Steganography using Stochastic Diffusion for the Covert Communication of Digital Images

Jonathan M Blackledge and AbdulRahman I Al-Rawi *

Abstract— **This paper is devoted to the study of a method called Stochastic Diffusion for encrypting digital images and embedding the information in another host image or image set. We introduce the theoretical background to the method and the mathematical models upon which it is based. This includes a comprehensive study of the diffusion equation and its properties leading to a convolution model for encrypting data with a stochastic field that is fundamental to the approach considered. Two methods of implementing the approach are then considered. The first method introduces a lossy algorithm for hiding an image in a single host image which is based on the binarization of the encrypted data. The second method considers a similar approach which uses three host images to produce a near perfect reconstruction from the decrypt. In both cases, details of the algorithms developed are provide and examples given. The methods considered have applications for covert cryptography and the authentication and self-authentication of documents and full colour images.**

Keywords: Image Encryption Information Hiding, Steganography, Stochastic Diffusion, Symmetric Encryption

1 Introduction

The relatively large amount of data contained in digital images makes them a good medium for undertaking information hiding. 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 for 24-bit colour images. 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 we change the least significant bit 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. If this is done for each colour component then a single letter can be represented using just three pixels. The larger the host image compared with the hidden message, the more difficult it is to detect the message. The host image represents the key to recovering the hidden image. 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.

The large majority of methods developed for image information hiding do not include encryption of the hidden image. In this paper we consider an approach in which a hidden image is encrypted by diffusion with a noise field before being embedded in the host image. The paper provides a short survey on encrypted information hiding and then presents a detailed account of the mathematical foundations upon which the method, known as Stochastic Diffusion, is based. Two applications are then consider: (i) Lossy information hiding which is based on the binarisation of the encrypted field; (ii) Lossless information hiding which is based on using three separate host images in which the encrypted information is embedded. The methods considered have a range of applications in document and full colour image authentication.

2 Survey on Encrypted Information Hiding

Compared with information hiding in general, there are relatively few publications that have addressed the issue of hiding encrypted information. We now provide an overview of some recent publications in this area.

In [1], a novel method is proposed for hiding the transmission of biometric images based on chaos and image content. To increase the security of the watermark, it is first encrypted using a chaotic map where the keys used for encryption are generated from a palm print image. The

*Information and Communications Security Research Group, Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland.

pixel value distribution, illumination and various image distortions are different for each palm print image (even if they are for the same person) because the palm print image is different each time the image is captured. In [1], the normalized mean value of three randomly selected pixels from the palm print image is used as an initial condition for the chaotic map. The logistic map is used to generate a one-dimensional sequence of real numbers which are then mapped into a binary stream to encrypt the watermark using an XOR operation. The encrypted watermark is then embedded into the same palm print image used to derive the secret keys. The stego-palm print image is hidden into the cover image using a novel content-based hidden transmission scheme. First the cover image is segmented into different regions using a classical watershed algorithm. Due to the over-segmentation resulting from this algorithm, a Region-based Fuzzy C-means Clustering algorithm is used to merge similar regions. The entropy of each region is then calculated and the stego-palm print image embedded into the cover image according to the entropy value with more information being embedded in highly textured regions compared to uniform regions. A threshold value T is used to partition the two regions. If the entropy is greater than T, the binary streams of the secret data are inserted into the 4 least significant bits of the region, and if the entropy is smaller than T, the binary streams of the secret data are inserted into the 2 least significant bits of the region. Colour host images are decomposed into RGB channels before embedding.

In [2] a method of hiding the transmission of biometric images based on chaos and image content is proposed that is similar to [1]. The secret data is a grayscale image of size 128×128 and before encrypting it, it is converted into a binary stream with the logistic map being used for encryption. The encryption keys used to produce the logistic chaotic map sequence are generated randomly using any pseudo random number generating algorithm. The authors use 256×256 color images as hosts which are converted into grayscale images and segmented into different regions using the watershed algorithm to eliminate over-segmentation. A Fuzzy C-means Clustering algorithm is used to implement similar region merging. Each region is classified into a certain cluster based on the regions of the watershed lines. A k-nearest neighbour method is used to partition the regions needing re-segmentation. For the resultant image without watershed lines, the entropy is calculated and the secret image is embedded according to the entropy values. The colour host image is decomposed into RGB channels for embedding. Highly textured regions are used to embed more information and a threshold value T is used to separate the two regions. If the entropy is smaller than T the binary streams of the secret data are inserted into the 2 Least Significant Bits (LSB) of the three channels of the region. If the entropy is greater than T, the binary streams of the secret data are inserted into the 4 LSB of the three channels of the region.

Another steganographic method is proposed in [3] for PNG images based on the information sharing technique. The secret image M is divided into shares using a (k, n)-threshold secret sharing algorithm. Secret shares are then embedded into the alpha-channel of the PNG cover image. The image M is first divided into t-bit segments which transforms each segment into a decimal number resulting in a decimal number sequence. A $(4, 4)$-threshold secret sharing algorithm is used to generate 'partial shares' which are then embedded into the host image by replacing the alpha-channel values of the host image with the values of the shares. The process is repeated for all decimal values of the secret data resulting in a stego-image. In general, if every four t-bit segments are transformed and embedded similarly, then the data hiding capacity is proportional to the chosen value of t in proportion to the dimension of the cover image. However, the larger the value of t the lower the visual quality of the stego-image which causes a wider range of the alpha-channel values to be altered leading to a more obvious non-uniform transparency effect appearing on the stego-image. The value of t is therefore selected to ensure a uniform distribution of the stego-image alpha-channel.

The principle of image scrambling and information hiding is introduced in [4] in which a double random scrambling scheme based on image blocks is proposed. A secret image of size $M \times N$ is divided into small sub-blocks of size 4×4 or 8×8, for example, and a scrambling algorithm used to randomize the sub-blocks using a given key. However, because the information in each inner sub-block remains the same, another scrambling algorithm is used with a second key to destroy the autocorrelation in each inner sub-block thereby increasing the difficulty of decoding the secret image. To make the hidden secret image more invisible, its histogram is compressed into a small range. Image hiding is then performed by simply adding the secret image to the cover image. The hidden image is recovered by expanding the histogram after extraction and decryption carried out for both the sub-blocks and inner sub-blocks to obtain a final reconstruction.

3 Diffusion and Confusion

The purpose of this section is to introduce the reader to some of the basic mathematical models associated with the processes of diffusion and confusion as based on the physical origins of these processes. This provides a theoretical framework for two of the principal underlying

concepts of cryptology in general as used in a variety of contexts.

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 [5], [6]. 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 attacker may be able to recover parts of the input from an observed output by a partially correct guess of the key used for encryption. The diffusion process is a function of sensitivity to initial conditions that a cryptographic system should have and further, the inherent topological transitivity that the system should also exhibit causing the plaintext to be mixed through the action of the encryption process.

Confusion ensures that the (statistical) properties of plaintext blocks are not reflected in the corresponding ciphertext blocks. Instead 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.

3.1 The Diffusion Equation

In a variety of physical problem, the process of diffusion can be modelled in terms of certain solutions to the diffusion equation whose basic homogeneous form is given by [7] - [10]

$$\nabla^2 u(\mathbf{r},t) = \sigma \frac{\partial}{\partial t} u(\mathbf{r},t), \quad \sigma = \frac{1}{D}$$

where D is the 'Diffusivity' and u is the diffused field which describes physical properties such as temperature, light, particle concentration and so on; $\mathbf{r}$ denotes the spatial vector and t denotes time.

The diffusion equation describes fields u that are the result of an ensemble of incoherent random walk processes, i.e. walks whose direction changes arbitrarily from one step to the next and where the most likely position after a time t is proportional to $\sqrt{t}$. Note that if $u(\mathbf{r},t)$ is a solution to the diffusion equation the function $u(\mathbf{r},-t)$ is not, i.e. it is a solution of the quite different equation,

$$\nabla^2 u(\mathbf{r},-t) = -\sigma \frac{\partial}{\partial t} u(\mathbf{r},-t).$$

Thus, the diffusion equation differentiates between past and future. This is because the diffusing field u represents the behaviour of some average property of an ensemble of many elements which cannot in general go back to their original state. This fundamental property of diffusive processes has a synergy with the use of one-way functions in cryptology, i.e. functions that, given an input, produce an output that is not reversible - an output from which it is not possible to compute the input.

Consider the process of diffusion in which a source of material diffuses into a surrounding homogeneous medium, the material being described by some initial condition $u(\mathbf{r},0)$ say. Physically, it is to be expected that the material will increasingly 'spread out' as time evolves and that the concentration of the material decreases further away from the source. The general solution to the diffusion equation yields a result in which the spatial concentration of material is given by the convolution of the initial condition with a Gaussian function, the time evolution of this process being governed by the same process. This solution is determined by considering how the process of diffusion responds to a single point source which yields the Green's function (in this case, a Gaussian function).

3.2 Green's Function for the Diffusion Equation

We evaluate the Green's function [10]-[12] for for the diffusion equation satisfying the causality condition

$$G(\mathbf{r} \mid \mathbf{r}_0, t \mid t_0) = 0 \quad \text{if} \quad t < t_0$$

where $\mathbf{r} \mid \mathbf{r}_0 \equiv \mid \mathbf{r} - \mathbf{r}_0 \mid$ and $t \mid t_0 \equiv t - t_0$. This can be accomplished for one-, two- and three-dimensions simultaneously [8]. Thus with $R = \mid \mathbf{r} - \mathbf{r}_0 \mid$ and $\tau = t - t_0$ we require the solution of the equation

$$\left(\nabla^2 - \sigma \frac{\partial}{\partial \tau}\right) G(R,\tau) = -\delta^n(R)\delta(\tau), \quad \tau > 0$$

where n is 1, 2 or 3 depending on the number of dimensions and δ is the corresponding Dirac delta function [13] - [15]. One way of solving this equation is first to take the Laplace transform with respect to τ, then solve for G (in Laplace space) and then inverse Laplace transform the result [16]. This requires an initial condition to be specified (the value of G at $\tau = 0$). Another way to solve this equation is to take its Fourier transform with respect to

R, solve for G (in Fourier space) and then inverse Fourier transform the result [17], [18]. Here, we adopt the latter approach. Let

$$G(R,\tau) = \frac{1}{(2\pi)^n}\int_{-\infty}^{\infty} \widetilde{G}(\mathbf{k},\tau)\exp(i\mathbf{k}\cdot\mathbf{R})d^n\mathbf{k}$$

and

$$\delta^n(R) = \frac{1}{(2\pi)^n}\int_{-\infty}^{\infty} \exp(i\mathbf{k}\cdot\mathbf{R})d^n\mathbf{k}.$$

Then the equation for G reduces to

$$\sigma\frac{\partial\widetilde{G}}{\partial\tau} + k^2\widetilde{G} = \delta(\tau)$$

where $k =\mid \mathbf{k} \mid$ which has the solution

$$\widetilde{G} = \frac{1}{\sigma}\exp(-k^2\tau/\sigma)H(\tau)$$

where $H(\tau)$ is the step function

$$H(\tau) = \begin{cases} 1, & \tau > 0; \\ 0, & \tau < 0. \end{cases}$$

Hence, the Green's functions are given by

$$G(R,\tau) = \frac{1}{\sigma(2\pi)^n}H(\tau)\int_{-\infty}^{\infty}\exp(i\mathbf{k}\cdot\mathbf{R})\exp(-k^2\tau/\sigma)d^n\mathbf{k}$$

$$= \frac{1}{\sigma(2\pi)^n}H(\tau)\left(\int_{-\infty}^{\infty}\exp(ik_xR_x)\exp(-k_x^2\tau/\sigma)dk_x\right)...$$

By rearranging the exponent in the integral, it becomes possible to evaluate each integral exactly. Thus, with

$$ik_xR_x - k_x^2\frac{\tau}{\sigma} = -\left(k_x\sqrt{\frac{\tau}{\sigma}} - i\frac{R_x}{2}\sqrt{\frac{\sigma}{\tau}}\right)^2 - \left(\frac{\sigma R_x^2}{4\tau}\right)$$

$$= -\frac{\tau}{\sigma}\xi^2 - \left(\frac{\sigma R_x^2}{4\tau}\right)$$

where

$$\xi = k_x - i\frac{\sigma R_x}{2\tau}.$$

The integral over k_x becomes

$$\int_{-\infty}^{\infty}\exp\left[-\left(\frac{\tau}{\sigma}\xi^2\right) - \left(\frac{\sigma R_x}{4\tau}\right)\right]d\xi$$

$$= e^{-(\sigma R_x^2/4\tau)}\int_{-\infty}^{\infty}e^{-(\tau\xi^2/\sigma)}d\xi$$

$$= \sqrt{\frac{\pi\sigma}{\tau}}\exp\left[-\left(\frac{\sigma R_x^2}{4\tau}\right)\right]$$

with similar results for the integrals over k_y and k_z giving the result

$$G(R,\tau) = \frac{1}{\sigma}\left(\frac{\sigma}{4\pi\tau}\right)^{\frac{n}{2}}\exp\left[-\left(\frac{\sigma R^2}{4\tau}\right)\right]H(\tau).$$

The function G satisfies an important property which is valid for all n:

$$\int_{-\infty}^{\infty} g(R,\tau)d^n\mathbf{r} = \frac{1}{\sigma};\quad \tau > 0.$$

This is the expression for the conservation of the Green's function associated with the diffusion equation. For example, if we consider the diffusion of heat, then if at a time t_0 and at a point in space $\mathbf{r}_0$ a source of heat is introduced, then the heat diffuses out through the medium characterized by σ in such a way that the total flux of heat energy is unchanged.

