2007-01-01

Digital Watermarking and Self-Authentication Using Chirp Coding

Jonathan Blackledge
Dublin Institute of Technology, jonathan.blackledge59@gmail.com

Digital Watermarking and Self-Authentication using Chirp Coding

Jonathan M Blackledge, Fellow, IET

***Abstract*— This paper discusses a new approach to 'watermarking' digital signals using linear frequency modulated or 'chirp' coding. The principles underlying this approach are based on the use of a matched filter to provide a reconstruction of a chirped code that is uniquely robust, i.e. in the case of very low signal-to-noise ratios.**

Chirp coding for authenticating data is generic in the sense that it can be used for a range of data types and applications (the authentication of speech and audio signals, for example). The theoretical and computational aspects of the matched filter and the properties of a chirp are revisited to provide the essential background to the method. Signal code generating schemes are then addressed and details of the coding and decoding techniques considered.

***Index Terms*— Digital Watermarking, Chirp Coding, Data Authentication, Self-Authentication**

I. INTRODUCTION

DIGITAL watermarking has been researched for many years in order to achieve methods which provide both anti-counterfeiting and authentication facilities [1]. One of equations that underpins this technology is based on the model a the signal given by (e.g. [2], [3] and [4])

$$s = \hat{P}f + n \tag{1}$$

where f is the information content for the signal, $\hat{P}$ is a linear operator, n is noise and s is the output signal. This equation is usually taken to describe a stationary process which includes the characterisation of n (i.e. the probability density function of n is assumed to be invariant of time).

In the field of cryptography, the operation $\hat{P}f$ is referred to as the processes of 'diffusion' and the process of adding noise (i.e. $\hat{P}f + n$) is referred to as the process of 'confusion'. The principal 'art' is to develop methods in which the processes of diffusion and confusion are maximized; one important criterion being that the output s should be dominated by the noise n which in turn should be characterized by maximum Entropy (i.e. a uniform statistical distribution) [6].

Instead of n being taken to be noise, suppose that n is a known signal and that $\|n\| >> \|\hat{P}f\|$. In this case it may be possible to embed or 'hide' the information contained in f in the signal n without significantly perturbing it. The process of hiding secret information in signals or images is known as Steganography [5] and being able to recover f from s in equation (1) can provide a way of authenticating the signal n. If, in addition, it is possible to determine that a copy of s has been made leading to some form of data degradation and/or corruption that can be conveyed through an appropriate analysis of f, then a scheme can be developed that provides a check on: (i) the authenticity of the data n; (ii) its fidelity [7], [8]. In this case, signal f is an example of a watermark.

Formally, the recovery of f from s is based on the inverse process

$$f = \hat{P}^{-1}(s - n)$$

where $\hat{P}^{-1}$ is the inverse operator. Clearly, this requires the signal n to be known *a priori* and that the inverse process $\hat{P}^{-1}$ is well defined and computationally stable. Since the host signal n must be known in order to recover the watermark f, this approach leads to a private watermarking scheme in which the field n represents a key. In addition, the operator $\hat{P}$ (and its inverse $\hat{P}^{-1}$) can be key dependent. The value of this operator key dependency relies on the nature and properties of the operator that is used and whether it is compounded in an algorithm that is required to be in the public domain, for example.

Another approach is to consider the case in which the signal n is unknown and to consider the problem of extracting the watermark f in the absence of knowledge of this signal. In this case, the reconstruction is based on the result

$$f = \hat{P}^{-1}s + m$$

where

$$m = -\hat{P}^{-1}n.$$

If a process $\hat{P}$ is available in which $\|\hat{P}^{-1}s\| >> \|m\|$, then an approximate reconstruction of f may be obtained in which m is determined by the original signal-to-noise ratio of the data s and hence, the level of covertness of the information $\hat{P}f$ - diffused watermark. In this case, it may be possible to post-process the reconstruction and recover a relatively high-fidelity version of the watermark, i.e.

$$f \sim \hat{P}^{-1}s.$$

This approach (if available) does not rely on a private key (assuming $\hat{P}$ is not key dependent). The ability to recover the watermark only requires knowledge of the operator $\hat{P}$ (and its inverse) and post-processing options as required. The problem is to find an operator that is able to diffuse and recover the watermark f effectively in the presence of the signal n when $\|\hat{P}f\| << \|n\|$, i.e. with very low signal-to-noise ratios.

Manuscript received June 1, 2007.

