2008-01-01

Covert Encryption and Document Authentication using Texture Coding

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

Mary Hallot
University of the Western Cape, RSA

Recommended Citation

Blackledge, J., Hallot, M.: Covert Encryption and Document Authentication using Texture Coding. Journal of Software Engineering. vol: 3, issue: 1, pages: 45-65, 2008. doi:10.21427/D7BK7T

COVERT ENCRYPTION AND DOCUMENT AUTHENTICATION USING TEXTURE CODING

By

J. M. BLACKLEDGE* M. L. HALLOT**

ABSTRACT

With the improvements in the quality of Commercial-Off-The-Shelf (COTS) printing and scanning devices, the ability to counterfeit documents has become a widespread problem. Consequently, there has been an increasing demand to develop digital watermarking techniques which can be applied to both electronic and printed images (and documents) that can be authenticated, prevent unauthorized copying and withstand abuse and degradation. In this paper, a new approach to digital watermarking is presented and a range of possible applications are considered. The process is defined by using concepts and techniques borrowed from Cryptography. It is based on computing a 'scramble image' by diffusing a watermark image with a noise field (a cipher). The cover image (covertext) is then introduced using a simple additive process (confusion). The watermark is subsequently recovered by removing the covertext and then correlating the output with the original (key dependent) noise field. For covert encryption, this approach provides the user with a method of hiding ciphertext (the scrambled image) in a host image before the transmission of the data. With regard to document authentication, a diffusion only or 'texture coding' approach that is robust to a wide variety of attacks including geometric attacks, crumpling and print/scan attacks are considered.

Keywords: Cryptography, Steganography, Watermarking, Information Security, Document Authentication.

INTRODUCTION

In this paper, a new approach to digital watermarking is presented and a range of possible applications are considered. The process is defined using concepts borrowed from Cryptography. It is based on computing a 'scrambled image' by diffusing a watermark image with a noise field (a cipher). The cover image (cover or covertext) is then introduced using a simple additive process (confusion). The watermark is subsequently recovered by removing the covertext and then correlating the output with the original (key dependent) noise field. For covert encryption, this approach provides the user with a method of hiding ciphertexts (the scrambled image) in a host image before the transmission of the data. In this sense, it provides a steganographic approach to cryptography in which the ciphertext is not apparent during an intercept. Decryption is based on the knowledge of the key and access to the host image. In terms of watermarking a digital image, the method provides a way of embedding information in an image that can be used for authentication from an identifiable source, a method that is relatively insensitive to lossy compression, making it well suited to digital image transmission. With regard to document authentication, the use of diffusion and confusion using a covertext is not robust. The reason for this is that the registration of pixels associated with a covertext can not be assured when the composite image is printed and scanned. Therefore a diffusion only approach is considered to document authentication which is robust to a wide variety of 'attacks' including geometric attacks, drawing, crumpling and print/scan attacks. This is because the process of diffusion (i.e. the convolution of information) is compatible with the physical principles of an imaging system and the theory of image formation. Thus, it is consistent with the image capture devices (digital cameras and scanners, for example) that, by default, conform to the 'physics' of optical image formation.

The diffusion of plaintext (in this case, an image) with a noise field (the cipher) has a synergy with the encryption of plaintext using a cipher and an XOR operation (when both the plaintext and cipher are represented by binary streams). However, decryption of a convolved image (deconvolution) is not as simple as XORing the ciphertext with the appropriate cipher. Here, an approach which is based on pre-conditioning the original cipher is considered in such a way that decryption (de-diffusion) can be undertaken by simply correlating the ciphertext with the cipher. If a high entropy cipher is used that is uniformly distributed, then the Power Spectral Density Function (PSDF) of the output will be determined by the PSDF of the plaintext (image). If the image is based on naturally occurring objects which are roughly of a self-affine type, then the PSDF may tend to scale as $| k |^{-q}$ where k is the spatial frequency and q is a measure of the Fractal Dimension. In other words, the diffusion of self-affine images with white noise will generate output images that are, in effect, random fractal images with fractal-type textures. In this sense, the use of white noise diffusion for document authentication is based on using texture maps which are either fully or partially fractal. Either way, the outputs considered for document authentication are based on printing textures (texture coding) of a type that are determined by the spectral characteristics of the plaintext which can be applied using low resolution COTS printers and scanners.

1. Basic Concepts in Cryptography

Irrespective of the wealth of computational techniques that can be invented to encrypt data, there are some basic concepts that are a common theme in modern cryptography. The application of these concepts typically involves the use of random number generators and/or the use of algorithms that originally evolved for the generation of random number streams, algorithms that are dominated by two fundamental and interrelated themes [1], [2], [3]: (i) the use of modular arithmetic; (ii) the application of prime numbers. The application of prime numbers is absolutely fundamental to a large range of encryption processes and international standards such as PKI (Public Key Infrastructure). Using a traditional paradigm, the problem of how Alice (A) and Bob (B) can pass a message to and from each other without it being compromised or 'attacked' by an intercept is considered. As illustrated in Figure 1, a simple box and a combination lock scenario is considered. Alice and Bob can write a message, place it in a box, lock the box and then send it through an open 'channel' - the postal services, for example.

In cryptography, the strength of the box is analogous to the strength of the cipher. If the box is 'weak' enough to be opened by brute force, then the strength of the lock is relatively insignificant. This is analogous to a cipher whose statistical properties are poor, i.e. whose Probability Density Function (PDF) is narrow and whose information Entropy is relatively low, with a similar value to the plaintext. The strength of the lock is analogous to the strength of the key in a real cryptographic system. This includes the size of the combination number which is equivalent to the length of the key that is used. Clearly a four rotor combination lock as illustrated in Figure 1 represents a very weak key since the number of ordered combinations required to attempt a brute force attack to open the lock are relatively low, i.e. for a 4-digit combination lock where each rotor has ten digits 0-9, the number of possible combinations is 10000 (including 0000). However, the box-and-lock paradigm being used here is for illustrative purposes only.

1.1 Symmetric Encryption

Symmetric encryption is the simplest and most obvious approach to Alice and Bob sending their messages. Alice and Bob agree on a combination number *a priori*. Alice writes a message, puts it in a box, locks it and sends it off. Upon receipt, Bob unlocks the box using the combination

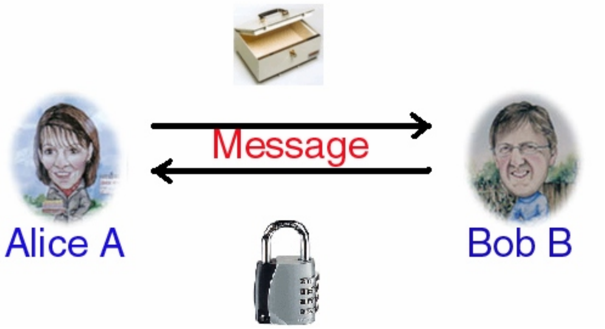


Figure 1. Alice and Bob can place a message in a box which can be secured using a combination lock and sent via a public network - the postal service, for example.

number that has been agreed and recovers the message. Similarly, Bob can send a message to Alice using exactly the same approach or 'protocol'. Since this protocol is exactly the same for Alice and Bob it has symmetry and thus, encryption methods that adopt this protocol are referred to as symmetric encryption methods.

Given that the box and the lock have been designed to be strong, the principal weakness associated with this method is its vulnerability to attack if a third party obtains the combination number at the point when Alice and Bob invent it and agree upon it. Thus, the principal problem in symmetric encryption is how Alice and Bob exchange the key. Irrespective of how strong the cipher and key are, unless the key exchange problem can be solved in an appropriate and a practicable way, symmetric encryption always suffers from the same fundamental problem - key exchange!

If E denotes the encryption algorithm that is used which depends upon a key K to encrypt a plaintext P, then the ciphertext C to be given by C = EK(P) can be considered. Decryption can then be denoted by the equation P = EK(C). Note that it is possible to encrypt a number of times using different keys K1, K2, ... with the same encryption algorithm to give a double encrypted cipher text C = EK2 (EK1 (P)) or a triple encrypted ciphertext C = EK3 (EK2 (EK1 (P))). Decryption, is then undertaken using the same keys in the reverse order to which they have been applied, i.e. P = EK1 (EK2 (EK3 (C))). Symmetric encryption systems, which are also referred to as shared secret systems or private key systems, are usually significantly easier to use than systems that employ different protocols (such as asymmetric encryption). However, the requirements and methods associated with key exchange sometimes make symmetric systems difficult to use.