3.3 Green's Function Solution

Working in three dimensions, we consider the Green's solution to the inhomogeneous diffusion equation [8], [9]

$$\left(\nabla^2 - \sigma\frac{\partial}{\partial t}\right)u(\mathbf{r},t) = -S(\mathbf{r},t)$$

where S is a source of compact support ($\mathbf{r}\in V$) and define the Green's function as the solution to the equation

$$\left(\nabla^2 - \sigma\frac{\partial}{\partial t}\right)G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0) = -\delta^3(\mathbf{r}-\mathbf{r}_0)\delta(t-t_0).$$

The function S describes a source that is being diffused - a source of heat, for example - and is taken to be localised in space.

It is convenient to first take the Laplace transform of these equations with respect to $\tau = t - t_0$ to obtain

$$\nabla^2\bar{u} - \sigma[-u_0 + p\bar{u}] = -\bar{S}$$

and

$$\nabla^2\bar{G} + \sigma[-G_0 + p\bar{G}] = -\delta^3$$

where

$$\bar{u}(\mathbf{r},p) = \int_0^{\infty}u(\mathbf{r}\mid\mathbf{r}_0,\tau)\exp(-p\tau)d\tau,$$

$$\bar{G}(\mathbf{r},p) = \int_0^{\infty}G(\mathbf{r}\mid\mathbf{r}_0,\tau)\exp(-p\tau)d\tau,$$

$$\bar{S}(\mathbf{r},p)=\int_0^\infty S(\mathbf{r},\tau)\exp(-p\tau)d\tau,$$

$$u_0\equiv u(\mathbf{r},\tau=0)\quad\text{and}\quad G_0\equiv G(\mathbf{r}\mid\mathbf{r}_0,\tau=0)=0.$$

Pre-multiplying the equation for $\bar{u}$ by $\bar{G}$ and the equation for $\bar{G}$ by $\bar{u}$, subtracting the two results and integrating over V we obtain

$$\int_V(\bar{G}\nabla^2\bar{u}-\bar{u}\nabla^2\bar{G})d^3\mathbf{r}+\sigma\int_V u_0\bar{G}d^3\mathbf{r}$$

$$=-\int_V\bar{S}\bar{G}d^3\mathbf{r}+\bar{u}(\mathbf{r}_0,\tau).$$

Using Green's theorem [19], i.e. given that (Gauss' theorem for any vector $\mathbf{F}$)

$$\int_V\nabla\cdot\mathbf{F}d^3\mathbf{r}=\oint_S\mathbf{F}\cdot\hat{\mathbf{n}}d^2\mathbf{r}$$

where S is the surface that encloses a volume V and $\hat{\mathbf{n}}$ is a unit vector perpendicular to the surface element $d^2\mathbf{r}$, then, for two scalars f and g

$$\int_V(f\nabla^2g-g\nabla^2f)d^3\mathbf{r}=\int_V\nabla\cdot(f\nabla g-g\nabla f)d^3\mathbf{r}$$

$$=\oint_S(f\nabla g-g\nabla f)\cdot\hat{\mathbf{n}}d^2\mathbf{r}$$

and rearranging the result gives

$$\bar{u}(\mathbf{r}_0,p)=\int_V\bar{S}(\mathbf{r},p)\bar{G}(\mathbf{r}\mid\mathbf{r}_0,p)d^3\mathbf{r}$$

$$+\sigma\int_V u_0(\mathbf{r})\bar{G}(\mathbf{r}\mid\mathbf{r},p)d^3\mathbf{r}+\oint_S(\bar{g}\nabla\bar{u}-\bar{u}\nabla\bar{g})\cdot\hat{\mathbf{n}}d^2\mathbf{r}$$

Finally, taking the inverse Laplace transform and using the convolution theorem for Laplace transforms, we can write

$$u(\mathbf{r}_0,\tau)=\int_0^\tau\int_V S(\mathbf{r},\tau')G(\mathbf{r}\mid\mathbf{r}_0,\tau-\tau')d^3\mathbf{r}d\tau'$$

$$+\sigma\int_V u_0(\mathbf{r})G(\mathbf{r}\mid\mathbf{r}_0,\tau)d^3\mathbf{r}$$

$$+\int_0^\tau\oint_S G(\mathbf{r}\mid\mathbf{r}_0,\tau')\nabla u(\mathbf{r},\tau-\tau')\cdot\hat{\mathbf{n}}d^2\mathbf{r}d\tau'$$

$$-\int_0^\tau\oint_S u(\mathbf{r},\tau')\nabla G(\mathbf{r}\mid\mathbf{r}_0,\tau-\tau')\cdot\hat{\mathbf{n}}d^2\mathbf{r}d\tau'.$$

The first two terms are convolutions of the Green's function with the source function S and the initial condition $u(\mathbf{r},\tau=0)$, respectively.

If we consider the equation for the Green's function

$$\left(\nabla^2-\sigma\frac{\partial}{\partial t}\right)G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)=-\delta^3(\mathbf{r}-\mathbf{r}_0)\delta(t-t_0)$$

together with the equivalent time reversed equation

$$\left(\nabla^2+\sigma\frac{\partial}{\partial t}\right)G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)=-\delta^3(\mathbf{r}-\mathbf{r}_1)\delta(t-t_1),$$

then pre-multiplying the first equation by $G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)$ and the second equation by $G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)$, subtracting the results and integrate over the volume of interest and over time t from $-\infty$ to t_0 then, using Green's theorem, we obtain

$$\int_{-\infty}^{t_0}dt\oint_S G(\mathbf{r}\mid\mathbf{r}_1,-t\mid t_1)\nabla G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)\cdot\hat{\mathbf{n}}d^2\mathbf{r}$$

$$-\int_{-\infty}^{t_0}dt\oint_S G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)\nabla G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)\cdot\hat{\mathbf{n}}d^2\mathbf{r}$$

$$-\sigma\int_V d^3\mathbf{r}\int_{-\infty}^{t_0}dtG(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)\frac{\partial}{\partial t}G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)$$

$$-\sigma\int_V d^3\mathbf{r}\int_{-\infty}^{t_0}dtG(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)\frac{\partial}{\partial t}G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)$$

$$=G(\mathbf{r}_1\mid\mathbf{r}_0,t_1\mid t_0)-G(\mathbf{r}_0\mid\mathbf{r}_1,-t_0\mid-t_1).$$

If we then consider the Green's functions and their gradients to vanish at the surface S (homogeneous boundary conditions) then the surface integral vanishes[1]. The second integral is

$$\int_V d^3\mathbf{r}\left[G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)\right]_{t=-\infty}^{t_0}$$

and since

$$G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0)=0,\quad t<t_0$$

then

$$G(\mathbf{r}\mid\mathbf{r}_0,t\mid t_0\mid_{t=-\infty}=0.$$

Also

$$G(\mathbf{r}\mid\mathbf{r}_1,-t\mid-t_1)\mid_{t=t_0}=0$$

for t in the range of integration given. Hence,

$$G(\mathbf{r}_1\mid\mathbf{r}_0,t_1\mid t_0)=G(\mathbf{r}\mid\mathbf{r}_1,-t_0\mid-t_1).$$

This is the reciprocity theorem of the Green's function for the diffusion equation.

[1]This is also the case if we consider an infinite domain

3.4 Infinite Domain Solution

In the infinite domain, the surface integral is zero and we can work with the solution

$$u(\mathbf{r}_0,\tau)=\int_0^\tau\int_V S(\mathbf{r},\tau')G(\mathbf{r}\mid\mathbf{r}_0,\tau-\tau')d^3\mathbf{r}d\tau'$$

$$+\sigma\int_V u_0(\mathbf{r})G(\mathbf{r}\mid\mathbf{r}_0,\tau)d^3\mathbf{r}$$

which requires that the spatial extent of the source function is infinite but can include functions that are localised provided that $S\to 0$ as $\mid\mathbf{r}\mid\to\infty$ - a Gaussian function for example. The solution is composed of two terms. The first term is the convolution (in space and time) of the source function with the Green's function and the second term is the convolution (in space only) of the initial condition $u(\mathbf{r},0)$ with the same Green's function. We can write this result in the form

$$u(\mathbf{r},t)=G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}\otimes_t S(\mathbf{r},t)+\sigma G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}u(\mathbf{r},0)$$

where $\otimes_{\mathbf{r}}$ denotes the convolution over $\mathbf{r}$ and $\otimes_t$ denotes the convolution over t.

In the case where we consider the domain of interest over which the process of diffusion occurs to be infinite in extent, the solution to the homogeneous diffusion equation (when the source function is zero) specified as

$$\nabla^2u(\mathbf{r},t)-\sigma\frac{\partial}{\partial t}u(\mathbf{r},t)=0,\quad u(\mathbf{r},0)=u_0(\mathbf{r})$$

is given by the convolution of the Green's function with u_0, i.e.

$$u(\mathbf{r}_0,t)=\sigma G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}u_0(\mathbf{r}),\quad t>0$$

Thus, in one-dimension, the solution reduces to

$$u(x,t)=\sqrt{\frac{\sigma}{4\pi\sigma t}}\exp\left[-\frac{\sigma x^2}{4t}\right]\otimes_x u_0(x),\quad t>0$$

where $\otimes_x$ denotes the convolution integral over independent variable x and we see that the field u at a time $t>0$ is given by the convolution of the field at time $t=0$ with the one-dimensional Gaussian function

$$\sqrt{\frac{\sigma}{4\pi t}}\exp\left(-\frac{\sigma x^2}{4t}\right).$$

In two-dimensions, the result is

$$u(x,y,t)=\frac{\sigma}{4\pi t}\exp\left(-\frac{\sigma}{4t}[x^2+y^2]\right)\otimes_x\otimes_y u_0(x,y),\quad t>0.$$

Ignoring scaling by the function $\sigma/(4\pi t)$, we can write this result in the form

$$u(x_0,y_0)=\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right]\otimes_x\otimes_y u_0(x,y)$$

Thus, the field at time $t>0$ is given by the field at time $t=0$ convolved with the two-dimensional Gaussian function

$$\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right].$$

This result can, for example, be used to model the diffusion of light through a diffuser that generates multiple light scattering processes.

4 Diffusion from a Stochastic Source

For the case when

$$\left(\nabla^2-\sigma\frac{\partial}{\partial t}\right)u(\mathbf{r},t)=-S(\mathbf{r},t),\quad u(\mathbf{r},0)=0$$

the solution is

$$u(\mathbf{r},t)=G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}\otimes_t S(\mathbf{r},t),\quad t>0$$

If a source is introduced in terms of an impulse in time, then the 'system' will react accordingly and the diffuse for $t>0$. This is equivalent to introducing a source function of the form

$$S(\mathbf{r},t)=s(\mathbf{r})\delta(t).$$

The solution is then given by

$$u(\mathbf{r},t)=G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}s(\mathbf{r}),\quad t>0.$$

Observe that this solution is of the same form as the homogeneous case with initial condition $u(\mathbf{r},0)=u_0(\mathbf{r})$ and the solution for initial condition $u(\mathbf{r},0)=u_0(\mathbf{r})$ is given by

$$u(\mathbf{r},t)=G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}[s(\mathbf{r})+u_0(\mathbf{r})]$$

$$=G(\mid r\mid,t)\otimes_{\mathbf{r}}u_0(\mathbf{r})+n(\mathbf{r},t),\quad t>0$$

where

$$n(\mathbf{r},t)=G(\mid\mathbf{r}\mid,t)\otimes_{\mathbf{r}}s(\mathbf{r})$$

If s is a stochastic function (i.e. a random dependent variable characterised, at least, by a Probability Density Function (PDF) denoted by $\Pr[s(\mathbf{r})]$), then n will also be a stochastic function. Note, that for the time-independent source function $S(\mathbf{r})$, we can construct an inverse solution (see Appendix A) given by

$$u_0(\mathbf{r})=u(\mathbf{r},T)$$

$$+\sum_{n=1}^{\infty}\frac{(-1)^n}{n!}[(DT)^n\nabla^{2n}u(\mathbf{r},T)+D^{-1}\nabla^{2n-2}S(\mathbf{r})]$$

and that if S is a stochastic function, then the field u_0 can not be recovered because the functional form of S is not known. Thus, any error or noise associated with diffusion leads to the process being irreversible - a 'one-way' process. This, however, depends on the magnitude of the diffusivity D which for large values cancels out the effect of any noise, thus making the process reversible, an effect that is observable experimentally in the mixing of two highly viscous fluids, for example.

The inclusion of a stochastic source function provides us with a self-consistent introduction to another important concept in cryptology, namely 'confusion'. Taking, for example, the two-dimensional case, the field u is given by (with scaling)

$$u(x,y) = \frac{1}{4\pi t}\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right] \otimes_x \otimes_y u_0(x,y) + n(x,y).$$

We thus arrive at a basic model for the process of diffusion and confusion, namely

$$\textit{Output=Diffusion+Confusion.}$$

Here, *diffusion* involves the 'mixing' of the initial condition with a Gaussian function and *confusion* is compounded in the addition of a stochastic or noise function to the diffused output. The relative magnitudes of the two terms determines the dominating effect. As the noise function n increases in amplitude relative to the diffusion term, the output will become increasingly determined by the effect of confusion alone. In the equation above, this will occur as t increases since the magnitude of the diffusion term depends of the scaling factor $1/t$. This is illustrated in Figure 1 which shows the combined effect of diffusion and confusion for an image of the phrase

Confusion
+
Diffusion

as it is (from left to right and from top to bottom) progressively diffused (increasing values of t) and increasingly confused for a stochastic function n that is uniformly distributed.