Jonathan Blackledge is Professor of Information and Communications Technology, Applied Signal Processing Research Group, Department of Electronic and Electrical Engineering, Loughborough University, England and Professor of Computer Science, Department of Computer Science, University of the Western Cape, Cape Town, Republic of South Africa (e-mail: jon.blackledge@btconnect.com).

Ideally, we require an operator $\hat{P}$ with properties such that $\hat{P}^{-1}n \rightarrow 0$.

In this paper, we consider the case where the operator $\hat{P}$ is based on a chirp function, specifically, a linear Frequency Modulated (FM) chirp of the (complex) type $\exp(i\alpha t^2)$ where α is the chirp parameter and t is the independent variable[1]. This function is then convolved with f. The inverse process is undertaken by correlating with the (complex) conjugate of the chirp $\exp(-i\alpha t^2)$. This provides a reconstruction for f that is accurate and robust. Further, we consider a watermark based on a coding scheme in which the signal n is the input. The watermark f is therefore n-dependent. This allows an authentication scheme to be developed in which the watermark is generated from the signal in which it is to be 'hidden'. Authentication of the watermarked data is then based on comparing the code generated from $s \sim n$ and that reconstructed from $s = \hat{P}f + n$ when $\|\hat{P}f\| << \|n\|$. This is an example of a self-generated coding scheme which avoids the use, distribution and application of reference data. In this paper, the coding scheme is based on the application of Daubechies wavelets.

There are numerous applications of this technique in areas such as telecommunications and speech recognition where authentication is often mandatory. For example, as demonstrated in this paper, the method can be applied to audio data with no detectable differences in the audio quality of the data.

II. The Matched Filter and Linear FM 'Chirp' Functions

The Matched Filter (e.g. [9], [10] and [11]) is one of the most common filters used for pattern recognition. It is based on correlating a signal/image with a matching template of the feature that is assumed to be present in the signal/image [4]. If the feature does indeed exist, then the output of the filter (the correlation signal/surface) produces a local maximum or spike where the feature occurs. This process can be applied generally, but when the template and feature are based on chirp functions, the result has some special and important properties which provide an output that is uniquely robust in the case when the signal-to-noise ratio is very low. It is this property that forms the basis for a variety of active imaging systems such as those used in Real and Synthetic Aperture Radar (e.g. [12], [13] and [14]), active sonar and some forms of seismic prospecting, for example. Interestingly, some mammals (including dolphins, whales and bats) use frequency modulation for communication and (target) detection. The reason for this is the unique properties that chirps provide in terms of the quality of extracting information from signals with very low signal-to-noise ratios and the simplicity of the process that is required to do this (i.e. correlation). The invention and use of chirps for man made information and communications recovery dates back to the early 1960s (the application of FM to radar, for example); 'mother nature' appears to have 'discovered' the idea some time ago.

[1] In practice this is undertaken using the real or imaginary part of the complex chirp function.

A. The Matched Filter

We start by considering the basic linear stationary (convolution) model for a signal s as a function of time t, namely

$$s(t) = p(t) \otimes f(t) + n(t)$$

where p is the Impulse Response Function (IRF), f is the object function (the information content of some input signal), n is the noise (which is typically taken to have stationary statistics) and $\otimes$ is the convolution operation, i.e.

$$p(t) \otimes f(t) = \int p(t-\tau) f(\tau) d\tau.$$

A fundamental inverse (deconvolution) problem is to find an estimate $\hat{f}$ of f given s. The Matched Filter is based on assuming a linear convolution model for this estimate of the form

$$\hat{f}(t) = q(t) \otimes s(t).$$

Clearly, the problem is to find the filter q. The Matched Filter is based on finding q subject to the condition that

$$r = \frac{\mid \int Q(\omega) P(\omega) d\omega \mid^2}{\int \mid N(\omega) \mid^2 \mid Q(\omega) \mid^2 \, d\omega} \tag{2}$$

is a maximum where Q, P and N are the Fourier transforms of q, p and n respectively and where we defined the Fourier transform pair as

$$F(\omega) = \int f(t) \exp(-i\omega t) dt,$$

$$f(t) = \frac{1}{2\pi} \int F(\omega) \exp(i\omega t) d\omega$$

in which the limits of the integrals are taken to be in $(-\infty, \infty)$ and ω is the (angular) frequency. Note that the ratio defining r is a 'measure' of the signal-to-noise ratio. In this sense, the matched filter maximizes the Signal-to-Noise Ratio (SNR) of the output.