Examples of symmetric encryption systems include the Digital Encryption Standard DES and DES3 (essentially, but not literally, the Digital Encryption Standard with triple encryption) and the Advanced Encryption Standard (AES). Symmetric systems are commonly used in many banking and other financial institutes and in some military applications.

1.2 Asymmetric Encryption

Instead of Alice and Bob agreeing on a combination number a priori, suppose that Alice sets her lock to be open with a combination number known only to her. If Bob then wishes to send Alice a message, he can make a request to her to send him an open lock. Bob can then write his message, place it in the box which is then locked and sent on to Alice. Alice can then unlock the box and recover the message using the combination number known only to her. The point here is that Bob does not need to know the combination number, he only needs to receive an open lock from Alice. Of course Bob can undertake exactly the same procedure in order to receive a message from Alice. Clearly, the processes that are undertaken by Alice and Bob in order to send and receive a single message are not the same. The protocol is asymmetric and we refer to encryption systems that use this protocol as being asymmetric. Note that Alice could use this protocol to receive messages from any number of senders provided they can get access to one of her open locks. This can be achieved by Alice distributing many such locks as required. One of the principal weaknesses of this approach relates to the lock being obtained by a third party whose interest is in sending bogus or disinformation to Alice.

The problem for Alice is to find a way of validating that a message sent from Bob (or anyone else who is entitled to send messages to her) is genuine, i.e. the message is authentic. Thus, data authentication becomes of particular importance when implementing asymmetric encryption systems. Asymmetric encryption relies on both parties having two keys. The first key (the public key) is shared publicly. The second key is private, and is kept secret. When working with asymmetric cryptography, the message is encrypted using the recipients' public key. The recipient then decrypts the message using the private key. Because asymmetric ciphers tend to be computationally intensive (compared to symmetric encryption), they are usually used in combination with symmetric systems to implement public key cryptography. Asymmetric encryption is often used to transfer a session key rather than information proper - plaintext. The session key is then used to encrypt information using a symmetric encryption system. This gives the key exchange benefits of asymmetric encryption with the speed of symmetric encryption.

A well known example of asymmetric encryption - also known as public key cryptography - is the RSA algorithm. This algorithm uses specific prime numbers (from which the private and public keys are composed) in order to realize the protocol. To provide users with such prime numbers, an infrastructure needs to be established by a third party whose 'business' is to distribute the public/private key pairs. This infrastructure is known as the Public Key Infrastructure or PKI. The use of a public key is convenient for those who wish to communicate with more than one individual and is thus, a many-to-one protocol that avoids multiple key-exchange. On the other hand, a public key provides a basis for cryptanalysis. Given that C = EK(P), where K is the public key, the analyst can guess P and check the answer by comparing C with the intercepted ciphertext, a guess that is made easier if it is based on a known Crib - i.e. information that can be assumed to be a likely component of the plaintext. Public key algorithms are therefore often designed to resist chosen plaintext attacks. Nevertheless, analysis of public key and asymmetric systems in general, reveals that the level of security is not as significant as that which can be achieved using a well-designed symmetric system. One obvious and fundamental issue relates to the third party responsible for the PKI and how much trust should be assumed, especially with regard to legislation concerning issues associated with the use of encrypted material.

1.3 Public-Private Key Encryption

Public-Private Key Encryption [4], [5] is fundamentally asymmetric and in terms of the box and combination-lock paradigm (Figure 1) is based on considering a lock which has two combinations, one to open the lock and another to lock it. The second constraint is the essential feature because one of the basic assumptions in the use of combination locks is that they can be locked irrespective of the rotor positions. Thus, after writing a message, Alice uses one of Bobs specially designed locks to lock the box using a combination number that is unique to Bob but is openly accessible to Alice and others who want to send Bob a message. This combination number is equivalent to the public key. Upon reception, Bob can open the lock using a combination number that is known only to himself - equivalent to a private key. However, to design such a lock, there must be some mechanical 'property' linking the combination numbers required to first lock it and then unlock it. It is this property that is the principal vulnerability associated with public/private key encryption, a property that is concerned with certain precise and exact relationships that are unique to the use of prime numbers and their applications with regard to generating pseudo random number streams and stochastic functions in general [6].

2. Steganography

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. In Figure 1, for example, if a postal worker observed a locked box passing through the post office, it would be natural for them to wonder what might be inside. It would also be natural to assume that the contents of the box would have a value in proportion with the strength of the box/lock. These aspects 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 in 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 obscure [7], [8]. This provides a significant advantage over cryptography alone is 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. With reference to Figure 1, Steganography is equivalent to transforming the 'strong box' into some other object that will pass through without being noticed an 'egg-box', for example. The word 'Steganography' is of Greek origin. It means 'covered' or 'hidden writing'.

In general, a steganographic message appears as something else known as a covertext. By way of a simple illustrative example, suppose we want to transmit the phrase *'The Queen likes horses'* which is encrypted to produce the cipher stream *'syoahfsuyTebhsiaulemNG'*. This is clearly a scrambled version of a message with no apparent meaning to the order of the letters from which it is composed. Thus, it is typical of an intercept that might be attacked because of the very nature of its incomprehensibility. However, suppose that the cipher stream above could be re-cast to produce the phrase *'Beware of Greeks bearing gifts'*. If this phrase is intercepted it may not be immediately obvious that there is an alternative information associated with such an apparently innocuous message i.e. if intercepted, it is not clear whether or not it is worth initiating an attack.

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:

$$\begin{tikzcd}
 & \text{Data} \arrow[r] & \text{Covertext} \arrow[d, dash] & \\
\text{Plaintext} \arrow[r] & \text{Ciphertext} \arrow[r] & \text{Stegotext} \arrow[r] & \text{Transmission}
\end{tikzcd}$$

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. A simple approach is to use a mask to delete all characters in a message except those that are to be read by the recipient of the message. For example, consider the following message: *At what time should I confirm our activities? kindly acknowledge*. This seemingly innocent plaintext could be used to hide the message *'Attack now'* through application of the following mask1100001000000000000000000000000000000110000 0000001000001000111000000 where 0 denotes that a character or space is to be ignored and 1 denotes that a character or space is used.

Apart from establishing a method of exchanging the mask which is equivalent to the key in cryptography, the principal problem with this approach is that different messages have to be continuously 'invented' in order to accommodate hidden messages and that these 'inventions' must appear to be legitimate. However, the wealth of data that is generated and transmitted in today's environment and the wide variety of formats that are used means that there is much greater potential for exploiting steganographic methods than were available before the IT revolution. In other words, the IT revolution 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 techniques 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 stegotext.

2.1 Hiding Data in Images

The relatively large amount of data contained in digital images makes them a good 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 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, the grey level value 128 has the binary representation 10000000. If the least significant bit is changed to give 10000001 (which corresponds to a grey level value of 129) then the difference in the output image will not be discernable. Hence, the least significant bit can be used to encode information other than pixel intensity. Further, if this is done for each colour component then a letter of ASCII text can be represented for every three pixels. The larger the host image compared with the hidden message, the more difficult it is to detect the message. Further, 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 2 shows the effect of hiding one image in another through the process of re-quantization and addition. The image to be embedded is re-quantised 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. 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 2 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

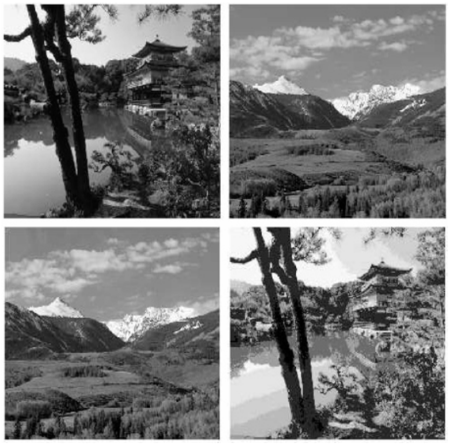


Figure 2. Example of 'hiding' one image (top left) in another image (top-right) through simple re-quantization and addition (bottom left). By subtracting the bottom left image from the top right image and re-quantising the output, the bottom right reconstruction is obtained.

algorithm from which the stream can be regenerate (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 a decrypt to be accomplished. However, by diffusing the image with a noise field, it is possible to hide the output in a host image without having to resort to quantization.