The specific characteristics of the diffusion process considered here is determined by an approach that is based on modelling the system in terms of the diffusion equation; the result being determined by the convolution of the initial condition with a Gaussian function. The process of confusion is determined by the statistical characteristics of the stochastic function n, i.e. its PDF. Stochastic functions with different PDFs will exhibit different characteristics with regard to the level of confusion inherent in the process as applied. In the example given in Figure 1, uniformly distributed noise has been used. Gaussian or 'normal' distributed noise is more common by virtue of fact that noise in general is the result of an additive accumulation of many statistically independent random processes combining to form a normal or Gaussian distributed field. Knowledge of the noise field,

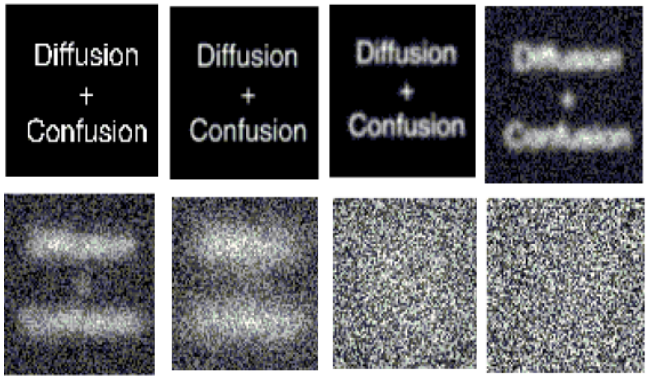

Figure 1: Progressive diffusion and confusion of an image (top-left) - from left to right and from top to bottom - for uniform distributed noise. The convolution is undertaken using the convolution theorem and a Fast Fourier Transform (FFT)

in particular, its PDF, provides a statistical approach to reconstructing the data based on the application of Bayesian estimation. For a Gaussian distributed noise field with a standard deviation of σ_n and a data field u_0 modelled in terms of Gaussian deviates with a standard deviation of σ_u, the estimate $\hat{u}_0(x,y)$ of $u_0(x,y)$ is given by [20], [21]

$$\hat{u}_0(x,y) = q(x,y) \otimes_x \otimes_y u(x,y)$$

where

$$q(x,y) = \frac{1}{(2\pi)^2}\int\int dk_x dk_y \exp(ik_x x)\exp(ik_y y) \times \frac{G^*(k_x,k_y)}{\mid G(k_x,k_y)\mid^2 + \sigma_n^2/\sigma_u^2}$$

and

$$G(k_x,k_y) = \frac{1}{4\pi t}\int\int dxdy \exp(ik_x x)\exp(ik_y y) \times \exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right]$$

Figure 2 illustrates the effect of applying this result to two digital outputs (using a Fast Fourier Transform) with low and high levels of noise, i.e. two cases for times t_1 (low) and $t_2 > t_1$ (high). This example shows the effect of

increasing the level of confusion that occurs with increasing time t on the output of the reconstruction clearly illustrating that it is not possible to recover u_0 to any degree of information assurance. This example demonstrates that the addition of a stochastic source function to an otherwise homogeneous diffusive process introduces a level of error (as time increases) from which is it not possible to recover the initial condition u_0. From a physical point of view, this is indicative of the fact that diffusive process are irreversible. From an information theoretic view point, Figure 2 illustrated that knowledge of the statistics of the stochastic field is not generally sufficient to recover the information we require. This is consistent with the basic principle of data processes - *Rubbish in Gives Rubbish Out*, i.e. given that

$$p(x,y) = \frac{1}{4\pi t}\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right],$$

the (Signal-to-Noise) ratio

$$\frac{\|p(x,y)\otimes_x \otimes_y u_0(x,y)\|}{\|n(x,y)\|}$$

tends to zero as t increases. In other words, the longer the time taken for the process of diffusion to occur, the more the output is dominated by confusion. This is consistent with all cases when the level of confusion is high and when the stochastic field used to generate this level of confusion is unknown (other than knowledge of its PDF). However, if the stochastic function has been synthesized[2] and is thus known *a priori*, then we can compute

$$u(x,y) - n(x,y) = \frac{1}{4\pi t}\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right]\otimes_x \otimes_y u_0(x,y)$$

from which u_0 can be computed via application of the convolution theorem to design an appropriate inverse filter.

5 Stochastic Fields

By considering the diffusion equation for a stochastic source, we have derived a basic model for the 'solution field' or 'output' $u(\mathbf{r},t)$ in terms of the initial condition or input $u_0(\mathbf{r})$ given by

$$u(\mathbf{r}) = p(\mathbf{r})\otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r})$$

where p is the PSF given by (with $a = \sigma/4t$)

$$\exp\left(-a \mid \mathbf{r} \mid^2\right)$$

and n - which is taken to denote noise - is a stochastic field, i.e. a random variable [22]. We shall now consider the principal properties of stochastic fields, considering the case where the fields are random variables that are functions of time t.

[2]The synthesis of stochastic functions is a principal issue in cryptology.

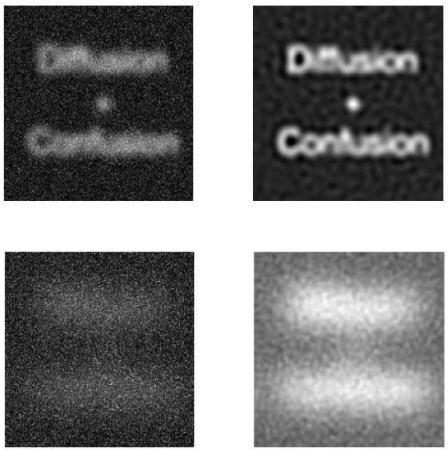

Figure 2: Bayesian reconstructions (right) for data (left) with low (above) and high (below) levels of confusion.

5.1 Independent Random Variables

Two random variables $f_1(t)$ and $f_2(t)$ are independent if their cross-correlation function is zero, i.e.

$$\int_{-\infty}^{\infty} f_1(t+\tau)f_2(\tau)d\tau = f_1(t)\odot f_2(t) = 0.$$

From the correlation theorem [20], it then follows that

$$F_1^*(\omega)F_2(\omega) = 0$$

where

$$F_1(\omega) = \int_{-\infty}^{\infty} f_1 \exp(-i\omega t)dt$$

and

$$F_2(\omega) = \int_{-\infty}^{\infty} f_1 \exp(-i\omega t)dt.$$

If each function has a PDF $\Pr[f_1(t)]$ and $\Pr[f_2(t)]$ respectively, the PDF of the function $f(t)$ that is the sum of $f_1(t)$ and $f_2(t)$ is given by the convolution of $\Pr[f_1(t)]$ and $\Pr[f_2(t)]$, i.e. the PDF of the function

$$f(t) = f_1(t) + f_2(t)$$

is given by [21], [22]

$$\Pr[f(t)] = \Pr[f_1(t)]\otimes_t \Pr[f_2(t)].$$

Further, for a number of statistically independent stochastic functions $f_1(t), f_2(t), ...,$ each with a PDF

$\Pr[f_1(t)], \Pr[f_2(t)], ...$, the PDF of the sum of these functions, i.e.

$$f(t) = f_1(t) + f_2(t) + f_3(t) + ...$$

is given by

$$\Pr[f(t)] = \Pr[f_1(t)] \otimes_t \Pr[f_2(t)] \otimes_t \Pr[f_1(t)] \otimes_t ...$$

These results can derived using the Characteristic Function [23]. For a strictly continuous random variable $f(t)$ with distribution function $P_f(x) = \Pr[f(t)]$ we define the expectation as

$$E(f) = \int_{-\infty}^{\infty} x P_f(x) dx,$$

which computes the mean value of the random variable, the Moment Generating Function as

$$E[\exp(-kf)] = \int_{-\infty}^{\infty} \exp(-kx) P_f(x) dx$$

which may not always exist and the Characteristic Function as

$$E[\exp(-ikf)] = \int_{-\infty}^{\infty} \exp(-ikx) P_f(x) dx$$

which will always exist. Observe that the moment generating function is the Laplace transform of P_f and the Characteristic Function is the Fourier transform of P_f. Thus, if $f(t)$ is a stochastic function which is the sum of N independent random variables $f_1(t), f_2(t), ..., f_N(t)$ with distributions $P_{f_1}(x), P_{f_2}(x), ..., P_{f_N}(x)$, then

$$f(t) = f_1(t) + f_2(t) + ... + f_N(t)$$

and

$$E[\exp(-ikf)] = E[\exp[-ik(f_1 + f_2 + ... + f_N)]$$

$$= E[\exp(-ikf_1)]E[\exp(-ikf_2)]...E[\exp(-ikf_N)]$$

$$= \hat{F}[P_{f_1}]\hat{\mathcal{F}}[P_{f_2}]...\hat{\mathcal{F}}[P_{f_N}]$$

where

$$\hat{\mathcal{F}} \equiv \int_{-\infty}^{\infty} dx \exp(ikx).$$

In other words, the Characteristic Function of the random variable $f(t)$ is the product of the Characteristic Functions for all random variables whose sum if $f(t)$. Using the convolution theorem for Fourier transforms, we then obtain

$$P_f(x) = \prod_{n=1}^{N} \otimes P_{f_n}(x) = P_{f_1}(x) \otimes_x P_{f_2}(x) \otimes_x ... \otimes_x P_{f_N}(x)$$

Further, we note that if f_1, f_2,... are all identically distributed then

$$E[\exp[-ik(f_1 + f_2 + ...)] = \left(\hat{\mathcal{F}}[P_{f_1}]\right)^N$$

and

$$P_f(x) = P_{f_1}(x) \otimes_x P_{f_1}(x) \otimes_x ...$$

5.2 The Central Limit Theorem

The Central Limit Theorem stems from the result that the convolution of two functions generally yields a function which is smoother than either of the functions that are being convolved. Moreover, if the convolution operation is repeated, then the result starts to look more and more like a Gaussian function - a normal distribution - at least in an approximate sense [24]. For example, suppose we have a number of independent random variables each of which is characterised by a distribution that is uniform. As we add more and more of these functions together, the resulting distribution is the given by convolving more and more of these (uniform) distributions. As the number of convolutions increases, the result tends to a Gaussian distribution. A proof of this theorem for a uniform distribution is given in Appendix B.

Figure 3 illustrates the effect of successively adding uniformly distributed but independent random times series (each consisting of 5000 elements) and plotting the resulting histograms (using 32 bins), i.e. given the discrete times series $f_1[i], f_2[i], f_3[i], f_4[i]$ for i=1 to 5000, Figure 3 shows the time series

$$s_1[i] = f_1[i]$$

$$s_2[i] = f_1[i] + f_2[i]$$

$$s_3[i] = f_1[i] + f_2[i] + f_3[i]$$

$$s_4[i] = f_1[i] + f_2[i] + f_3[i] + f_4[i]$$

and the corresponding 32-bin histograms of the signals $s_j, j = 1, 2, 3, 4$. Clearly as$j$ increases, the histogram starts to 'look' increasing normally distributed. Here, the uniformly distributed discrete time series $f_i, i = 1, 2, 3, 4$ have been computed using the uniform random number generator

$$f_{i+1} = f_i 7^7 \mathrm{mod} P$$

where $P = 2^{32} - 1$ is a Mersenne prime number, by using different four digit seeds f_0 in order to provide time series that are 'independent'.