Assuming that the noise n has a 'white' or uniform power spectrum, the filter Q which maximizes the SNR defined by r can be shown to be given by the simple result (see Appendix I)

$$Q(\omega) = P^*(\omega).$$

The required solution is therefore given by

$$\hat{f}(t) = \frac{1}{2\pi} \int P^*(\omega) S(\omega) \exp(i\omega t) d\omega.$$

Using the 'correlation theorem' we can write

$$\hat{f}(t) = p(t) \odot s(t) \equiv \int p(\tau + t) s(\tau) d\tau.$$

Hence, the matched filter is based on correlating the signal s with the instrument function p.

B. Deconvolution of Linear Frequency Modulated Chirps

The matched filter is frequently used in systems that utilize linear Frequency Modulated (FM) signals. Signals of this type are known as 'chirped signals'. A linear FM signal which is taken to be of compact support ($t \in [-T/2, T/2]$) is given (in complex form) by

$$p(t) = \exp(i\alpha t^2), \quad | t | \leq \frac{T}{2}$$

where α is a constant (this defines the 'chirp rate') and T is the length of the signal. The phase of this signal is given by αt^2 (i.e. it has a quadratic phase factor) and its instantaneous frequency is therefore given by

$$\frac{d}{dt}(\alpha t^2) = 2\alpha t$$

which varies linearly with time t. Hence, the frequency modulations are linear which is why the signal is referred to as a 'linear' FM pulse.

For the purpose of clarity, let us first consider the case when the additive noise term is neglected and consider a signal given by

$$s(t) = \exp(i\alpha t^2) \otimes f(x), \quad | t | \leq \frac{T}{2}.$$

If we now apply a (white noise) matched filter, then we have

$$\hat{f}(t) = \exp(-i\alpha t^2) \odot \exp(i\alpha t^2) \otimes f(t), \quad | t | \leq \frac{T}{2}.$$

The correlation integral can now be evaluated thus

$$\exp(-i\alpha t^2) \odot \exp(i\alpha t^2) = \int_{-T/2}^{T/2} \exp[-i\alpha(\tau + t)^2] \exp(i\alpha\tau^2) d\tau$$

$$= \exp(-i\alpha t^2) \int_{-T/2}^{T/2} \exp(-2i\alpha t\tau) d\tau$$

Evaluating the integral over τ, we have

$$\exp(-i\alpha t^2) \odot \exp(i\alpha t^2) = T \exp(-i\alpha t^2) \text{sinc}(\alpha T t)$$

and hence

$$\hat{f}(t) = T \exp(-i\alpha t^2) \text{sinc}(\alpha T t) \otimes f(t).$$

A further useful simplification can now be made to the result for $\hat{f}$ which allows the exponential term to be ignored. In particular, if we consider $T >> 1$ then

$$\cos(\alpha t^2) \text{sinc}(\alpha T t) \simeq \text{sinc}(\alpha T t)$$

and

$$\sin(\alpha t^2) \text{sinc}(\alpha T t) \simeq 0$$

so that

$$\hat{f}(t) \simeq T \text{sinc}(\alpha T t) \otimes f(t)$$

This simplification, under a condition that is usually practically applicable, allows the result for $\hat{f}$ to be easily analysed in Fourier space. Using the convolution theorem we can write (ignoring scaling by π/α)

$$\hat{F}(\omega) = \begin{cases} F(\omega), & | \omega | \leq \alpha T; \\ 0, & | \omega | > \alpha T. \end{cases}$$

which describes $\hat{f}$ as being a band-limited version of f (assuming the f is not band-limited) where the bandwidth is determined by αT.

In the presence of additive noise, the result is

$$\hat{f}(t) \simeq T \text{sinc}(\alpha T t) \otimes f(t) + \exp(-i\alpha t^2) \odot n(t).$$

The correlation function produced by the correlation of $\exp(-i\alpha t)$ with $n(t)$ will in general be relatively low in amplitude since $n(t)$ will not normally have features that match those of a (complex) chirp. Thus, it is reasonable to assume that

$$\| T \text{sinc}(\alpha T t) \otimes f(t) \| >> \| \exp(-i\alpha t^2) \odot n(t) \|$$

and that in practice, $\hat{f}$ is a band-limited reconstruction of f with high SNR. Thus, the process of using chirp signals with matched filtering for the purpose of reconstruction in the presence of additive noise provides a relatively simple and computationally reliable method of 'diffusing' and reconstructing information encoded in the function f. This is the underlying principle behind the method of watermarking described in this paper.