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. This is equivalent to hiding a 2-bit image in a host image as illustrated in Figure 3 which uses the same method as discussed above. In this example, a columnar transposition cipher has been used to encrypt this sentence using the keyword: *Steganography.*

As in the previous example, the host image is required to recover the ciphertext information and is thus the 'key' to the process. The methods discussed above refer to electronic-to-electronic type communications in which there is no loss of information. 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 which has only relatively recently been disclosed.

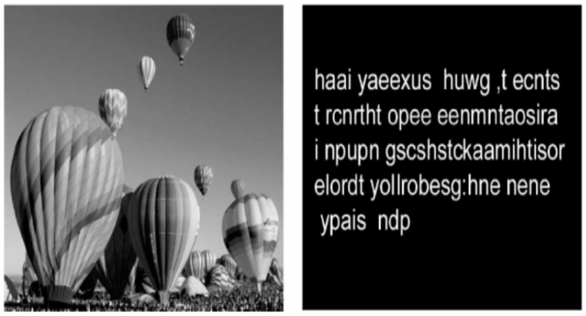

haai yaeexus huwg ,t ecnts
t rcnrtht opee eenmntaosira
i npupn gscshstckaamihtisor
elordt yollrobesg:hne nene
ypais ndp

Figure 3. Binary image of encrypted information (right), obtained by subtraction of the covertext image from the stegotext image (left).

2.2 Hiding Information in Noise

The art of steganography is to use what ever covertext is readily available to make the detection of plaintext or, ideally, the ciphertext as difficult as possible. This means that the embedding method used to introduce the plaintext/cipherext into the covertext should produce a stegotext that is indistinguishable from the covertext in terms of its statistical characteristics and/or the information it conveys. From an information theoretic point of view, this means that the covertext should have significantly more capacity than the cipheretext, i.e. there must be a high level of redundancy. Utilizing noisy environments often provides an effective solution to this problem. There are three approaches that can be considered: (i) embedding the ciphertext in real noise; (ii) transforming the ciphertext into noise that is then added to data; (iii) replacing real noise with ciphertext that has been transformed into synthetic noise with exactly the same properties as the real noise. In the first case, we can make use of noise sources such as thermal noise, flicker noise and shot noise associated with electronics that digitize an analogue signal. In digital imaging this may be noise from the imaging charge couple device (CCD) element; for digital audio, it may be noise associated with the recording techniques used or amplification equipment.

Natural noise generated in electronic equipment usually provides enough variation in the captured digital information that it can be exploited as a noise source to 'cover' hidden data. Because such noise is usually a linear combination of different noise types generated by different physical mechanisms, it is usually characterized by a normal or Gaussian distribution as a result of the Central Limit Theorem. In the second case, the ciphertext is transformed into noise whose properties are consistent with the noise that is to be expected in certain data fields.

For example, lossy compression schemes (such as JPEG - Joint Photographic Expert Group) always introduce some error (numerical error) into the decompressed data and this can be exploited for steganographic purposes. By taking a clean image and adding ciphertext noise to it, information can be transmitted covertly providing all users of the image assume that it is the output of a JPEG or some other lossy compressor. Of course, if such an image is JPEG compressed, then the covert information may be badly corrupted. In the third case, we are required to analyse real noise and derive an algorithm for its synthesis. Here, the noise has to be carefully synthesized because it may be readily observable as it represents the data stream in its entirety rather than data that is 'cloaked' in natural noise. This technique also requires that the reconstruction/decryption method is robust in the presence of real noise that we should assume will be added to the synthesized noise during a transmission phase. In this case, random fractal models are of value because the spectral properties of many noise types found in nature signify fractal properties to a good approximation. This includes transmission noise over a range of radio and microwave spectra, for example, and Internet traffic noise.

3. Watermarking using Noise Diffusion and Covertext Confusion

An approach to watermarking plaintext using both diffusion and confusion has been considered. The basic approach is as follows: Given a plaintext image and a covertext image, the stegotext image is given by,

$$Stegotext=ciphertext+covertext$$

where,

$$Ciphertext = cipher \ast plaintext$$

Where $\ast$ denotes the (two-dimensional) diffusion process which is based on a convolution operation. The problem is to find a cipher which provides a ciphertext that, given the equations above, can be covertly embedded in the covertext while allowing the plaintext to be recovered. Let the cipher be some pre-defined noise field denoted by n and let p denote the plaintext and c denote the ciphertext so that $c= n \ast p$. There are two approaches to solving this problem, i.e. obtaining a solution for p given c and n. We can invert or deconvolve c through the use of appropriate regularization methods, e.g. the constrained deconvolution filter. Alternatively, if n is the result of some random number generating algorithm, and since the functional form of n is arbitrary, a stochastic field given by $m=F\ [N/|N|^2]$ can be constructed where F denotes the (inverse Fourier transform operator) and N denotes the Fourier transform of n. The ciphertext is then given by $c= m \ast p$ and the plaintext can be recovered by correlating c with n, since from the correlation theorem, in Fourier space, $N^*C=N^*NP/|N|^2=P$ where C and P denote the Fourier transform of c and p respectively (N^* denotes the complex conjugate of N). The necessary condition that $|N|^2 > 0$ is simply achieved by regularizing m such that if $|N|^2$ is zero for any array value, then that array value is set to 1. This approach can be used to 'embed' one data field in another as described below.

We consider the case when we have two independent images p (plaintext) and v (covertext) both of which are normalized, i.e. they are floating point arrays with pixel values between 0 and 1 inclusively. A noise field m *a priori* is constructed and an output (the stegotext) given by $s = r(m \ast p) + v$ is generated where $0<r<1$. By normalizing the images p and v, the coefficient r can be used to adjust the relative magnitudes of the terms such that the diffused image p is a perturbation of the 'host image' v. This provides us with a way of watermarking one image with another, r being referred to as the watermarking ratio (equivalent, in this application, to the standard term 'Signal-to-Noise' or SNR ratio as used in signal and image analysis).

For applications in image watermarking, the diffusion of an image with noise is used because: (i) a noise field provides uniform diffusion; (ii) noise fields can be generated using random number generators that depend on a single initial value or seed (i.e. a private key). The noise field n (which should be uniformly distributed) can be created using any number of random number generators designed independently or through application of various commercial cryptosystems. In each case, the algorithm used provides an output that is key dependent and thus, given a known algorithm, reconstruction of the watermark is achieved with knowledge of the private key together with the host image. An example of this approach is shown in Figure 4. Here, the image v (the 'host image' or covertext) is watermarked by another image p (the 'watermark image' or plaintext) to produce an output image s (stegotext) with $r=0.01$. The relatively small perturbation of the term $r(m * p)$ to the host image v for $r=0.01$ does not affect the output image in any way that is visually significant.

A further advantage of noise diffusion is that it is not limited to watermarking covertexts with binary image plaintexts. However, the effect of adding a diffused greyscale watermark image to the host image yields a different, slightly brighter image because of the perturbation of v by $r(m * p)$. This effect can be minimized by introducing a smaller watermarking ratio such that the perturbation is still recoverable by subtracting the host image from the watermarked image, an example being given in Figure 5. For the purpose of further quantifying the basic algorithms involved in this method of watermarking the reader is referred to the pseudo-coded (void) functions given in the Appendix.

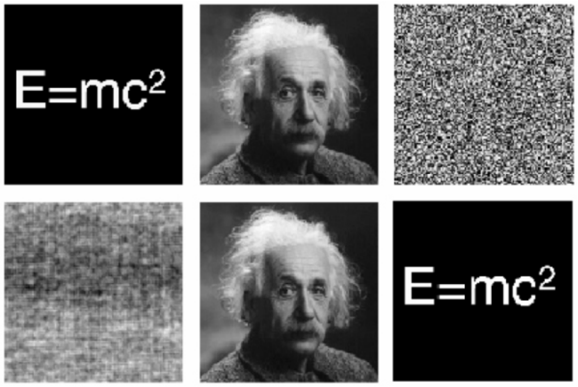


Figure 4. From top to bottom and from left to right: Watermark p, host image v, noise field m, diffused image $m*p$, host image after watermarking $r=0.01$, reconstruction of watermark.