The Central Limit Theorem has been considered specifically for the case of uniformly distributed independent

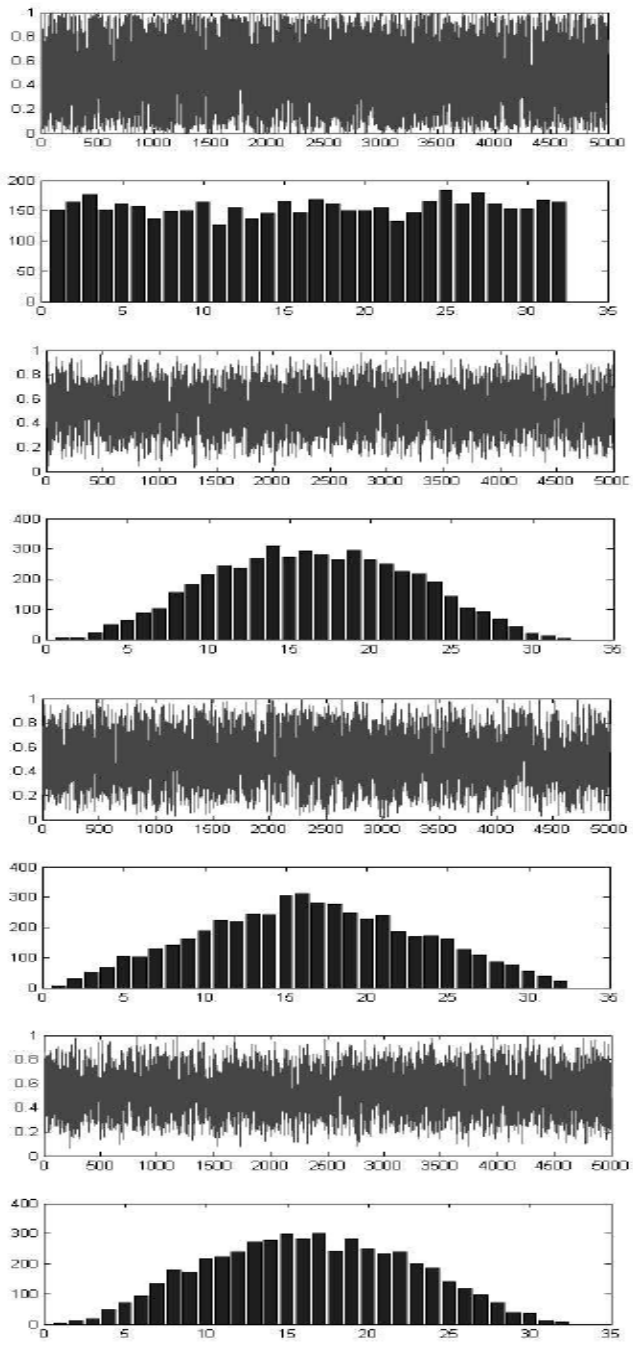

Figure 3: Illustration of the Central Limit Theorem. The top-left image shows plots of a 5000 element uniformly distributed time series and its histogram using 32 bins. The top-right image shows that result of adding two uniformly distributed and independent time series together and the 32 bin histogram. The bottom-left image is the result after adding three uniformly distributed times series and the bottom-right image is the result of adding four uniformly distributed times series.

random variables. However, in general, it is approximately applicable for all independent random variables, irrespective of their distribution. In particular, we note that for a standard normal (Gaussian) distribution given by

$$\text{Gauss}(x;\sigma,\mu) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

where

$$\int_{-\infty}^{\infty}\text{Gauss}(x)dx = 1$$

and

$$\int_{-\infty}^{\infty}\text{Gauss}(x)\exp(-ikx)dx = \exp(ik\mu)\exp\left(-\frac{\sigma^2k^2}{2}\right).$$

Thus, since

$$\text{Gauss}(x) \Longleftrightarrow \exp(ik\mu)\exp\left(-\frac{\sigma^2k^2}{2}\right)$$

then

$$\prod_{n=1}^{N}{}^{\otimes}\ \text{Gauss}(x) \Longleftrightarrow \exp(ikN\mu)\exp\left(-\frac{N\sigma^2k^2}{2}\right)$$

so that

$$\prod_{n=1}^{N}{}^{\otimes}\ \text{Gauss}(x) = \left(\frac{1}{2\pi N\sigma^2}\right)\exp\left[-\frac{1}{2N}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

In other words, the addition of Gaussian distributed fields produces a Gaussian distributed field.

6 Other 'Diffusion' Models

The diffusion model given by

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})$$

where (ignoring scaling)

$$p(\mathbf{r}) = \exp(-a \mid \mathbf{r} \mid^2)\otimes_{\mathbf{r}}$$

is specific to the case when we consider the homogeneous diffusion equation. This is an example of 'Gaussian diffusion' since the characteristic Point Spread Function is a Gaussian function. We can consider a number of different diffusing functions by exploring the effect of using different Point Spread Functions p. Although arbitrary changes to the PSF are inconsistent with classical diffusion, in cryptology we can, in principal, choose any PSF that is of value in 'diffusing' the data.

6.1 Diffusion by Noise

Given the classical diffusion/confusion model of the type

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r})$$

discussed above, we note that both the operator and the functional form of p are derived from solving a physical problem (using a Green's function solution) compounded in a particular PDE - diffusion or wave equation. We can use this basic model and consider a variety of PSFs as required; this include PSFs that are stochastic functions. Noise diffusion involves interchanging the roles of p and n, i.e. replacing $p(\mathbf{r})$ - a deterministic PSF - with $n(\mathbf{r})$ - a stochastic function. Thus, noise diffusion is compounded in the result

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + p(\mathbf{r})$$

or

$$u(\mathbf{r}) = n_1(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n_2(\mathbf{r})$$

where both n_1 and n_2 are stochastic function which may be of the same (i.e. have the same PDFs) or of different types (with different PDFs). This form of diffusion is not 'physical' in the sense that it does not conform to a physical model as defined by the diffusion or wave equation, for example. Here $n(\mathbf{r})$ can be any stochastic function (synthesized or otherwise).

The simplest form of noise diffusion is

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}).$$

The expected statistical distribution associated with the output of noise diffusion process is Gaussian. This can be shown if we consider u_0 to be a strictly deterministic function described by a sum of delta functions, equivalent to a binary stream in 1D or a binary image in 2D (discrete cases), for example. Thus if

$$u_0(\mathbf{r}) = \sum_i \delta^n(\mathbf{r} - \mathbf{r}_i)$$

then

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) = \sum_{i=1}^{N} n(\mathbf{r} - \mathbf{r}_i).$$

Now, each function $n(\mathbf{r}-\mathbf{r}_i)$ is just $n(\mathbf{r})$ shifted by $\mathbf{r}_i$ and will thus be identically distributed. Hence

$$\Pr[u(\mathbf{r})] = \Pr\left[\sum_{i=1}^{N} n(\mathbf{r} - \mathbf{r}_i)\right] = \prod_{i=1}^{N}{}_{\otimes} \Pr[n(\mathbf{r})]$$

and from the Central Limit Theorem, we can expect $\Pr[u(\mathbf{r})]$ to be normally distributed for large N. In particular, if

$$\Pr[n(\mathbf{r})] = \begin{cases} \frac{1}{X}, & \mid x \mid \leq X/2; \\ 0, & \text{otherwise} \end{cases}$$

then

$$\prod_{i=1}^{N}{}_{\otimes} \Pr[n(\mathbf{r})] \simeq \sqrt{\frac{6}{\pi X N}} \exp(-6x^2/XN).$$

This is illustrated in Figure 4 which shows the statistical distributions associated with a binary image, a uniformly distributed noise field and the output obtained by convolving the two fields together.

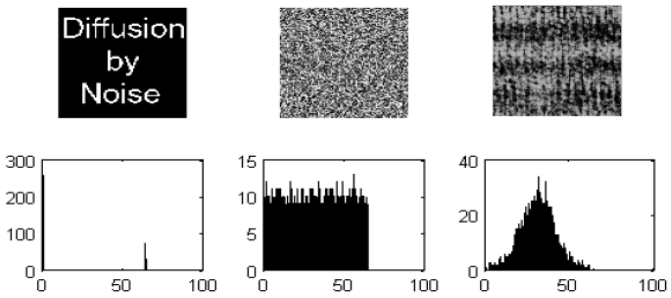

Figure 4: Binary image (top-left), uniformly distributed 2D noise field (top-centre), convolution (top-right) and associated histograms (bottom-left, -centre and -right respectively).

6.2 Diffusion of Noise

Given the equation

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r}),$$

if the diffusion by noise is based on interchanging p and n, then the diffusion of noise is based on interchanging u_0 and n. In effect, this means that we consider the initial field u_0 to be a stochastic function. Note that the solution to the inhomogeneous diffusion equation for a stochastic source $S(\mathbf{r}, t) = s(\mathbf{r})\delta(t)$ is

$$n(\mathbf{r}, t) = G(\mid \mathbf{r} \mid, t) \otimes_{\mathbf{r}} s(\mathbf{r})$$

and thus, n can be considered to be diffused noise. If we consider the model

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} n(\mathbf{r}),$$

then for the classical diffusion equation, the PSF is a Gaussian function. In general, given the convolution operation, p can be regarded as only one of a number of PSFs that can be considered in the 'production' of different stochastic fields u. This includes PSFs that define self-affine stochastic fields or random scaling fractals [27]-[29].

7 Information and Entropy

Consider a simple linear array such as a deck of eight cards which contains the ace of diamonds for example and where we are allowed to ask a series of sequential questions as to where in the array the card is. The first question we could ask is in which half of the array does the card occur which reduces the number of cards to four. The second question is in which half of the remaining four cards is the ace of diamonds to be found leaving just two cards and the final question is which card is it. Each successive question is the same but applied to successive subdivisions of the deck and in this way we obtain the result in three steps regardless of where the card happens to be in the deck. Each question is a binary choice and in this example, 3 is the minimum number of binary choices which represents the amount of information required to locate the card in a particular arrangement. This is the same as taking the binary logarithm of the number of possibilities, since $\log_2 8 = 3$. Another way of appreciating this result, is to consider a binary representation of the array of cards, i.e. 000,001,010,011,100,101,110,111, which requires three digits or bits to describe any one card. If the deck contained 16 cards, the information would be 4 bits and if it contained 32 cards, the information would be 5 bits and so on. Thus, in general, for any number of possibilities N, the information I for specifying a member in such a linear array, is given by

$$I = -\log_2 N = \log_2 \frac{1}{N}$$

where the negative sign is introduced to denote that information has to be acquired in order to make the correct choice, i.e. I is negative for all values of N larger than 1. We can now generalize further by considering the case where the number of choices N are subdivided into subsets of uniform size n_i. In this case, the information needed to specify the membership of a subset is given not by N but by N/n_i and hence, the information is given by

$$I_i = \log_2 P_i$$

where $P_i = n_i/N$ which is the proportion of the subsets. Finally, if we consider the most general case, where the subsets are non-uniform in size, then the information will no longer be the same for all subsets. In this case, we can consider the mean information given by

$$I = \sum_i P_i \log_2 P_i$$

which is the Shannon Entropy measure established in his classic works on information theory in the 1940s [30]. Information, as defined here, is a dimensionless quantity. However, its partner entity in physics has a dimension called 'Entropy' which was first introduced by Ludwig Boltzmann as a measure of the dispersal of energy, in a sense, a measure of disorder, just as information is a measure of order. In fact, Boltzmann's Entropy concept has the same mathematical roots as Shannon's information concept in terms of computing the probabilities of sorting objects into bins (a set of N into subsets of size n_i) and in statistical mechanics the Entropy is defined as [31]

$$E = -k \sum_i P_i \ln P_i$$

where k is Boltzmann's constant. Shannon's and Boltzmann's equations are similar. E and I have opposite signs, but otherwise differ only by their scaling factors and they convert to one another by $E = -(k \ln 2)I$. Thus, an Entropy unit is equal to $-k \ln 2$ of a bit. In Boltzmann's equation, the probabilities P_i refer to internal energy levels. In Shannon's equations P_i are not *a priori* assigned such specific roles and the expression can be applied to any physical system to provide a measure of order. Thus, information becomes a concept equivalent to Entropy and any system can be described in terms of one or the other. An increase in Entropy implies a decrease of information and vise versa. This gives rise to the fundamental conservation law: ˇ*The sum of (macroscopic) information change and Entropy change in a given system is zero.*

7.1 Entropy Based Information Extraction

In signal analysis, the Entropy is a measure of the lack of information about the exact information content of the signal, i.e. the value of f_i for a given i. Thus, noisy signals (and data in general) have a larger Entropy. The general definition for the Entropy of a system E is

$$E = -\sum_i P_i \ln P_i$$

where P_i is the probability that the system is in a state i. The negative sign is introduced because the probability is a value between 0 and 1 and therefore, $\ln P_i$ is a value between 0 and $-\infty$, but the Entropy is by definition, a positive value.

An Entropy based approach to the extraction of information from noise [32] can be designed using an Entropy measure defined in terms of the data f_i (rather than the PDF). A reconstruction for f_i is found such that

$$E = -\sum_i f_i \ln f_i$$

is a maximum which requires that $f_i > 0 \forall i$. Note that the function $x \ln x$ has a single local minimum value between 0 and 1 whereas the function $-x \ln x$ has a single local maximum value. It is a matter of convention as to whether a criteria of the type

$$E = \sum_i f_i \ln f_i$$

or

$$E = -\sum_i f_i \ln f_i$$

is used leading to (strictly speaking) a minimum or maximum Entropy criterion respectively. In some ways, the term 'Maximum Entropy' is misleading because it implies that we are attempting to recover information from noise with minimum information content and the term 'Minimum Entropy' conveys a method that is more consistent with the philosophy of what is being attempted, i.e. to recover useful and unambiguous information from a signal whose information content has been distorted or confused by (additive) noise. For example, suppose we input a binary stream into some time invariant linear system, where $\mathbf{f} = (...010011011011101...)$. Then, the input has an Entropy of zero since $0 \ln 0 = 0$ and $1 \ln 1 = 0$. We can expect the output of such a system to generate a new array of values (via the diffusion process) which are then perturbed (via the confusion process) through additive noise. The output $u_i = p_i \otimes_i f_i + n_i$ (where it is assumed that $u_i > 0 \forall i$ and $\otimes_i$ denotes the convolution sum over i) will therefore have an Entropy that is greater than 0. Clearly, as the magnitude of the noise increases, so, the value of the Entropy increases leading to greater loss of information on the exact state of the input (in terms of f_i, for some value of i being 0 or 1). With the inverse process, we ideally want to recover the input without any bit-errors. In such a hypothetical case, the Entropy of the restoration would be zero. In practice, we approach the problem in terms of an inverse solution that is based a Minimum Entropy criterion, i.e. find f_i such that

$$E = \sum_i f_i \ln f_i$$

is a minimum or for a continuous field $f(\mathbf{r})$ in n-dimensions, find f such that

$$E = \int f(\mathbf{r}) \ln f(\mathbf{r}) d^n\mathbf{r}$$

is a minimum.