An example of a matched filter reconstruction is given in Figure 1. Here, two spikes have been convolved with a linear FM chirp of the form $p(t) = \sin(\alpha t^2)$ whose width or pulse length T is significantly greater than that of the input signal. The output signal

$$s(t) = p(t) \otimes f(t) + n(t)$$

has been generated using an SNR of 1, the SNR being defined by

$$SNR = \frac{\| p(t) \otimes f(t) \|_\infty}{\| n(t) \|_\infty}$$

where $\| \bullet \|_\infty$ is the uniform norm. Clearly, this example illustrates the quality of the restoration of the input $f(t)$ using a relatively simple operation for processing data that has been badly distorted by additive noise.

III. Chirp Coding, Decoding and Watermarking

We now consider the an approach to watermarking signals using chirp functions. The basic model for the watermarked signal (which is real) is

$$s(t) = \text{chirp}(t) \otimes f(t) + n(t)$$

where

$$\text{chirp}(t) = \sin(\alpha t^2)$$

We consider the field $n(t)$ to be some pre-defined signal to which a watermark is to be 'added' to generate $s(t)$. In principle, any watermark described by the function $f(t)$ can be used. On the other hand, for the purpose of authentication we require two criterion: (i) $f(t)$ should represent a code which can be reconstructed accurately and robustly; (ii)

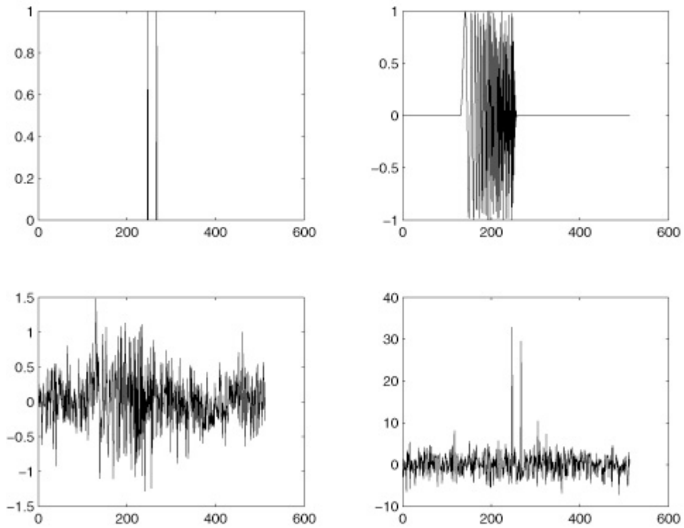

Fig. 1. Example of a reconstruction using the matched filter (bottom right) from a noisy signal based on the convolution of an input consisting of two spikes f (top left) with a linear FM chirp p (top right) and the addition of noise n with a SNR of 1 to give the signal s (bottom left).

the watermark code should be sensitive (and ideally ultra-sensitive) to any degradation in the field $n(t)$ due to lossy compression and/or copying. To satisfy condition (i), it is reasonable to consider $f(t)$ to represent a bit stream, i.e. to consider the discretized version of $f(t)$ - the vector f_i - to be composed of a set of elements with value 0 or 1. This binary code can of course be based on a key or set of keys which, when reconstructed, is compared to the key(s) for the purpose of authenticating the data. However, this requires the distribution of such keys. Instead, we consider the case where a binary sequence is generated from the signal $n(t)$. There are a number of approaches that can be considered based on the spectral characteristics of $n(t)$, for example. These are discussed in Section IV in which binary sequences are produced from the application power spectrum segmentation and wavelet decomposition.

A. Chirp Coding

Given that a binary sequence has been generated from $n(t)$, we now consider the method of chirp coding. The purpose of chirp coding is to 'diffuse' each bit over a range of compact support. However, it is necessary to differentiate between 0 and 1 in the sequences. The simplest way to achieve this is to change the polarity of the chirp. Thus, for 1 we apply the chirp $\sin(\alpha t^2), \quad t \in T$ and for 0 we apply the chirp -$\sin(\alpha t^2), \quad t \in T$ where T is the chirp period. The chirps are then concatenated to produce a contiguous stream of data, i.e. a signal composed of ±chirps. Thus, the binary sequence 010, for example, is transformed to the signal

$$s(t) = \begin{cases} -\text{chirp}(t), & t \in [0, T); \\ +\text{chirp}(t), & t \in [T, 2T); \\ -\text{chirp}(t), & t \in [2T, 3T). \end{cases}$$

The period over which the chirp is applied depends on the length of the signal to which the watermark is to be applied and the length of the binary sequence. In the example given above the length of the signal is taken to be $3T$. In practice, care must be taken over the chirping parameter α that is applied given a period T in order to avoid aliasing and in some cases it is of value to apply a logarithmic frequency sweep instead of a linear sweep as used in the MATLAB code given in Appendix II.