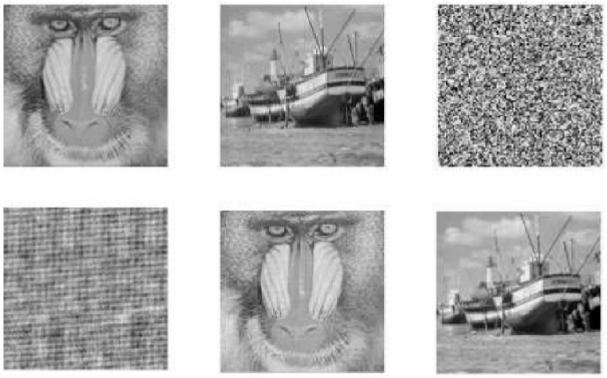


Figure 5. Example of watermarking an image with another image using noise based diffusion. The 'host image' v (top-left) is watermarked with the 'watermark image' p (top-centre) using the diffuser (top-right) given by a uniform noise field n whose pixel-by-pixel values depend upon the seed used (the private key). The result of computing $r(m * p)$ (bottom-left) is added to the host image for $r=0.1$ to generate the watermarked image s (bottom-centre). Recovery of the watermark image (bottom-right) is accomplished by subtracting the host image from the watermarked image and correlating the result with the noise field n.

4. Hardcopy Steganography and Document Security

The use of the model *stegotext = ciphertext + covertext* can be applied for watermarking digital images associated with electronic-to-electronic type communications in which there is no loss of information. This method can be used to watermark digital images for the purpose of authentication but can also be viewed as a method of covertly transmitting ciphertext when the plaintext is converted to the form of a digital image. Steganography and watermarking techniques are also of value for hardcopy 'data' which has a range of applications for authenticating printed material and copyright validation, for example. However, to be of practical value to the security printing industry the methods must be robust to the significant distortions generated by the printing and/or scanning process.

4.1 Diffusion Only Watermarking

If a stegotext image is printed and scanned back into electronic form, then the print/scan process will yield an array of pixels that will be significantly different from the original electronic image even though it might 'look' the same. These differences can include the size of the image, its orientation, brightness, contrast and so on. Of all the processes involved in the recovery of the watermark, the subtraction of the host image from the watermarked image is critical. If this process is not accurate on a pixel-by-pixel basis and deregistered for any of many reasons, then recovery of the watermark by correlation will not be effective. However, if we make use of the diffusion process alone, then the watermark can be recovered via a print/scan because of the compatibility of the processes involved. In this case, the 'watermark' is not covert but overt.

Depending on the printing process applied, a number of distortions will occur which diffuse the information being printed. Thus, in general, the printing process can be considered to introduce an effect that can be modelled by the function $p*f$ where f is the original electronic form of a diffused image and p is the Point Spread Function (PSF) characteristic of the printed material. An incoherent image of the data, obtained using a flat bed scanner, for example (or any other incoherent optical imaging system), will have a similar effect. Thus, a scanned image to be given by $s*p*f$ can be considered where s is the PSF of the scan process that generates the digital output. Since convolution is a linear and commutative process, then, if $f=m*u$ where u is the original image (watermark) and n is the noise field, we can write $s*p*m*u=m*s*p*u$. The operator $s*p*$ represents the data transformation operator associated with the processing cycle of printing the image and then scanning it back into electronic form. By applying the method discussed earlier and correlating the data with the appropriate noise field n, we can thus obtain a reconstruction of the watermark w whose fidelity is determined by the scan/print characteristics, i.e. $w\sim s*p*u$.

The size of any image captured by a scanner or other device will depend on the resolution used. The size of the image obtained will inevitably be different from the original because of the resolution and window size used to print the diffused image f and the resolution used to scan the image. Since scaling in the spatial domain causes inverse scaling in the Fourier domain, the scaling effect must be 'inverted' before the watermark can be recovered by correlation since correlation is not a scale invariant process. Re-sizing the image (using an appropriate interpolation scheme such as the bi-cubic method, for example) requires a set of two numbers a and b (i.e. the a x b array used to generate the noise field and execute the diffusion process) that, along with the seed required to regenerate the noise field, provides the 'private keys' needed to recover the data from the diffused image. An example of this approach is given in Figure 6 which shows the result of reconstructing four different images (a photograph, finger-print, signature and text) used in the design of an impersonalized debit/credit card.

The use of 'diffusion only' watermarking for print security can be undertaken in colour by applying exactly the same diffusion/reconstruction methods to the red, green and blue components independently. This provides two additional advantages: (i) the effect of using colour tends to yield better quality reconstructions because of the colour combination process; (ii) for each colour component, it is possible to apply a noise field with a different seed. In this case, three keys are required to recover the watermark although it should be noted that, due to the errors associated in the extraction of each colour component from a colour scan, this approach does not yield reconstructions with the same degree of robustness as in the case when the same key/algorithm is used for each colour component.

Because this method is based on convolution alone, the recovery of the function u will not be negated by the distortion of the PSF associated with the print/scan process, just limited or otherwise by its characteristics. Thus, if an image is obtained of the printed data field

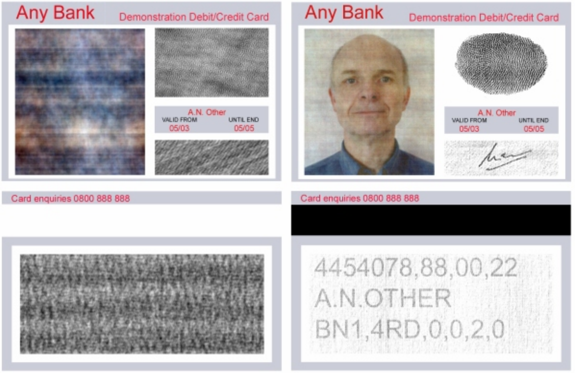


Figure 6. Example application of 'diffusion only' watermarking. In this example, four images of a face, finger-print, signature and text have been diffused using the same cipher and printed on the front (top-left) and back (bottom-left) of an impersonalized identity card using a 600 dpi printer. The reconstructions (top-right and bottom-right, respectively) are obtained using a conventional flat-bed scanner based on a 300 dpi colour or grey-level scan.

which is out of focus due to the characteristics of print/scan process, then the reconstruction of the watermark will be out of focus to the same degree. Decryption of images with this characteristic is only possible using an encryption scheme that is based on a diffusion only approach.

Uniform noise diffusion of the type illustrated in Figure 6 can be used to replace complex print security features with the added advantage that, by de-diffusing them, information can be recovered. Further, these fields are very robust to data degradation created by soiling, for example.

In the case of binary watermark images, data redundancy allows reconstructions to be generated from a binary output, i.e. after binarizing the diffusion field (with a threshold of 50%, for example). This allows the output to be transmitted in a form that can tolerate low resolution and low contrast copying, e.g. a fax. The tolerance of this method to printing and scanning is excellent provided the output is cropped accurately (to within a few pixels) and oriented correctly. The processes of cropping and orientation can be enhanced and automated by providing a reference frame in which the diffused image is inserted. This is illustrated in Figure 7 which, in addition, shows the effect of diffusing a combination of images.

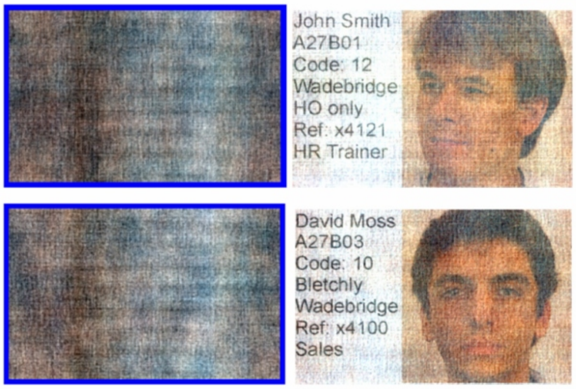


Figure 7. Composite images that include a (blue) reference frame for automatic cropping and orientation. In each case, the data fields (left hand side) have been printed and scanned at 300 dpi. The reconstructions on the right-hand side are the outputs obtained after re-sizing the scanned data to the size of the noise field used for encrypting the original images and then correlating the (re-sized) data with the noise field.