Given that

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r}) + n(\mathbf{r})$$

where $\otimes_{\mathbf{r}}$ is the convolution integral over $\mathbf{r}$ we can write

$$\lambda \int \left([u(\mathbf{r}) - p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r})]^2 - [n(\mathbf{r})]^2 \right) d^n\mathbf{r} = 0$$

an equation that holds for any constant λ (the Lagrange multiplier). We can therefore write the equation for E as

$$E = -\int f(\mathbf{r}) \ln f(\mathbf{r}) d^n\mathbf{r}$$

$$+\lambda \int \left([u(\mathbf{r}) - p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r})]^2 - [n(\mathbf{r})]^2 \right) d^n\mathbf{r}$$

because the second term on the right hand side is zero anyway (for all values of λ). Given this equation, our problem is to find f such that the Entropy E is a maximum when

$$\frac{\partial E}{\partial f} = 0,$$

i.e. when

$$-1 - \ln f(\mathbf{r}) + 2\lambda[u(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r}) - p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r})] = 0$$

where $\odot_{\mathbf{r}}$ denotes the correlation integral over $\mathbf{r}$. Rearranging,

$$f(\mathbf{r}) = \exp[-1 + 2\lambda[u(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r}) - p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r})].$$

This equation is transcendental in f and as such, requires that f is evaluated iteratively, i.e.

$$[f(\mathbf{r})]^{n+1} = \exp[-1 + 2\lambda[u(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r})$$

$$-p(\mathbf{r}) \otimes_{\mathbf{r}} [f(\mathbf{r})]^n \odot_{\mathbf{r}} p(\mathbf{r}))]$$

The rate of convergence of this solution is determined by the value of the Lagrange multiplier given an initial estimate of $f(\mathbf{r})$, i.e. $[f(\mathbf{r})]^0$. However, the solution can be linearized by retaining the first two terms (the linear terms) in the series representation of the exponential function leaving us with the following result

$$f(\mathbf{r}) = 2\lambda[u(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r}) - p(\mathbf{r}) \otimes_{\mathbf{r}} f(\mathbf{r}) \odot_{\mathbf{r}} p(\mathbf{r})].$$

Using the convolution and correlation theorems, in Fourier space, this equation becomes

$$F(\mathbf{k}) = 2\lambda U(\mathbf{k})[P(\mathbf{k})]^* - 2\lambda \mid P(\mathbf{k}) \mid^2 F(\mathbf{k})$$

which after rearranging gives

$$F(\mathbf{k}) = \frac{U(\mathbf{k})[P(\mathbf{k})]^*}{\mid P(\mathbf{k}) \mid^2 + \frac{1}{2\lambda}}.$$

so that

$$f(\mathbf{r}) = \frac{1}{(2\pi)^n} \int_{-\infty}^{\infty} \frac{[P(\mathbf{k})]^* U(\mathbf{k})}{\mid P(\mathbf{k}) \mid^2 + \frac{1}{2\lambda}} \exp(i\mathbf{k} \cdot \mathbf{r}) d^n\mathbf{k}.$$

The cross Entropy or Patterson Entropy uses a criterion in which the Entropy measure

$$E = -\int d^n\mathbf{r} f(\mathbf{r}) \ln \left[\frac{f(\mathbf{r})}{w(\mathbf{r})} \right]$$

is maximized where $w(\mathbf{r})$ is some weighting function based on any available *a priori* information on $f(\mathbf{r})$. If the calculation above is re-worked using this definition of the cross Entropy, then we obtain the result

$$f(\mathbf{r}) = w(\mathbf{r})\exp(-1+2\lambda[u(\mathbf{r})\odot_{\mathbf{r}}p(\mathbf{r})-p(\mathbf{r})\otimes_{\mathbf{r}}f(\mathbf{r})\odot_{\mathbf{r}}p(\mathbf{r})]).$$

The cross Entropy method has a synergy with the Wilkinson test in which a PDF $P_n(x)$ say of a stochastic field $n(\mathbf{r})$ is tested against the PDF $P_m(x)$ of a stochastic field $m(\mathbf{r})$. A standard test to quantify how close the stochastic behaviour of n is to m (the null-hypothesis test) is to use the Chi-squared test in which we compute

$$\chi^2 = \int \left(\frac{P_n(x) - P_m(x)}{P_m(x)}\right)^2 dx.$$

The Wilkinson test uses the metric

$$E = -\int P_n(x)\ln\left(\frac{P_n(x)}{P_m(x)}\right) dx.$$

7.2 Entropy Conscious Confusion and Diffusion

From the point of view of designing an appropriate substitution cipher, the discussion above clearly dictates that the cipher $n[i]$ should be such that the Entropy of the ciphertext $u[i]$ is a maximum. This requires that a Pseudo Random Number Generation (PRNG) algorithm be designed that outputs a number stream whose Entropy is a maximum. There are a wide range of algorithms for generating pseudo random number streams that are continually being developed and improved upon for applications to random pattern generation [33] and image encryption [34], for example, and are usually based on some form of numerical iteration or 'round transformation'. However, irrespective of the application, a governing condition in the design of a PRNG is determined by the Information Entropy of the stream that is produced. Since the Information Entropy of the stream is defined as

$$E = \sum_{i=1}^{N} P_i \log_2 P_i$$

it is clear that the stream should have a PDF P_i that yields the largest possible values for E. Figure 5 shows a uniformly distributed and a Gaussian distributed random number stream consisting of 3000 elements and the characteristic discrete PDFs using 64-bins (i.e. for $N = 64$). The Information Entropy, which is computed directly from the PDFs using the expression for E given above, is always greater for the uniformly distributed field. This is to be expected because, for a uniformly distributed field, there is no bias associated with any particular numerical range and hence, no likelihood can be associated with a particular state. Hence, one of the underlying principals

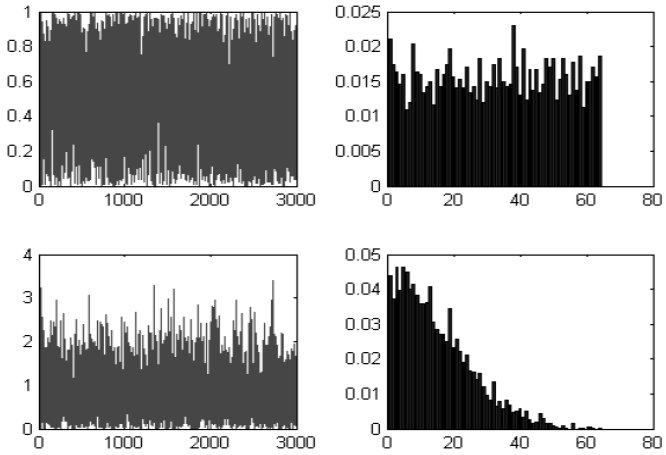

Figure 5: A 3000 element uniformly distributed random number stream (top left) and its 64-bin discrete PDF (top right) with $E = 4.1825$ and a 3000 element Gaussian distributed random number stream (bottom left) and its 64-bin discrete PDF (bottom right) with $E = 3.2678$.

associated with the design of a cipher $n[i]$ is that it should output a uniformly distributed sequence of random numbers. However, this does not mean that the ciphertext itself will be uniformly distributed since if

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[u_0(\mathbf{r})] \otimes_{\mathbf{r}} \Pr[n(\mathbf{r})].$$

This is illustrated in Figure 6 which shows 256-bin histograms for an 8-bit ASCII plaintext (the LaTeX file associated with this paper) $u_0[i]$, a stream of uniformly distributed integers $n[i]$, $0 \leq n \leq 255$ and the ciphertext $u[i] = u_0[i] + n[i]$. The spike associate with the plaintext histogram reflects the 'character' that is most likely to occur in the plaintext of a natural Indo-European language, i.e. a space with ASCII value 32. Although the distribution of the ciphertext is broader than the plaintext it is not as broad as the cipher and certainly not uniform. Thus, the Entropy of the ciphertext, although larger than the plaintext (in this example $E_{u_0} = 3.4491$ and $E_u = 5.3200$), the Entropy of the ciphertext is still less that then that of the cipher (in this example $E_n = 5.5302$). There are two ways in which this problem can be solved. The first method is to construct a cipher n with a PDF such that

$$P_n(x) \otimes_x P_{u_0}(x) = U(x)$$

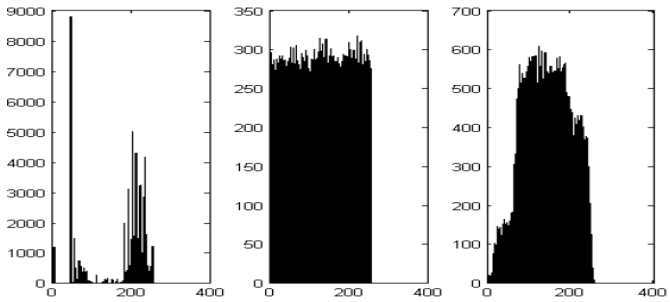

Figure 6: 256-bin histograms for an 8-bit ASCII plaintext $u_0[i]$ (left), a stream of uniformly distributed integers between 0 and 255 $n[i]$ (centre) and the substitution cipher $u[i]$ (right).

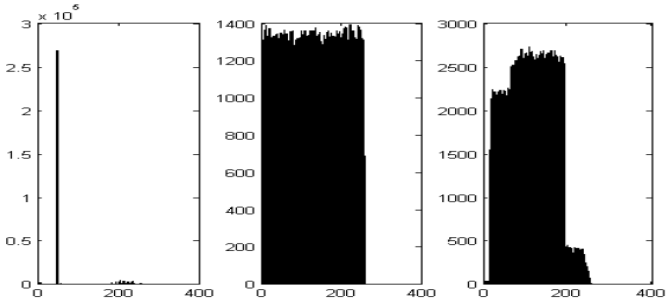

Figure 7: 256-bin histograms for an 8-bit ASCII plaintext $u_0[i]$ (left) after space-character padding, a stream of uniformly distributed integers between 0 and 255 $n[i]$ (centre) and the substitution cipher $u[i]$ (right).

where $U(x) = 1, \quad \forall x$. Then

$$P_n(x) = U(x) \otimes_x Q(x)$$

where

$$Q(x) = \hat{\mathcal{F}}^{-1}\left(\frac{1}{\hat{\mathcal{F}}[P_{u_0}(x)]}\right).$$

But this requires that the cipher is generated in such a way that its output conforms to an arbitrary PDF as determined by the plaintext to be encrypted. The second method is based on assuming that the PDF of all plaintexts will be of the form given in Figure 9 with a characteristic dominant spike associated with the number of spaces that occur in the plaintext[3] Noting that

$$P_n(x) \otimes_x \delta(x) = P_n(x)$$

then as the amplitude of the spike increases, the output increasingly approximates a uniform distribution; the Entropy of the ciphertext increases as the Entropy of the plaintext decreases. One simple way to implement this result is to pad-out the plaintext with spaces [4] The statistical effect of this is illustrated in Figure 7 where $E_{u_0} = 1.1615, \quad E_n = 5.5308$ and $E_u = 5.2537$.

8 Discussion

The purpose of this paper has been to introduce two of the most fundamental processes associated with cryptology, namely, diffusion and confusion. Diffusion has been considered via the properties associated with the homogeneous (classical) diffusion equation and the general Green's function solution. Confusion has been considered through the application of the inhomogeneous diffusion equation with a stochastic source function and it has been shown that

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r})$$

where p is a Gaussian Point Spread Function and n is a stochastic function.

Diffusion of noise involves the case when u_0 is a stochastic function. Diffusion by noise involves the use of a PSF p that is a stochastic function. If u_0 is taken to be deterministic information, then we can consider the processes of noise diffusion and confusion to be compounded in terms of the following:

Diffusion

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})$$

Confusion

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r})$$

Diffusion and Confusion

$$u(\mathbf{r}) = n_1(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n_2(\mathbf{r})$$

The principal effects of diffusion and confusion have been illustrated using various test images. This has been undertaken for visual purposes only but on the understanding that such 'effects' apply to fields in different dimensions in a similar way.

The statistical properties associated with independent random variables has also bee considered. One of the most significant results associated with random variable theory is compounded in the Central Limit Theorem.

[3] This is only possible provided the plaintext is an Indo-European alpha-numeric array and is not some other language or file format - a compressed image file, for example.

[4] Padding out a plaintext file with any character will provides a ciphertext with a broader distribution, the character @ (with an ASCII DEC of 64) providing a symmetric result, but space-character padding does not impinge on legibility.

When data is recorded, the stochastic term n, is often the result of many independent sources of noise due to a variety of physical, electronic and measuring errors. Each of these sources may have a well-defined PDF but if n is the result of the addition of each of them, then the PDF of n tends to be Gaussian distributed. Thus, Gaussian distributed noise tends to be common in the large majority of applications in which u is a record of a physical quantity.

In cryptology, the diffusion/confusion model is used in a variety of applications that are based on diffusion only, confusion only and combined diffusion/confusion models. One such example of the combined model is illustrated in Figure 8 which shows how one data field can be embedded in another field (i.e. how one image can be used to watermark another image using noise diffusion). In standard cryptography, one of the most conventional methods of encrypting information is through application of a confusion only model. This is equivalent to implementing a model where it is assumed that the PSF is a delta function so that

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r}).$$

If we consider the discrete case in one-dimension, then

$$u[i] = u_0[i] + n[i]$$

where $u_0[i]$ is the plaintext array or just 'plaintext' (a stream of integer numbers, each element representing a symbol associated with some natural language, for example), $n[i]$ is the 'cipher' and $u[i]$ is the 'ciphertext'. Methods are then considered for the generation of stochastic functions $n[i]$ that are best suited for the generation of the ciphertext. This is the basis for the majority of substitution ciphers where each value of each element of $u_0[i]$ is substituted for another value through the addition of a stochastic function $n[i]$, a function that should:

- include outputs that are zero in order that the spectrum of random numbers is complete[5]
- have a uniform PDF.