B. Decoding

Decoding or reconstruction of the binary sequence requires the application of a correlator using the function $\text{chirp}(t), \quad t \in [0, T)$. This produces a correlation function that is either -1 or +1 depending upon whether $-\text{chirp}(t)$ or $+\text{chirp}(t)$ has been applied respectively. For example, after correlating the chirp coded sequence 010 given above, the correlation function $c(t)$becomes

$$c(t) = \begin{cases} -1, & t \in [0, T); \\ +1, & t \in [T, 2T); \\ -1, & t \in [2T, 3T). \end{cases}$$

from which the original sequence 010 is easily inferred - the change in sign of the correlation function identifying a change of bit (from 0 to 1 or from 1 to 0). Note that in practice the correlation function may not be exactly 1 or -1 when reconstruction is undertaken in the presence of additive noise; the binary sequence is effectively recovered by searching the correlation function for changes in sign.

C. Watermarking

The watermarking process is based on adding the chirp coded data to the signal $n(t)$. Let the chirp coded signal be given by the function $h(t)$, then the watermarking process is described by the equation

$$s(t) = a\left[\frac{bh(t)}{\|h(t)\|_\infty} + \frac{n(t)}{\|n(t)\|_\infty}\right]$$

The coefficients $a > 0$ and $0 < b < 1$ determine the amplitude and the SNR of s respectively where

$$a = \|n(t)\|_\infty.$$

The coefficient a is required to provide a watermarked signal whose amplitude is compatible with the original signal n. The value of b is adjusted to provide an output that is acceptable in the application to be considered and to provide a robust reconstruction of the binary sequence by correlating $s(t)$ with $\text{chirp}(t), t \in [0, T)$.

IV. Code Generation

In the previous section, the method of chirp coding a binary sequence and watermarking the signal $n(t)$ has been discussed where it is assumed that the sequence is generated from this same signal. In this section, the details of this method are presented. There are a wide variety of coding methods that can be applied [15]. The problem is to convert the salient characteristics of the signal $n(t)$ into a sequence of bits that is relatively short and conveys information on the signal that is unique to its overall properties. In principle, there are a number of ways of undertaking this. For example, in practice the digital signal n_i - which is composed of an array of floating point numbers - could be expressed in binary form and each element concatenated to form a contiguous bit stream. However, the length of the code (i.e. the total number of bits in the stream) will tend to be large leading to high computational costs in terms of the application of chirp coding/decoding. What is required, is a process that yields a relatively short binary sequence (when compared with the original signal) that reflects the important properties of the signal in its entirety. Two approaches are considered here: (i) Power Spectral Density decomposition and (ii) Wavelet decomposition [16].

A. Power Spectral Density Decomposition

Let $N(\omega)$ be the Fourier transform $n(t)$ and define the Power Spectrum $P(\omega)$ as

$$P(\omega) = \mid N(\omega) \mid^2$$

An important property of the binary sequence is that it should describe the spectral characteristics of the signal in its entirety. Thus, if, for example, the binary sequence is based on just the low frequency components of the signal, then any distortion of the high frequencies of the watermarked signal will not affect the recovered watermark and the signal will be authenticated. Hence, we consider the case where the power spectrum is segmented into N components, i.e.

$$P_1(\omega) = P(\omega), \quad \omega \in [0, \Omega_1)$$
$$P_2(\omega) = P(\omega), \quad \omega \in [\Omega_1, \Omega_2)$$
$$\vdots$$
$$P_N(\omega) = P(\omega), \quad \omega \in [\Omega_{N-1}, \Omega_N)$$

Note that it is assumed that the signal $n(t)$ is band-limited with a bandwidth of Ω_N.