This has the effect of producing a diffused field that is very similar but nevertheless conveys entirely different information.

4. 2 Covertext Addition and Removal

Because diffusion only watermarking is based on convolution/correlation operations it is relatively insensitive to contrast stretching and compression. This provides the opportunity to introduce covertext in the form of the addition of foreground information (e.g. text) to a printed document that has been watermarked *a priori* with a texture map whose brightness and contrast has been adjusted to be unobtrusive with regard to the covertext (i.e. the watermark is made bright compared to black text). Alternative, once the texture field has been designed, it may be introduced into a text editor that provides the inclusion of watermarks. For example, Microsoft Word has the facility to include a printed watermark (Format - Background - Printed Watermark) that provides the option to select a Picture Watermark (Existing Watermark) with options on scale and 'Washout'.

In order to extract the watermark, it is then necessary to remove the text after a scan has been undertaken under the assumption that the covertext is not available. This can be accomplished using a median filter which is effective in removing isolated noise spikes, i.e. in this application, foreground text. However, in this case, the median filter is not applied to the image in its entirety. Instead, it is applied only to the neighbourhood of pixels (i.e. a user defined moving window) that exist below a user defined threshold that is specified in order to differentiate between the watermark and those pixels associated with the covertext. After the removal of the covertext, the image watermark is reconstructed by correlation with the cipher.

4. 3 Covert Document Watermarking using Diffusion

Watermarking is usually considered to be a method in which the watermark is embedded into a host image in an unobtrusive way. Another approach is to consider the host image to be a data field that, when processed with another data field, generates new information. Consider two images given by p and v. Suppose we construct the following function in Fourier space: $N = PV/|P|^2$ where P is the Fourier transform of p, V is the Fourier transform of v and N denotes the Fourier transform of n. If is correlated n with p, then from the correlation theorem, in Fourier space, this is equivalent to computing $P^*PV/|P|^2 = P$. In other words, we can recover v from p with knowledge of n. This process is based on convolution and correlation alone, and it is therefore compatible and robust to printing and scanning, i.e. incoherent optical imaging. An example of this approach is given in Figure 8. In this case, the noise field n is the private key required to reconstruct the watermark and the host image can be considered to be a public key.

5. Example Applications of Texture Coding

Some applications of diffusion only coding are already evident from the examples already given to introduce the technique in previous sections. Strictly speaking, the method is not a watermarking technique unless a covertext can be used to hide the ciphertext. However, as discussed previously, application of a covertext is not applicable for the authentication of printed documents due to the degradation of the covertext when printing and scanning a document. Hence, for applications to low resolution print security, we refer to the method as texture coding. In this section, a range of applications to which the method can be applied are considered.

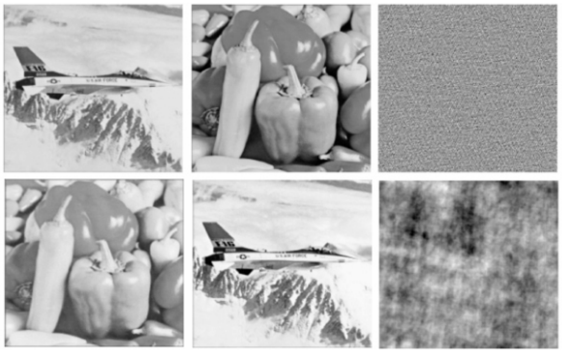


Figure 8. An example of covert watermarking. The image p (top-left) is 'processed' with image v (top-middle) to produce the noise field (top-right). Image v is printed at 600 dpi, scanned at 300 dpi and then re-sampled back to its original size (bottom-left). Correlating this image with the noise field given generates the reconstruction (bottom-centre). The reconstruction depends on just the host image and noise field. If the noise field and/or the host image are different or corrupted, then a reconstruction is not achieved, as in the example given (bottom-right).

5. 1 Authentication

Authentication of a document image should ensure that the document has not been altered from the time it is created and signed by the author to the time it is received at the destination. Authentication of paper documents is an important concern as the ability of counterfeiters has increased substantially in recent years. This is contributed by the dramatic improvement in the capability of high resolution scanners and printers. Moreover, digital documents can be accessed and modified by intruders relatively easily. This is especially true in the case of documents that are exchanged over the Internet. Using the model and methods discussed here, a selective authentication approach can be applied in which only significant changes cause authentication to fail. This can be verified by embedding information in a document that can later be verified as to whether it has been tampered with or otherwise.

5. 2 Photo Verification

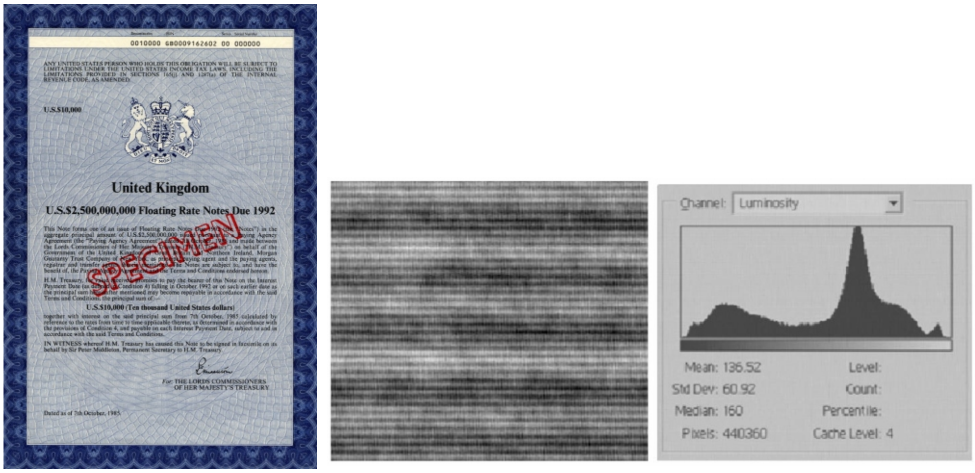
Figure 7 shows an example of a photo verification system that can be incorporated into an ID card where a photograph of the card holder is texture coded and printed beside the original image. Substantial editing, such as changing the original photo, will be illegitimate because it will completely change the interpretation of the card. Thus, a photo verification system can be designed to do the following: (i) capture the diffused watermark using any tool (scanner, camera, etc); (ii) read the key that could be, for example, encoded using a bar code, stored in a local database or stored in a distance database that can be accessed via the Internet; (iii) extract the watermark; (iv) verify the authenticity by comparing the original photo with the extracted one. Verification can be done either by: (i) application of a subjective test, using the judgment of human beings (details on the scales that have been suggested for use in evaluation of watermarking quality being given in [9], for example); (ii) quality metrics, such as the Mean Square Error or Chi-square test; (iii) any other matching algorithm including the application of an Artificial Neural Network as required. Such a system can be modified to include more information in the diffused watermark as required, such as the name of the ID card holder. Moreover, texture coding can be used to generate a de-personalized ID card either on an individual image base (Figure 6) or in terms of a composite image (Figure 7).

5. 3 Statistical Verification

When a document is prepared using MS Word, for example, or any other major word processing package, statistical information from the document can be gathered including information about the author, date and time, number of characters and spaces and so on. A verification system can use this information to check the authenticity of the document. Any attempt at modification will then be reflected in its statistics. The system can either incorporate these data in plaintext or as a diffused code into a patch on the document which is encoded into an indecipherable image. The image needs to be attractively packaged in an appropriate place on the document - the bottom right hand corner for example. It is assumed that the recipient of the document (scanned or electronic) will have the appropriate software available. The encoded image is read into the decoding software and text-recognition used to reveal the text which is then compared with the plain-text statistics of the document. The data in the image can alternatively be checked manually against the statistics of the file instead of using text recognition. Each author can have a particular key for encoding an image. Upon receipt, the recipient applies that particular key to decode the image. Alternatively, a separate one-time PIN can be transmitted to the recipient in order to decode the image.