The conventional approach to doing this is to design appropriate Pseudo Random Number Generators (PRNGs) or pseudo chaotic ciphers. In either case, a cipher should be generated with maximum Entropy which is equivalent to ensuring that the cipher is a uniformly distributed stochastic field. However, it is important to appreciate that the statistics of a plaintext are not the same as those of the cipher when encryption is undertaken using a confusion only model; instead the statistics are determined by the convolution of the PDF of the plaintext with the PDF of the cipher. Thus, if

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[n(\mathbf{r})] \otimes_{\mathbf{r}} \Pr[u_0(\mathbf{r})].$$

One way of maximising the Entropy of u is to construct u_0 such that $\Pr[u_0(\mathbf{r})] = \delta(\mathbf{r})$. A simple and practical method of doing this is to pad the data u_0 with a single element that increase the data size but does not intrude on the legibility of the plaintext.

Assuming that the encryption of a plaintext u_0 is undertaken using a confusion only model, there exist the possibility of encrypting the ciphertext again. This is an example of double encryption, a process that can be repeated an arbitrary number of times to give triple and quadruple encrypted outputs. However, multiple encryption procedures in which

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n_1(\mathbf{r}) + n_2(\mathbf{r}) + ...$$

where n_1, n_2,... are different ciphers, each consisting of uniformly distributed noise, suffer from the fact that the resultant cipher is normally distributed because, from the Central Limit Theorem

$$\Pr[n_1 + n_2 + ...] \sim \text{Gauss}(x).$$

For this reason, multiple encryption systems are generally not preferable to single encryption systems. A notable example is the triple DES (Data Encryption Standard) or DES3 system [35] that is based on a form of triple encryption and originally introduced to increase the key length associated with the generation of a single cipher n_1. DES3 was endorsed by the National Institute of Standards and Technology (NIST) as a temporary standard to be used until the Advanced Encryption Standard (AES) was completed in 2001 [36].

The statistics of an encrypted field formed by the diffusion of u_0 (assumed to be a binary field) with noise produces an output that is Gaussian distributed, i.e. if

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})] \sim \text{Gauss}(x).$$

Thus, the diffusion of u_0 produces an output whose statistics are not uniform but normally distributed. The Entropy of a diffused field using uniformly distributed noise

[5]The Enigma cipher, for example, suffered from a design fault with regard to this issue in that a letter could not reproduce its self - $u[i] \neq u_0[i] \forall i$. This provided a small statistical bias which was nevertheless significant in the decryption of Enigma ciphers.

is therefore less than the Entropy of a confused field. It is for this reason, that a process of diffusion should ideally be accompanied by a process of confusion when such processes are applied to cryptology in general.

The application of noise diffusion for embedding or watermarking one information field in another is an approach that has a range of applications including diffusion only cryptology for applications to low resolution print security for example which is discussed later on in this work.

Since the diffusion of noise by a deterministic PSF produces an output whose statistics tend to be normally distributed, such fields are not best suited for encryption. However, this process is important in the design of stochastic fields that have important properties for the camouflage of encrypted data. This includes the generation of random fractal fields and the use of methods such a fractal modulation for covert data communications.

9 Lossy Watermarking Method

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 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).

9.1 Binary Image Watermarking

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 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, 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. Thus, binarization of a stochastically diffused data field is entirely irreversible.

9.2 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

$$\begin{aligned} c(x,y) &= m(x,y) \otimes_x \otimes_y p(x,y) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} m(x - x_i, y - y_j). \end{aligned}$$

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

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

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

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

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 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.

9.3 Principal Algorithms

The principal algorithms associated with the application of stochastic diffusion for watermarking with ciphers 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 must ideally be the same size as the plaintext image or else resized accordingly. However, in resembling 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).

9.4 StegoText

StegoText is a prototype tool designed using MATLAB to examine the applications to which stochastic diffusion can be used. A demonstration version of the system is available at http://eleceng.dit.ie/arg/downloads/Stegocrypt which has been designed with a simple Graphical User Interface as shown in Figure 8 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 image	*Output:* Decrypted watermark
Operation: Encrypt by clicking on buttom E (for Encrypt)	*Operation:* Decrypt by clicking on button D (for Dycrypt)

The PIN (Personal Identity Number) can be an numerical string with upto 16 elements. In principal, any existing encryption algorithm, application or system can be used to generate the cipher required by *StegoText* by encrypting an image composed of random noise. The output is 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

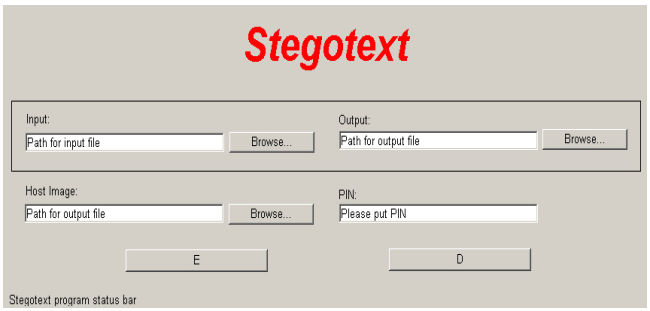

Figure 8: Graphical User Interface for *Stegotext* software system.

given system. In this way, *StegoText* can be used in conjunction with any existing encryption standard.

The principal aim of *StegoText* 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'.

9.5 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 [37], [38]. 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* [39]. 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 *Stegotext* GUI shown in Figure 8, this involves using the same file for the *Input* and *Host Image*. An example of this is shown in Figure 9 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 qualification of the holder, the date and signature, for example. Figure 10 shows the 'Coat of Arms' and the signatures associated with this decrypt which have been cut from the original decrypt given in Figure 9. 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 9: Certificate with binary watermark (left) and decrypt (right).

Figure 10: 'Coat of Arms' (left) and signatures (right) of decrypt given in Figure 9.

10 Lossless Watermarking Method

The method discussed in the previous section is suitable for document authentication, but the lossy nature of the reconstruction generated through binarisation of the cipher, illustrated in Figure 9, is not suitable for full colour images. In this section we introduce an algorithms for hiding grey scale image in a colour image and full colour images using three host colour images. Figure 11 shows a block diagram for hiding an encrypted 8-bit grey level image in a 24-bit colour image and Figure 12 shows the equivalent block diagram for hiding encrypted 24-bit colour image in three 24-bit colour host images. In the latter case, the same approach is used applied to each colour component of the colour image. Referring to Figure 11, stochastic diffusion is used to encrypt an 8-bit grey level image into a 24-bit colour host image with a near perfect decrypt. In this scheme, the cipher is not binarised but is converted into binary form. The first and second Least Significant Bits (LSBs) are ignored and the third and fourth bits are embedded into the two LSBs of the host image's red channel. Similarly, the 5^{th} and 6^{th} bits are embedded into the two LSBs of the host image's green channel, and finally the 7^{th} and 8^{th} bits are embedded into the two LSBs of the host image's blue channel. The inverse process is based on extracting the relevant bits from the associated channels with the first and second bits being set to zero. The extracted bits are then used to re-generate the original cipher and the reconstruction obtained by correlation with the original noise field.

Figure 13 shows an example of the method based on the block diagram given in Figure 12 using the MATLAB code given in Appendix C. The three 24-bit colour host images after application of the embedding process are given in Figure 14.

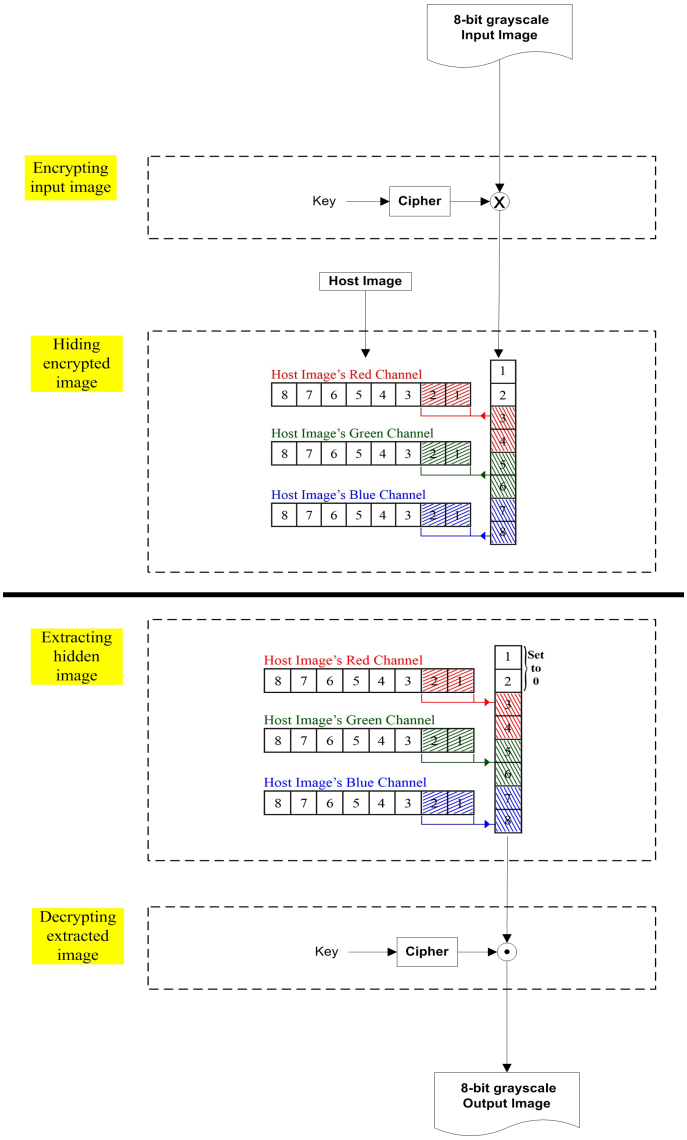

Figure 11: Block Diagram for hiding an encrypted 8-bit grey level image in a 24-bit colour host image.

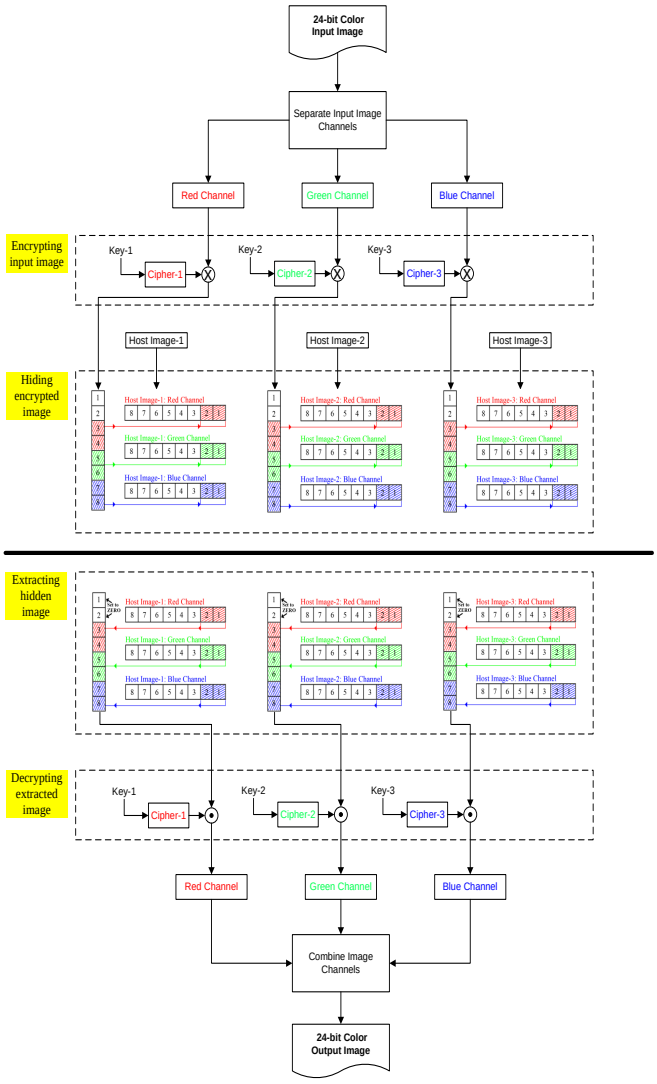

Figure 12: Block Diagram for hiding an encrypted 24-bit colour image in three 24-bit colour host images.