The set of the functions $P_1, P_2, ..., P_N$ now represent the complete spectral characteristics of the signal $n(t)$. Since each of these functions represents a unique part of the spectrum, we can consider a single measure as an identifier or tag. A natural measure to consider is the energy which is given by the integral of the functions over their frequency range. In particular, we consider the energy values in terms of their contribution to the spectrum as a percentage, i.e.

$$E_1 = \frac{100}{E} \int\limits_0^{\Omega_1} P_1(\omega) d\omega$$

$$E_2 = \frac{100}{E} \int\limits_{\Omega_1}^{\Omega_2} P_2(\omega) d\omega$$

$$\vdots$$

$$E_N = \frac{100}{E} \int\limits_{\Omega_{N-1}}^{\Omega_N} P_N(\omega) d\omega$$

where

$$E = \int\limits_0^{\Omega_N} P(\omega) d\omega.$$

Code generation is then based on the following steps:

1) Rounding to the nearest integer the (floating point) values of E_i to decimal integer form:
$$e_i = \text{round}(E_i), \quad \forall i$$
2) Decimal integer to binary string conversion:
$$b_i = \text{binary}(e_i)$$
3) Concatenation of the binary string array b_i to a binary sequence:
$$f_j = \text{cat}(b_i)$$

The watermark f_j is then chirp coded as discussed in Section V.

B. Wavelet decomposition

Wavelet signal analysis is based on convolution type operations which include a scaling property in terms of the amplitude and temporal extent of the convolution kernel (e.g. [3], [17], [18] and [19]). There is a close synergy between the wavelet transform and imaging science. For example, in Fresnel optics, the two-dimensional (coherent) optical wavefield u generated by an object function f (in the object plane at a distance z) is given by (e.g. [4] and [20])

$$u(x, y, L) = p(x, y, L) \otimes \otimes f(x, y)$$

where

$$p(x, y, L) = i \exp\left(i \frac{2\pi z}{\lambda}\right) \frac{1}{L} \exp\left[\frac{i\pi}{L}(x^2 + y^2)\right]$$

and $L = \lambda z$ for wavelength λ. An important feature of this result is that the amplitude of the kernel p and its scale length is determined by the reciprocal of the wavelength λ. Physically, this implies that as the wavelength decreases, the 'resolving power' of an image given by $I(x, y, L) = \mid u(x, y, L) \mid^2$ increases, the bandwidth u being proportional to λ^{-1}. Thus, by considering a hypothetical Fresnel imaging system, in which the wavelength can be varied by the user, we can consider the imaging system to have multi-resolution properties. The Frensel transform is essentially a wavelet transform with a wavelet determined by a (two-dimensional) chirp function.

The multi-resolution properties of the wavelet transform have been crucial to their development and success in the analysis and processing of signals. Wavelet transformations play a central role in the study of self-similar or fractal signals. The transform constitutes as natural a tool for the manipulation of self-similar or scale invariant signals as the Fourier transform does for translation invariant signals such as stationary and periodic signals.

In general, the wavelet transformation of a signal $f(t)$ say

$$f(t) \leftrightarrow F_L(t)$$

is defined in terms of projections of $f(t)$ onto a family of functions that are all normalized dilations and translations of a prototype 'wavelet' function W, i.e.

$$\hat{W}[f(t)] = F_L(\tau) = \int f(t) w_L(t,\tau) dt$$

where

$$w_L(t,\tau) = \frac{1}{\sqrt{\mid L \mid}} w\left(\frac{\tau - t}{L}\right).$$

The parameters L and τ are continuous dilation and translation parameters respectively, and take on values in the range $-\infty < L, \tau < \infty, L \neq 0$. Note that the wavelet transformation is essentially a convolution transform in which $w(t)$ is the convolution kernel but with a factor L introduced. The introduction of this factor provides dilation and translation properties into the convolution integral that gives it the ability to analyse signals in a multi-resolution role (the convolution integral is now a function of L). A multi-resolution signal analysis is a framework for analysing signals based on isolating variations that occur on different temporal or spatial scales. The basic analysis involves approximating the signal at successively coarser scales through repeated application of a smoothing (convolution) operator.

