsa1.htm.

[23] J. Buchmann, *Introduction to Cryptography*, Springer 2001.

[24] O. Goldreich, *Foundations of Cryptography*, Cambridge University Press, 2001.

[25] Office of Public Sector Information, *Regulation of Investigatory Powers Act 2000*, 2000 CHAPTER 23.
http://www.opsi.gov.uk/acts/acts2000/ukpga_20000023_en_1