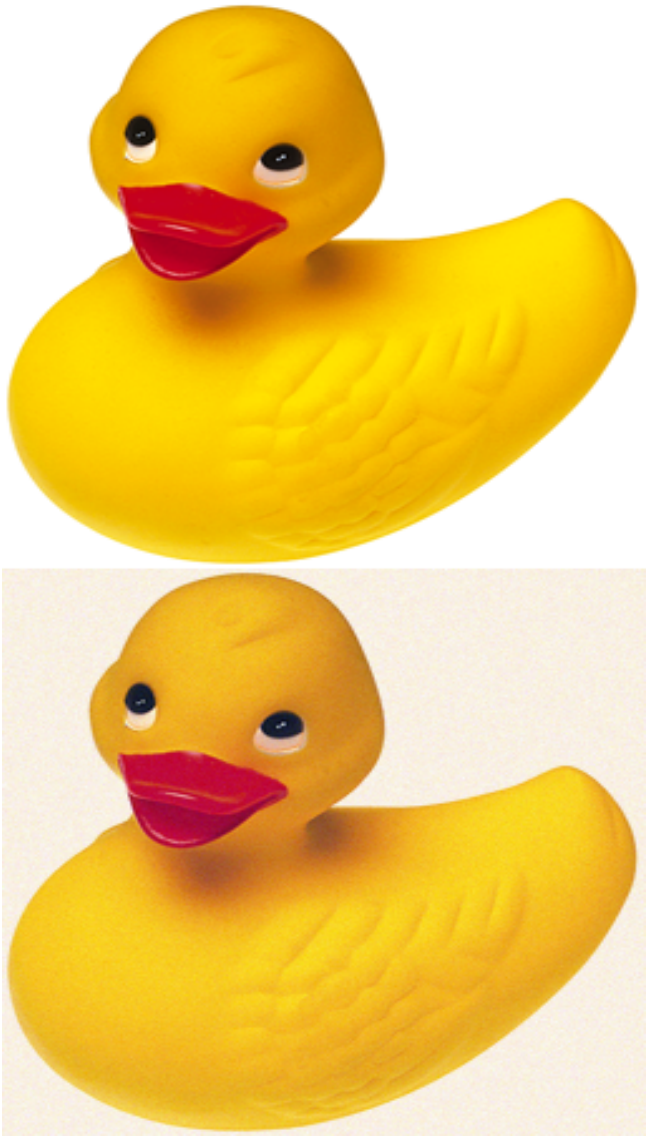

Figure 13: Orginal Image (above) and reconstructed image after decryption (below).

Figure 14: Host images used to hide the image given in Figure 13 after embedding the ciphers.

11 Conclusions

This paper has focused on the application of stochastic diffusion for transmitting e-documents and digital images over the internet in such a way that encrypted information can be communicated covertly and the information authenticated. The use of the Internet to transfer documents as image attachments has and continues to grow rapidly. 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 (as discussed in Section 9) provides a cryptographically secure solution. This is because binarization is an entirely one-way process. Thus, although the watermark may be removed from the covertext image, it can not be decrypted without the recipient having access to the correct cryptographically secure algorithm and key. The approach discussed in Section 9.4 and the StegoText system currently available has a range of application for e-document authentication. 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. *StegoText* provides the facility to issue electronic certificates (in addition or in substitution to a hardcopy) which can then be authenticated as discussed in Section 9.4. 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. [40].

In this paper, the method of stochastic diffusion has been extended to hide 24-bit colour images in a set of three 24-bit colour images. This provides a lossless method of encrypting and covertly communicating 24-bit colour images over the Internet as required and as illustrated in Section 10. The applications to which stochastic diffusion can be applied are numerous and, coupled with appropriate key-exchange protocols, provides a generic method of encrypting and hiding digital image information.

Appendix A: Inverse Diffusion

Suppose we consider the homogeneous diffusion problem defined by the equation

$$D\nabla^2 u(\mathbf{r},t) - \frac{\partial}{\partial t}u(\mathbf{r},t) = 0, \quad u(\mathbf{r},0) = u_0(\mathbf{r})$$

with the solution

$$u(\mathbf{r},\tau)=\frac{1}{D}G(\mid \mathbf{r} \mid,t)\otimes_{\mathbf{r}} u_0(\mathbf{r}),\quad t>0.$$

If we record a diffused field u after some time $t=T$, is it possible to reconstruct the field at time $t=0$, i.e. to solve the inverse problem or de-diffuse the field measured? We can express $u(\mathbf{r},0)$ in terms of $u(\mathbf{r},T)$ using the Taylor series

$$u_0(\mathbf{r})\equiv u(\mathbf{r},0)=u(\mathbf{r},T)+\sum_{n=1}^{\infty}\frac{(-1)^n}{n!}T^n\left[\frac{\partial^n}{\partial t^n}u(\mathbf{r},t)\right]_{t=T}.$$

Now, from the diffusion equation

$$\frac{\partial^2 u}{\partial t^2}=D\nabla^2\frac{\partial u}{\partial t}=D^2\nabla^4 u,$$

$$\frac{\partial^3 u}{\partial t^3}=D\nabla^2\frac{\partial^2 u}{\partial t^2}=D^3\nabla^6 u$$

and so on. Thus, in general we can write

$$\left[\frac{\partial^n}{\partial t^n}u(\mathbf{r},t)\right]_{t=T}=D^n\nabla^{2n}u(x,y,T).$$

Substituting this result into the series for u_0 given above, we obtain

$$u_0(\mathbf{r})=u(\mathbf{r},T)+\sum_{n=1}^{\infty}\frac{(-1)^n}{n!}(DT)^n\nabla^{2n}u(\mathbf{r},T).$$

For a time-independent source function $S(\mathbf{r})$ the equivalent solution to the equation

$$D\nabla^2 u(\mathbf{r},t)-\frac{\partial}{\partial t}u(\mathbf{r},t)=--S(\mathbf{r}),\quad u(\mathbf{r},0)=u_0(\mathbf{r})$$

is then given by

$$u_0(\mathbf{r})=u(\mathbf{r},T)$$
$$+\sum_{n=1}^{\infty}\frac{(-1)^n}{n!}[(DT)^n\nabla^{2n}u(\mathbf{r},T)+D^{-1}\nabla^{2n-2}S(\mathbf{r})]$$

Appendix B: Proof by Induction of the Central Limit Theorem for a Uniformly Distributed Stochastic Function

We consider the effect of applying multiple convolutions of the uniform distribution

$$P(x)=\begin{cases}\frac{1}{X}, & \mid x \mid\leq X/2;\\ 0, & \text{otherwise}\end{cases}$$

and show that

$$\prod_{n=1}^{N}{}_{\otimes}\ P_n(x)\equiv P_1(x)\otimes_x P_2(x)\otimes_x ...\otimes_x P_N(x)$$

$$\simeq\sqrt{\frac{6}{\pi N}}\exp(-6x^2/XN)$$

where $P_n(x)=P(x),\ \ \forall n$ and N is large. This result is based on considering the effect of multiple convolutions in Fourier space (through application of the convolution theorem) and then working with a series representation of the result.

The Fourier transform of $P(x)$ is given by

$$\widetilde{P}(k)=\int_{-\infty}^{\infty}P(x)\exp(-ikx)dx$$

$$=\int_{-X/2}^{X/2}\frac{1}{X}\exp(-ikx)dx=\mathrm{sinc}(kX/2)$$

where $\mathrm{sinc}(x)=\sin(x)/x$ - the 'sinc' function. Thus,

$$P(x)\Longleftrightarrow\mathrm{sinc}(kX/2)$$

where $\Longleftrightarrow$ denotes transformation into Fourier space, and from the convolution theorem in follows that

$$Q(x)=\prod_{n=1}^{N}{}_{\otimes}\ P_n(x)\Longleftrightarrow\mathrm{sinc}^N(kX/2).$$

Using the series expansion of the sin function for an arbitrary constant α,

$$\mathrm{sinc}(\alpha k)$$
$$=\frac{1}{\alpha k}\left(\alpha k-\frac{1}{3!}(\alpha k)^3+\frac{1}{5!}(\alpha k)^5-\frac{1}{7!}(\alpha k)^7+\ldots\right)$$
$$=1-\frac{1}{3!}(\alpha k)^2+\frac{1}{5!}(\alpha k)^4-\frac{1}{7!}(\alpha k)^6+\ldots$$

The N^{th} power of $\mathrm{sinc}(\alpha k)$ can be written in terms of a binomial expansion giving

$$\mathrm{sinc}^N(\alpha k)=\left(1-\frac{1}{3!}(\alpha k)^2+\frac{1}{5!}(\alpha k)^4-\frac{1}{7!}(\alpha k)^6+\ldots\right)^N$$
$$=1-N\left(\frac{1}{3!}(\alpha k)^2-\frac{1}{5!}(\alpha k)^4+\frac{1}{7!}(\alpha k)^6-\ldots\right)$$
$$+\frac{N(N-1)}{2!}\left(\frac{1}{3!}(\alpha k)^2-\frac{1}{5!}(\alpha k)^4+\frac{1}{7!}(\alpha k)^6-\ldots\right)^2$$
$$-\frac{N(N-1)(N-2)}{3!}$$
$$\times\left(\frac{1}{3!}(\alpha k)^2-\frac{1}{5!}(\alpha k)^4+\frac{1}{7!}(\alpha k)^6-\ldots\right)^3+\ldots$$
$$=1-N\frac{\alpha^2k^2}{3!}+N\frac{\alpha^4k^4}{5!}-k\frac{\alpha^6k^6}{7!}-\ldots$$

$$+\frac{N(N-1)}{2!}\left(\frac{\alpha^4k^4}{(3!)^2}-2\frac{\alpha^6k^6}{3!5!}+\ldots\right)$$
$$-\frac{N(N-1)(N-2)}{3!}\left(\frac{\alpha^6k^6}{(3!)^3}+\ldots\right)$$
$$=1-\frac{N}{3!}\alpha^2k^2+\left(\frac{N}{5!}\alpha^4+\frac{N(N-1)}{2!(3!)^2}\alpha^4\right)k^4$$
$$-\left(\frac{N}{7!}\alpha^6+\frac{N(N-1)}{3!5!}\alpha^6+\frac{N(N-1)(N-2)}{3!(3!)^3}\alpha^6\right)k^6+\ldots$$

Now the series representation of the exponential (for an arbitrary positive constant c) is

$$\exp(-ck^2)=1-ck^2+\frac{1}{2!}c^2k^4-\frac{1}{3!}c^3k^6+\ldots$$

Equating terms involving k^2, k^4 and k^6 it is clear that (evaluating the factorials),

$$c=\frac{1}{6}N\alpha^2,$$
$$\frac{1}{2}c^2=\left(\frac{1}{120}N+\frac{1}{72}N(N-1)\right)\alpha^4$$

or

$$c^2=\left(\frac{1}{36}N^2-\frac{1}{90}N\right)\alpha^4,$$

and

$$\frac{1}{6}c^3=$$
$$\left(\frac{1}{5040}N+\frac{1}{720}N(N-1)+\frac{1}{1296}N(N-1)(N-2)\right)\alpha^6$$

or

$$c^3=\left(\frac{1}{216}N^3-\frac{1}{1080}N^2+\frac{1}{2835}N\right)\alpha^6.$$

Thus, by deduction, we can conclude that

$$c^n=\left(\frac{1}{6}N\right)^n\alpha^{2n}+O(N^{n-1}\alpha^{2n}).$$

Now, for large N, the first term in the equation above dominates to give the following approximation for the constant c,

$$c\simeq\frac{1}{6}N\alpha^2.$$

We have therefore shown that the N^{th} power of the $\text{sinc}(\alpha k)$ function approximates to a Gaussian function (for large N), i.e.

$$\text{sinc}^N(\alpha k)\simeq\exp\left(-\frac{1}{6}N\alpha^2k^2\right).$$

Thus, if $\alpha=\frac{X}{2}$, then

$$Q(x)\Longleftrightarrow\exp\left(-\frac{X}{24}Nk^2\right)$$

approximately. The final part of the proof is therefore to Fourier invert the function $\exp(-XNk^2/24)$, i.e. to compute the integral

$$I=\frac{1}{2\pi}\int_{-\infty}^{\infty}\exp\left(-\frac{1}{24}XNk^2\right)\exp(ikx)dk.$$

Now,

$$I=\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathrm{e}^{-\left[\left(\sqrt{\frac{XN}{24}}k-\sqrt{\frac{24}{XN}}\frac{ix}{2}\right)^2-\frac{6x^2}{XN}\right]}dk$$
$$=\frac{1}{\pi}\sqrt{\frac{6}{XN}}\mathrm{e}^{-\frac{6x^2}{XN}}\int_{-\infty+ix\sqrt{\frac{6}{XN}}}^{\infty+ix\sqrt{\frac{6}{XN}}}\mathrm{e}^{-y^2}dy$$

after making the substitution

$$y=\sqrt{\frac{XN}{6}}\frac{k}{2}-ix\sqrt{\frac{6}{XN}}.$$

By Cauchy's theorem

$$I=\frac{1}{\pi}\sqrt{\frac{6}{XN}}\mathrm{e}^{-\frac{6x^2}{XN}}\int_{-\infty}^{\infty}\mathrm{e}^{-z^2}dz=\sqrt{\frac{6}{\pi XN}}\mathrm{e}^{-\frac{6x^2}{XN}}$$

where we have use the result

$$\int_{-\infty}^{\infty}\exp(-y^2)dy=\sqrt{\pi}.$$

Thus, we can write

$$Q(x)=\prod_{n=1}^{N}{}^{\otimes}\,P_n(x)\simeq\sqrt{\frac{6}{\pi XN}}\exp[-6x^2/(XN)]$$

for large N.