5. 4 Original Copy Verification

When a document is scanned subject to the scanner type and settings (including the resolution, for example), the output digital image file will have a specific statistical characteristic compounded in the histogram. If the document is copied and scanned again then this characteristic histogram will change because of the copy process. In general, a copied document will tend to have a smoother histogram since it is, in effect, the original document image convolved with a PSF that is characteristic of the copier (a function of the composite scan/print process). By printing a texture code of the histogram of the original document, typically on the back of the document, the document can be scanned and the histogram compared with the watermark, at least within an acceptable tolerance. This application has value in the authentication of high value documents such a Bank Bonds, an example of which is given in Figure 9, which illustrates the texture map and the reconstruction of the plaintext, i.e. a histogram of the luminance of the original colour image together with some basic statistical information.


U.S.$10,000
United Kingdom
U.S.$2,500,000,000 Floating Rate Notes Due 1992
SPECIMEN
U.S.$10,000 (Ten thousand United States dollars)
For THE LORDS COMMISSIONERS OF HER MAJESTY'S TREASURY
Channel:
Luminosity

Mean: 136.52
Std Dev: 60.92
Median: 160
Pixels: 440360
Level:
Count:
Percentile:
Cache Level: 4

Figure 9. Example of a high value Bank Bond (left), a texture map (center) obtained by diffusing the information given on the right-hand side which is a reconstruction of statistical information relating to scan of original document (Bank Bond) using Adobe Photoshop V5.

By specifying the type of scanner and operational constraints, statistical information of this type (i.e. the mean, standard deviation and median, for example) can be used to qualify whether or not a Bank Bond or other high value documents has been copied. This statistical verification may include measures relating to the RGB components for the case of high value colour documents (which is usually the case as reproduced here). Although each scan (using the same scanner with identical settings) does not output an identical digital image (due to slight differences in the crop, for example, as well as the natural 'jitter' of the scanner), the statistical information should not change significantly unless a copy has been made, acceptable tolerances having been established *a priori*.

5. 5 Component Verification and Transaction Tracking

The method discussed can be extended to include a 'specific parts' from of the text that must be correct, e.g. a sum of money, name of beneficiary etc. The diffused code can be placed into the background of each data field. Also called fingerprinting, transaction tracking involves the embedding of different watermark into each distributed copy. This is especially useful for identifying people who obtain a document legally but illegally redistribute it.

5. 6 Leaked Document Monitoring

One common method to monitor and discover any 'leak' associated with a very important document is to use visible marks. For example, highly sensitive documents are sometimes printed on backgrounds containing large grey digits using a different number for each copy. Records are then kept about who has which copy. Of course, imperceptible watermarks (or at least diffused watermarks) are preferable to visible marks. They are easy to remove/replace from a document when it is copied. Using this model for document watermarking, the tracking number is diffused and inserted into the background, the diffused watermark being inseparable from the document. The adversary (a person who attempts to remove, disable, or forge a watermark for the purpose of circumventing its original purpose) does not know the embedded number and can not recognize the difference between copies (it is difficult for human eyes to find a difference between two copies with different watermarks).

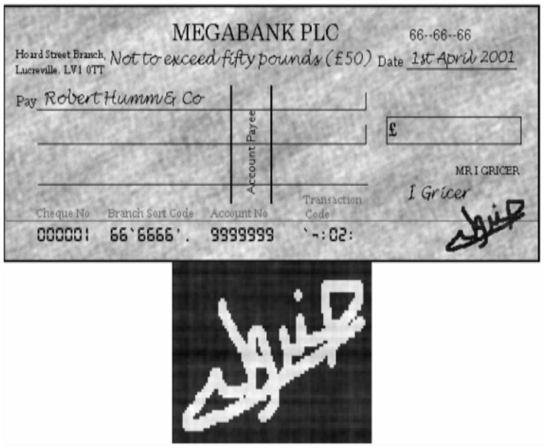
5. 7 Owner Identification (Copyright)

Copyright can be undertaken by embedding the identity of a document's copyright holder as a watermark in order to prevent other parties from claiming the copyright of the document. The embedded data can be a biometric characteristic (such as a signature, for example). The receiver of the document reconstructs the signature used to watermark the document, which is then used to verify the authors claimed identity.

5. 8 Signature Verification

Handwritten signatures are commonly used to certify the contents of a document or to authenticate legal transactions. A handwritten signature is a well-known biometric attribute. Other biometric attributes, which are commonly used for authentication include iris, hand geometry, face and fingerprints ([10] and [11]). While attributes like the iris and fingerprints do not change over time, they require special and relatively expensive hardware to capture the biometric data. An important advantage of the signature over other biometric attributes is that it has been traditionally used in authenticating documents and hence is socially accepted.

Signature verification is usually done by visual inspection. In automatic signature verification, a computer takes over the task of comparing two signatures to determine if the similarity between the signatures exceeds some pre-specified threshold. There a many similarity measures that can be used for this purpose. Figure 10 shows an example for this approach. The signature of the customer is diffused and inserted into the background of the cheque. Each customer has their own key that is known only to them and their bank. They use the key to generate the background and then print the cheque. The bank then uses the key to extract the customer signature from the cheque. If the extracted and the existing signatures on the cheque are matched to each other, then the cheque is accepted.



MEGABANK PLC 66--66--66

Hoard Street Branch,
Lucreville, LV1 0TT

Not to exceed fifty pounds (£50) Date 1st April 2001

Pay Robert Humm & Co

Account Payee

£

MR I GRICER

I Gricer

Cheque No	Branch Sort Code	Account No	Transaction Code
000001	66`6666'.	9999999	`¬:02:

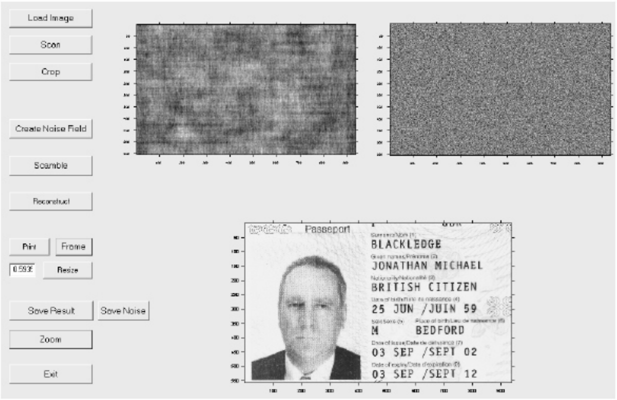
Figure 10. Watermarked cheque background texture - obtained by diffusing sample signature with noise and signature recovery obtained after covertext removal (through application of median filtering) and noise correlation (below).

5. 9 Passport Authentication

Like any other security document, ID card and so on, a passport consists of a number of security features depending on the sophistication of the design associated with the authority responsible for an issue. These range from the use of printing complex backgrounds, micro-printing, conventional paper watermarking, UV watermarking, foil holograms, ghost images and so on. Each of these security features may be more or less difficult to counterfeit depending on the sophistication of the feature and the counterfeiter.

The consider the use of texture coding within the context of authenticating a passport including the protocol associated with a typical 'cycle' has been considered. The method is simple and cost effective to implement in terms of the hardware required, i.e. Standard PC, flatbed scanner and printer, all of which are COTS. All that is required is a remote web-site hub to which digital scans of the texture code can be emailed and where a decrypt can take place, forwarding the result back to the point of enquiry. The principal idea is to take a low resolution scan (say 600dpi) of the page (or pages) of a passport that contains the primary information, e.g. Passport number, Name, Date of birth, Signature and Photograph of the passport holder - the plaintext. This plaintext is then forwarded to a designated Hub where it is diffused with a unique noise field that is maintained at the Hub alone to produce the ciphertext. The result is then emailed back to the user, printed and the result (permanently or as required) inserted into the passport, a process that is similar to issuing a Visa, for example.

At any point of contact, if the passport requires authentication, the ciphertext is scanned and the digital image emailed to the appropriate Hub where upon it is decrypted and the result (the watermark) is sent back to the point of origin. Automation of this cycle would require a new infrastructure to be established which is both time consuming and expensive. Instead, the cycle described above would be best suited for the use with regard to spot checks at an airport terminal, for example, especially if the holder of the passport or the passport itself is suspect. The scanning process (using a standard flat bed scanner, for example) can then be undertaken while the holder of the passport is waiting for it to be authenticated (or otherwise) based on a visual comparison between the decrypt and the plaintext. An example of the system designed to implement this method working with full colour images in which each RGB component is diffused with the same cipher is shown in Figure 11. This figure shows the GUI and the result of decrypting the texture code associated with the author's passport after being scanned at 300dpi and the output is emailed as a JPEG attachment.



BLACKLEDGE
JONATHAN MICHAEL
BRITISH CITIZEN
25 JUN /JUIN 59
M BEDFORD
03 SEP /SEPT 02
03 SEP /SEPT 12

Figure 11. Example passport authentication system (GUI) and a decrypt of the texture code (top-left image) generated by diffusing the cipher (top-right image) with the passport holders details revealed in the decrypt as shown (bottom image).

6. Discussion

Commercial realization of the approach discussed in this paper has been invested by Lexicon Data Limited which has developed various imaging technologies that involves the use of high strength encryption algorithms to scramble mixed images (i.e. colour photos and data) so that both printing and scanning can be kept simple and at a low resolution without compromising security. The 'noise fields' used to scramble the images make it is impossible to forge or counterfeit the document or card. However, printing the scrambled image onto an RFID card (a card with embedded Radio Frequency Identity chip), for example, enables it to be authenticated even with a poor image taken via a mobile phone. Lexicon Data Limited calls this product Cryptos^{TM} and it is available through the contact details given at http://www.lexicon-data.com/.

6. 1 Steganography and Cryptography

One of the principal components associated with the development of methods and algorithms to 'break' ciphertext is the analysis of the output generated by an attempted decrypt and its evaluation in terms of an expected type. The output type is normally assumed to be plaintext, i.e. the output is assumed to be in the form of characters, words and phrases associated with a natural language such as English or German, for example.

If a plaintext document is converted into an image file then the method discussed in this paper can be used to diffuse the plaintext image p using any other covertext image v to produce stegotext s. If both s and v are then encrypted, any attack on these data will not be able to make use of an 'analysis cycle' which is based on the assumption that the decrypted output is plaintext. This approach provides the user with a relatively simple method of 'confusing' the cryptanalyst and invalidates attack strategies that have been designed and developed on the assumption that the encrypted data have been derived from plaintext alone. In steganography, one message is hidden inside another, without disclosing the existence of the hidden message or making it apparent to an observer that this message contains a hidden message [12]. Moreover, the information hidden by a watermarking system is always associated with the object to be protected or its owner while steganographic systems just hide information. On the other hand, cryptography can be defined as the study of secret writing (i.e. concealing the contents of a secret message by transforming the original message into a form that cannot be easily interpreted by an observer). The method considered here (diffusion and confusion) can be used in both applications. The hidden message can be transformed into a diffused form (i.e. encrypted) and inserted into the background. The hidden information might have no relation with the text (foreground). At the same time, backgrounds are usually used with documents and so diffused data will not necessarily trigger the attention of an observer. Moreover, the hidden message may also be encrypted which increases the security level of such documents.

6.2 Covert Encryption using Digital Image Steganography

The principles discussed in the previous section can be used to design an entirely covert encryption system. By inputting any encrypted file as binary data, a binary image can be generated (consisting entirely of pixels with values of 0 or 1). For example, consider the plaintext Cryptology which is encrypted to provide the ciphertext string ydr39bkLP9 and is equivalent to the 7-bit ASCII bit stream 11110011100100111001001100110111001 11000101101011100110010100000111001. This bit stream is converted into 9 x 9 square image (the image does not necessarily need to be square and is used here for illustrative purposes only) with zero padding being used to complete the array as given below:

```
1 1 1 1 0 0 1 1 1
0 0 1 0 0 1 1 1 0
0 1 0 0 1 1 0 0 1
1 0 1 1 1 0 0 1 1
1 0 0 0 1 0 1 1 0
1 0 1 1 1 0 0 1 1
0 0 1 0 1 0 0 0 0
0 1 1 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0
```

The binary image is then diffused with a random image field and the output is embedded in a covertext through addition using a suitable diffusion-to-confusion ratio (suitable in the sense that the binary image is recovered with no bit errors for the case when the difference between the covertext and stegotext is insignificant).

The size of the image that is required to implement this method is related to the binary length of the ciphertext. Assuming that the ciphertext and plaintext are of the same size (i.e. no padding is applied to the plaintext before encryption), and, given that the average number of letters per word (in the English language) is 6 (including the space), then a n^2 binary image will provide for approximately $n^2/(7x6)$ words.

In order to enhance the cryptographic strength associated with this approach, the cipher can be obtained from a genuine random number generator such as HotBits (http://www.fourmilab.ch/hotbits/) and then encrypted (to secure the data file) using a specified cryptosystem. Cleary, in addition to the receiver of the stegotext requiring the facility to decrypt the reconstruction, in order to obtain this reconstruction, the receiver must have the cipher and the covertext. The covertext should be one of the database of images maintained by both parties together with the cipher that is ideally stored in encrypted form. Because the stegotext and covertext images look identical, the receiver can search through the image database to select the appropriate covertext. The whole point of this process is that it provides a way of camouflaging the encrypted data during transmission, the difference between sending the ciphertext and the stegotext as digital images. However, in this process, a macro-key is required to be exchanged which is composed of the following: (i) the cipher; (ii) the covertext database; (iii) the decryption system. A covertext database is required for two reasons. First, each time a transmission is undertaken, it is safer to transmit a different stegotext in order not to alert a potential attacker to multiple transmissions of the same data; second, a database of images should be stored rather than a single image in order that no apparent significance is given to a single image should the platform (i.e. PC or USB stick, for example) be compromised.

Conclusion

Valuable paper documents are subjected to misuse by criminals. This is largely due to the dramatic improvement in personal computer hardware and peripheral equipment. Embedding watermarks into a printed document is one way to secure them. The ability to extract the watermark from a printed copy is generally useful to help establish ownership, authenticity, and to establish the origin of an unauthorized disclosure. However, finding a robust watermarking technique is a continuing challenge. This is due to extensive amount of noise that is added when a document is printed and scanned. Moreover, printed documents do not maintain their quality over time. In this paper, a robust watermarking method for document security has been presented. Unlike conventional watermarking techniques, this approach can extract the hidden watermark after a print/scan attack which is achieved by using the convolution and correlation processes for coding and decoding respectively. This approach is chosen because of its compatibility with the principles of the physical optics involved in scanning a document. The watermark is diffused (convolved) with a modified noise field and placed into the background of a covertext, typically a text document. The watermark can be extracted by removing the covertext using a modified median filter. The diffused watermark is then correlated with the original noise field.

Although details are beyond the scope of this paper, the method is robust to a wide variety of attacks including geometric attacks, drawing, crumpling and print/scan attacks. The method is relatively insensitive to lossy compression, filtering, amplitude adjustments, additive noise and thresholding. The principal weakness of the system is its sensitivity to rotation and cropping. This can be minimized by orienting the document correctly and accurately before scanning and using automatic cropping software which is available with selected scanners (e.g. Cannon scanners).

Alternatively, introduction of a frame provides a reference feature from which an accurate crop can be obtained. The visibility of the diffused watermark and the compatibility of this system with the physical principles of an imaging system, increase the robustness of the system and provides a successful approach to the extraction of the watermark after scanning at low resolution. Moreover, using correlation in the extraction phase increases the robustness of the system to some important attacks such as translation and cropping (most likely to occur during a scan).

The system is secure in that it can not be attacked easily. First, the feature is not 'suspicious' as many documents have a background texture. Second, the attacker does not know the algorithm used to generate the diffused watermark. Even if the attacker does know the algorithm, he/she must still know a significant amount of information before the system can be broken, such as: the correct key, the diffusion operator type, the original image size and so on.

Acknowledgement

Jonathan Blackledge is the Stokes Professor of Digital Signal Processing which is funded by the Science Foundation Ireland.

References

[1]. **Ferguson N. and Schneier B.**, *Practical Cryptography,* Wiley, 2003.

[2]. **Menezes A. J., van Oorschot P.C. and Vanstone S. A.**, *Handbook of Applied Cryptography*, CRC Press, 2001.