Appendix C: MATLAB Code for Loss-less Watermarking Method

```
function []  = CIE( ImageName )
% This function - Covert Image Encryption (CIE) -
% inputs a 24-bit color image and encrypts it
% using the Stochastic Diffusion method.

% Read input image
InImage = imread(ImageName);
row = size(InImage,1);
col = size(InImage,2);
InImage = double(InImage);
%-------------------------
```

```
% Generate the noise field
% using Matlab's rand function
NoiseImageR = rand(row,col);
NoiseImageG = rand(row,col);
NoiseImageB = rand(row,col);

NR = NoiseImageR;
NG = NoiseImageG;
NB = NoiseImageB;
% -------------------------

% Convolve the input image with the
% noise image using a 2D FFT
% with pre-conditioning
mR = PreCondition(NoiseImageR);
mG = PreCondition(NoiseImageG);
mB = PreCondition(NoiseImageB);
% ----------------------------
% Encrypt the Red Channel
CR=ifft2(fft2(mR).*fft2(InImage(:,:,1)) );
% Encrypt the Green Channel.
CG=ifft2(fft2(mG).*fft2(InImage(:,:,2)) );
% Encrypt the Blue Channel
CB=ifft2(fft2(mB).*fft2(InImage(:,:,3)) );

% Normalize Cipher Images to range 0:255.
CR = Normalize(CR) .* 255;
CG = Normalize(CG) .* 255;
CB = Normalize(CB) .* 255;
% ------------------------
CR = uint8(CR);
CG = uint8(CG);
CB = uint8(CB);
% ------------------------

% Embed cipher images into three
% named cover images:
%
% cover1.bmp
% cover2.bmp
% cover3.bmp

% Embed red channel cipher into cover image 1
CoverImage1 = imread('cover1.bmp');
CoverImage1 = imresize(CoverImage1 , [row col]);
figure(1);
subplot(1,2,1), imshow(CoverImage1),
title('Cover Image1 before embedding');
for i = 1 : size(CoverImage1,1)
    for j = 1 : size(CoverImage1,2)

        CoverImage1(i,j,1) =
        bitand( CoverImage1(i,j,1) , 252 );
        CoverImage1(i,j,1) =
        bitor( CoverImage1(i,j,1),
        bitand(bitshift(CR(i,j),-2),3) );

        CoverImage1(i,j,2) =
        bitand( CoverImage1(i,j,2) , 252 );
        CoverImage1(i,j,2) =
        bitor( CoverImage1(i,j,2),
        bitand(bitshift(CR(i,j),-4),3) );

        CoverImage1(i,j,3) =
        bitand( CoverImage1(i,j,3) , 252 );
        CoverImage1(i,j,3) =
        bitor( CoverImage1(i,j,3),
        bitand(bitshift(CR(i,j),-6),3));

    end
end
subplot(1,2,2), imshow(CoverImage1),
title('Cover Image1 after Embedding');
% ----------------------------------
% Embed green channel cipher into Cover Image 2
CoverImage2 = imread('cover2.bmp');
CoverImage2 = imresize(CoverImage2 , [row col]);
figure(2);
subplot(1,2,1), imshow(CoverImage2),
title('Cover Image2 before Embedding');

for i = 1 : size(CoverImage2,1)
    for j = 1 : size(CoverImage2,2)

        CoverImage2(i,j,1) =
        bitand( CoverImage2(i,j,1) , 252 );
        CoverImage2(i,j,1) =
        bitor( CoverImage2(i,j,1),
        bitand(bitshift(CG(i,j),-2),3) );

        CoverImage2(i,j,2) =
        bitand( CoverImage2(i,j,2) , 252 );
        CoverImage2(i,j,2) =
        bitor( CoverImage2(i,j,2),
        bitand(bitshift(CG(i,j),-4),3) );

        CoverImage2(i,j,3) =
        bitand( CoverImage2(i,j,3) , 252 );
        CoverImage2(i,j,3) =
        bitor( CoverImage2(i,j,3),
        bitand(bitshift(CG(i,j),-6),3));

    end
end
subplot(1,2,2), imshow(CoverImage2),
```

```
title('Cover Image2 after Embedding');
% ------------------------------------
% Embed blue channel cipher into Cover Image 3
CoverImage3 = imread('cover3.bmp');
CoverImage3 = imresize(CoverImage3 , [row col]);
figure(3);
subplot(1,2,1), imshow(CoverImage3),
title('Cover Image3 before Embedding');

for i = 1 : size(CoverImage3,1)
    for j = 1 : size(CoverImage3,2)

        CoverImage3(i,j,1) =
        bitand( CoverImage3(i,j,1) , 252 );

        CoverImage3(i,j,1) =
        bitor( CoverImage3(i,j,1),
        bitand(bitshift(CB(i,j),-2),3) );

        CoverImage3(i,j,2) =
        bitand( CoverImage3(i,j,2) , 252 );
        CoverImage3(i,j,2) =
        bitor( CoverImage3(i,j,2),
        bitand(bitshift(CB(i,j),-4),3) );

        CoverImage3(i,j,3) =
        bitand( CoverImage3(i,j,3) , 252 );
        CoverImage3(i,j,3) =
        bitor( CoverImage3(i,j,3),
        bitand(bitshift(CB(i,j),-6),3));

    end
end
subplot(1,2,2), imshow(CoverImage3),
title('Cover Image3 after Embedding');
% ------------------------------------

% Extract the hidden ciphers from cover images
% Extract red channel cipher from cover image 1
for i = 1 : size(CoverImage1,1)
    for j = 1 : size(CoverImage1,2)

    R = bitand( CoverImage1(i,j,1), 3);
    G = bitand( CoverImage1(i,j,2), 3);
    B = bitand( CoverImage1(i,j,3), 3);
    ExImageR(i,j) = bitor( bitor(bitshift(R,2),
    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageR = uint8(ExImageR);
% -------------------------
%
% Extract green channel cipher
% from cover image 2
for i = 1 : size(CoverImage2,1)
    for j = 1 : size(CoverImage2,2)

    R = bitand( CoverImage2(i,j,1), 3);
    G = bitand( CoverImage2(i,j,2), 3);
    B = bitand( CoverImage2(i,j,3), 3);
    ExImageG(i,j) = bitor( bitor(bitshift(R,2),
    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageG = uint8(ExImageG);
% -------------------------
%
% Extract blue channel cipher
% from cover image 3
for i = 1 : size(CoverImage3,1)
    for j = 1 : size(CoverImage3,2)

    R = bitand( CoverImage3(i,j,1), 3);
    G = bitand( CoverImage3(i,j,2), 3);
    B = bitand( CoverImage3(i,j,3), 3);
    ExImageB(i,j) = bitor( bitor(bitshift(R,2),
    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageB = uint8(ExImageB);
% -------------------------

% Correlate the Extracted ciphers with the
% noise field using a 2D FFT
ExImageR = double(ExImageR);
ExImageG = double(ExImageG);
ExImageB = double(ExImageB);

PlainImR =
ifft2( conj(fft2(NR)) .* fft2(ExImageR) );

PlainImG =
ifft2( conj(fft2(NG)) .* fft2(ExImageG) );

PlainImB =
ifft2( conj(fft2(NB)) .* fft2(ExImageB) );

% Normalize images to raneg 0:255
PlainImR = Normalize(PlainImR) .* 255;
PlainImG = Normalize(PlainImG) .* 255;
PlainImB = Normalize(PlainImB) .* 255;
%-------------------------------------
```

```
Result(:,:,1) = PlainImR;
Result(:,:,2) = PlainImG;
Result(:,:,3) = PlainImB;
Result = uint8(Result);
imwrite(Result,'Output_Color.bmp');

figure(4);
subplot(1,2,1), imshow(uint8(InImage)),
title('Input Image before Encryption');

subplot(1,2,2), imshow(Result),
title('Output Image after Decryption');

end

%-------------------------------------

function [ x ] = Normalize( mat )
% Function to normalise images

MAX = max(mat(:)); MIN = min(mat(:));

for i = 1:size(mat,1)
    for j = 1:size(mat,2)
    x(i,j) = ((mat(i,j) - MIN )/(MAX - MIN));
    end
end

return;
end

%-----------------------------------

function [ m ] = PreCondition( arr )
% Pre-conditioning function

arrF = fft2(arr);
for i = 1:size(arrF,1)
    for j = 1:size(arrF,2)
        if abs(arrF(i,j)) == 0
            M(i,j) = arrF(i,j);
        else

    M(i,j)=
    arrF(i,j)/(abs(arrF(i,j))*abs(arrF(i,j)));

        end
    end
end
m = ifft2(M);

return;
end
```

Acknowledgments

The authors are grateful for the support of the Science Foundation Ireland.

References

[1] Xiaolu, Li., Zhi, Qi., Zhiqiang, Yang, and Jun, Kong., "A Novel Hidden Transmission of Biometric Images Base on Choas and Image Content", First International Workshop on Education Technology and Computer Science, 2009

[2] Jun, Kong., Hongru, Jia., Xiaolu. Li., and Zhi, Qi., "A Novel Content-based Information Hiding Scheme", International Conference on Computer Engineering and Technology, 2009

[3] Che-Wei, Lee and Wen-Hasiang, Tsai, "A New Steganographic Method Based on Information Sharing via PNG Images", IEEE transactions, 2010.

[4] Chen Uuefen, Lin Junhuan, Zhang Shiqing, and Chen Caiming, "Double Random Scrambling Algorithm Based on Subblocks for Image Hiding", International Conference on Computer and Communication Technologies in Agriculture Engineering, 2010.

[5] Ptitsyn, N. V., Blackledge, J. M. and Chernenky V. M., "Deterministic Chaos in Digital Cryptography", Proceedings of the First IMA Conference on Fractal Geometry: Mathematical Methods, Algorithms and Applications (Eds. J M Blackledge, A K Evans and M Turner), Horwood Publishing Series in Mathematics and Applications, pp. 189-222, 2002

[6] Ptitsyn, N. V., *Deterministic Chaos if Digital Cryptography*, PhD Thesis, De Montfort University, 2003.

[7] Webster, A. G., *Partial Differential Equations of Mathematical Physics*, Stechert, 1933.

[8] Morse, P. M. and Feshbach, H., *Methods of Theoretical Physics*, McGraw-Hill, 1953.

[9] Butkov, E., *Mathematical Physics*, Addison-Wesley, 1973.

[10] Evans, G. A., Blackledge, J. M. and Yardley, P., *Analytical Solutions to Partial Differential Equations*, Springer, 1999.

[11] Roach, G. F., *Green's Functions (Introductory Theory with Applications)*, Van Nostrand Reihold, 1970.

[12] Stakgold, I., *Green's Functions and Boundary Value Problems*, Wiley, 1979.

[13] Dirac, P. A. M., *The Principles of Quantum Mechanics*, Oxford University Press, 1947.

[14] Hoskins, R. F., *The Delta Function*, Horwood Publishing, 1999.

[15] Hoskins, R. G. and Sousa Pinto, J., *Theories of Generalised Functions: Distributions, Ultradistributions and Other Generalised Functions*, Horwood, 2005.

[16] Watson, E. J., *Laplace Transforms and Applications*, Van Nostrand Reinhold, 1981.

[17] Papoulis, A., *The Fourier Integral and its Applications*, McGraw-Hill, 1962.

[18] Bracewell, R. N., *The Fourier Transform and its Applications*, McGraw-Hill, 1978.

[19] Ferrers, N. M. (Ed.), *Mathematical Papers of George Green*, Chelsea, 1970.

[20] Wadsworth, G. P. and Bryan, J. G., *Introduction to Probability and Random Variables*, McGraw-Hill, 1960.

[21] Van der Waerden, B. L., *Mathematical Statistics*, Springer-Verlag, 1969.

[22] Wilks, S. S., *Mathematical Statistics*, Wiley, 1962.

[23] Laha, R. G. and Lukacs, E., *Applications of Characteristic Functions*, Griffin, 1964.

[24] Wackerly, D., Scheaffer, R. L. and Mendenhall, W., *Mathematical Statistics with Applications (6th Edition)*, Duxbury, May 2001.

[25] Steward, E. G., *Fourier Optics: An Introduction*, Horwood Scientific Publishing, 1987.

[26] Hecht, E., *Optics*, Addison-Wesley, 1987.

[27] Mandelbrot, B. B., *The Fractal Geometry of Nature*, Freeman, 1983.

[28] Barnsley, M. F., Dalvaney, R. L., Mandelbrot, B. B., Peitgen, H. O., Saupe, D. and Mandelbrot, R. F., *The Science of Fractal Images*, Springer, 1988.

[29] Turner, M. J., Blackledge, J. M. and Andrews, P. R., *Fractal Geometry in Digital Imaging*, Academic Press, 1997.

[30] Shannon, C. E., *A Mathematical Theory of Communication*, Bell System Technical Journal, Vol. 27, pp. 379-423, 1948.

[31] Sethna, J., *Statistical Mechanics : Entropy, Order Parameters and Complexity*, Oxford University Press, 2006.

[32] Buck, B. B. and Macaulay, V. A. (Eds.), *Maximum Entropy in Action*, Clarendon Press, 1992.

[33] Seth, A., Bandyopadhyay, S. and Maulik, U., *Probabilistic Analysis of Cellular Automata Rules and its Application in Pseudo Random Pattern Generation*, IAENG International Journal of Applied Mathematics, Vol. 38, No. 4, pp. 1-9, 2008. http://www.iaeng.org/IJAM/issues_v38/issue_4/IJAM_38_4_07.pdf

[34] Awad, A. and Saadane, A., *New Chaotic Permutation Methods for Image Encryption*, IAENG International Journal of Computer Science, Vol. 37, No. 4, pp. 1-9, 2010. http://www.iaeng.org/IJCS/issues_v37/issue_4/IJCS_37_4_10.pdf

[35] http://www.freedownloadscenter.com/Best/des3-source.html

[36] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[37] Blackledge, J M and Mahmoud, K. W., "Printed Document Authentication using Texture Coding", ISAST Journal on Electronics and Signal Processing, Vol. 4, No 1, 81-98, 2009

[38] http://en.wikipedia.org/wiki/Electronic_Data_Interchange

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

[40] http://www.opsi.gov.uk/acts/acts2000/ukpga_20000023_en_1