[3]. **Schneier B.**, *Applied Cryptography*, Second Edition Wiley, 1996.

[4]. **Salomaa A.**, *Public Key Cryptography*, Springer, 1996.

[5]. **Articsoft Technologies**, *Introduction to Encryption*, 2005;http://www.articsoft.com/wp_explaining_encryption.htm

[6]. **Ellison C. and Shneier B.**, Ten Risks of PKI: What Your Not Being Told About Public Key Infrastructure, *Computer Security Journal XVI(1), 2000;* http://www.schneier.compaper-pki.pdf.

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

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

[9]. **Kutter M. and Hartung**, *Introduction to watermarking techniques,* Information Hiding: Techniques for Stegangraphy and Digital Watermarking, S.Katezenbeisser and F. A. P. Petitcolas, Eds., Ch. 5, 97-120, ArtechHouse, Boston, 2000.

[10]. **Jain A K, Pankanti S., and Bolle R., Eds.**, *BIOMETRICS: Personal Identification in Networked Society,* Kluwer, 1999.

[11]. **Jain A. K., Griess F. D. and Connell S. D.**, "On-line signature verification", *Pattern Recognition* 35, 2963-2972, December 2002.

[12]. **Anderson R. J. and Petitcolas F.**, "On the limits of steganography", *IEEE: Selected Areas in Communications,* 16, 474-481, May 1998.

[13]. **Mahmoud K. W.**, *Low Resolution Watermarking for Print Security,* PhD Thesis, Loughborough University, 2005.

Appendix

Pseudo Code (No Standard) for Watermark Generation and Recovery

```
void function WM(cipher,plaintext,covertext,watermark,size,pth)
\\Function: Applies a watermark to a covertext image based on diffusing a
\\          plaintext image with a noise field (cipher).
\\Input arrays: cipher - noise field
\\              plaintext - input image
\\              covertext - host image
```

```
\\Output array: watermark - watermarked image
\\All input arrays are taken to be of type float with values ranging from 0 to 1 inclusively.
\\Parameters: size - image size (assumed to be size x size)
\\                  pth  - diffused plaintext-to-host image ratio
\\Required Internal Functions:
\\FFT    Function to Compute the Forward Fast Fourier Transform
\\IFFT  Function to compute the Inverse Fast Fourier Transform
\\MAX  - Function to compute the maximum value of an array
\\REAL - Funtion to extracts the real component of an array
\\ABS  - Function to compute the absolute value of a (complex) array
\\Start algorithm
 BEGIN:
\\Compute spectrum of cipher
  Cipher=FFT(cipher);
\\Compute spectrum of plaintext
  Plaintext=FFT(plaintext);
\\Compute Power Spectrum
  powerspectrum=ABS(cipher)*ABS(cipher);
\\Pre-condition power spectrum of cipher
 FOR i=1 to size AND j=1 to size DO:
        Temp=powerspectrum[i,j];
            IF temp==0
        Powerspectrum[i,j]=1;
              ELSE
       Powerspectrum[i,j]=powerspectrum[i,j];
           END IF
END DO
\\Diffuse plaintext image with pre-conditioned cipher
  FOR i=1 to size AND j=1 to size DO:
      diffusion[i,j]=cipher[i,j]*plaintext[i,j]/powerspectrum[i,j];
  END DO
\\Compute absolute value of IFFT
  diffusion=ABS(IFFT(diffusion));
\\Normalise diffused field
  Diffusion=diffusion/MAX(diffusion);
\\Compute the watermarked image
```

```
  FOR i=1 to size AND j=1 to size DO:
      watermark[i,j]=pth*diffusion[i,j]+covertext[i,j];
  END DO
\\Normalise output
  watermark=watermark/MAX(watermark);
  FINISH
void function RECWM(cipher,watermark,covertext,plaintext,size)
\\Function: Recovers watermark from a diffused image field
\\Input arrays: cipher    - noise field image
                watermark - watermarked image
                Covertext - host image
\\All input arrays are taken to be of type float with values ranging from 0 to 1 inclusively.
\\Parameters: size - image size (size x size)
\\Output: plaintext - recovered watermark image
\\Required Internal Functions:
\\FFT   Function to Compute the Forward Fast Fourier Transform
\\IFFT  Function to compute the Inverse Fast Fourier Transform
\\MAX  - Function to computes the maximum value of an array
\\ABS  - Function to compute the absolute value of a (complex) array
\\CONJ  Function to compute the conjugate of a complex array
\\Start algorithm
BEGIN:
\\Subtract covertext from watermarked image
  FOR i=1 to size AND j=1 to size DO:
     Diffusion[i,j]=watermark[i,j]-covertext[i,j];
  END DO
\\Compute spectrum of cipher
  cipher=FFT(cipher);
\\Compute spectrum of diffused field
  diffusion=FFT(diffusion);
\\Correlate diffused field with cipher
  FOR i=1 to n AND j=1 to n DO:
      Plaintext[i,j]=CONJ(cipher[i,j])*diffusion[i,j];
  END DO
\\Compute absolute part of IFFT
  Plaintext=ABS(IFFT(plaintext));
```

```
\\Normalise output
 plaintext=plaintext/MAX(plaintext);
 FINISH
```

ABOUT THE AUTHORS

** Stokes Professor of Digital Signal Processing, School of Electrical Engineering Systems, Dublin Institute of Technology, Ireland.*
*** Department of Computer Science, University of the Western Cape, South Africa.*

Jonathan Blackledge graduated in physics from Imperial College in 1980. He gained a Ph.D in theoretical physics from London University in 1984 and was then appointed as Research Fellow of Physics at Kings College, London from 1984 to 1988, specializing in inverse problems in electromagnetism and acoustics. During this period, he worked on a number of industrial research contracts undertaking theoretical and computational research into the applications of inverse scattering theory for the analysis of signals and images. In 1988, he joined the Applied Mathematics and Computing Group at Cranfield University as a Lecturer and later, as a Senior Lecturer and Head of Group where he promoted postgraduate teaching and research in applied and engineering mathematics in areas which included computer aided engineering, digital signal processing and computer graphics. While at Cranfield, he co-founded Management and Personnel Services Limited through the Cranfield Business School which was originally established for the promotion of management consultancy working in partnership with the Chamber of Commerce. He managed the growth of the company from 1993 to 2007 to include the delivery of a range of National Vocational Qualifications, primarily through the City and Guilds London Institute, including engineering, ICT, business administration and management. In 1994, Jonathan Blackledge was appointed as a Professor of Applied Mathematics and Head of the Department of Mathematical Sciences at De Montfort University where he expanded the post-graduate and research portfolio of the Department and established the Institute of Simulation Sciences. In 2002, he was appointed as a Visiting Professor of Information and Communications Technology in the Advanced Signal Processing Research Group, Department of Electronics and Electrical Engineering at Loughborough University, England (a group which he co-founded in 2002 as part of his appointment). In 2004, he was appointed as a Professor Extraordinaire of Computer Science in the Department of Computer Science at the University of the Western Cape, South Africa. His principal roles at these institutes include the supervision of M.Sc. and M.Phil/Ph.D students and the delivery of specialist short courses for their Continuous Professional Development programmes. He currently holds the prestigious Stokes Professorship in Digital Signal Processing under the Science Foundation Ireland Programme based in the School of Electrical Engineering Systems at Dublin Institute of Technology, Ireland.

Mary Lynne Hallot started her career as an arts teacher in a variety of high schools. After training as an Educational Psychologist, she worked for the Wits Rural Facility (University of the Witwatersrand) as a Research Officer involved in community-based education and health programmes. Subsequent to this, she also worked as a Senior Lecturer in the Department of Educational Psychology at the University of Cape Town, Rhodes and Wits. There she co-managed the Community Psychology Centre and coordinated the Honours Degree course in Educational Psychology. She was appointed as the Head of the Department of Arts at the University of Midrand in 2000 and then as the Head of the Department of Arts at Witwatersrand Technikon in 2001. In 2002, she joined the Department of Computer Science at the University of the Western Cape to pursue her Ph.D in Computer Science where she successfully completed her thesis entitled Digital Watermarking Methods for Data Security and Authentication in 2008 under the supervision of Professor Blackledge.