A necessary and sufficient condition for a wavelet transformation to be invertible is that $w(t)$ satisfy the *admissibility condition*

$$\int \mid W(\omega) \mid^2 \mid \omega \mid^{-1} d\omega = C_w < \infty$$

where W is the wavelets Fourier transform, i.e.

$$W(\omega) = \int w_L(t) \exp(-i\omega t) dt.$$

For any admissible $w(t)$, the wavelet transform has an inverse given by [3]

$$f(t) = \hat{W}^{-1}[F_L(\tau)] = \frac{1}{C_w} \int \int F_L(\tau) w_L(t,\tau) L^{-2} dL d\tau.$$

There are a wide variety of wavelets available [i.e. functional forms for $w_L(t)$] which are useful for processing digital signals in 'wavelet space' when applied in discrete form. The properties of the wavelets vary from one application to another but in each case, the digital signal f_i is decomposed into a matrix (a set of vectors) F_{ij} where j is the 'level' of the decomposition.

The wavelet transform can be used to generate a suitable code by computing the energies of the wavelet transformation over N levels. Thus, the signal $f(t)$ is decomposed into wavelet space to yield the following set of functions:

$$F_{L_1}(\tau), \quad F_{L_2}(\tau), \quad ... \quad F_{L_N}(\tau)$$

The (percentage) energies of these functions are then computed, i.e.

$$E_1 = \frac{100}{E} \int \mid F_{L_1}(\tau) \mid^2 d\tau$$

$$E_2 = \frac{100}{E} \int \mid F_{L_1}(\tau) \mid^2 d\tau$$

$$\vdots$$

$$E_N = \frac{100}{E} \int \mid F_{L_N}(\tau) \mid^2 d\tau$$

where

$$E = \sum_{i=1}^{N} E_i$$

The method of computing the binary sequence for chirp coding from these energy values follows that described in the method of power spectral segmentation given in previous Section.

V. MATLAB Application Programs

Two MATLAB programs have been developed to implement the watermarking method discussed in this paper. The *coding* program reads in a named file, applies the watermark to the data using wavelet decomposition and writes out a new file using the same file format. The *Decoding* program reads a named file (assumed to contain the watermark or otherwise), recovers the code from the watermarked data and then recovers the (same or otherwise) code from the watermark. The coding program displays the decimal integer and binary codes for analysis. The decoding program displays the decimal integer streams generated by the wavelet analysis of the input signal and the stream obtained by processing the signal to extract the watermark code or otherwise. This program also provides an error measure based on the result

$$e = \frac{\sum\limits_i \mid x_i - y_i \mid}{\sum\limits_i \mid x_i + y_i \mid}$$

where x_i and y_i are the decimal integer arrays obtained from the input signal and the watermark (or otherwise). In the application considered here, the watermarking method has been applied to audio (.wav) files in order to test the method on data which requires that the watermark does not affect the fidelity of the output (i.e. audio quality). Only a specified segment of the data is extracted for watermarking. The segment can be user defined and if required, form the basis for a (private) key system. In this application, the watermarked segment has been 'hard-wired' and represents a public key.

A. Coding process

The coding process is compounded in the following basic steps:

1) Read a .wav file.
2) Extract a section of a single vector of the data (note that a .wav contains stereo data, i.e. two vectors arrays).
3) Apply wavelet decomposition using Daubechies wavelets with 7 levels. Note that in addition to wavelet decomposition, the approximation coefficients for the input signal are computed to provide a measure on the global effect of introducing the watermark into the signal. Thus, 8 decomposition vectors in total are generated.
4) Compute the (percentage) 'energy values'.
5) Round to the nearest integer and convert to binary form.
6) Concatenate both the decimal and binary integer arrays.
7) Chirp code the binary sequence.

8) Scale the output and add to the original input signal.
9) Re-scale the watermarked signal.
10) Write to a file.

B. Decoding process

The decoding process is as follows:

1) Steps 1-6 in the coding processes are repeated
2) Correlate the data with a chirp identical to that used for chirp coding
3) Extract the binary sequence
4) Convert from binary to decimal
5) Display the original and reconstructed decimal sequence
6) Display the error

Note that in a practical application of this method for authenticating audio files, for example, a threshold can be applied to the error value. If and only if the error lies below this threshold is the data taken to be authentic.

The prototype MATLAB programs for implementing this scheme are given in Appendix II (Coding) and Appendix III (Decoding). They have been developed to explore the applications of the method for different audio (.wav) signals but can be tailored for different signals and file formats. Note that in the decoding program, the correlation process is carried out using a spatial cross-correlation scheme (using the MATLAB function *xcorr*), i.e. the watermark is recovered using the process $\mathrm{chirp}(t) \odot s(t)$ instead of the Fourier equivalent $\mathrm{CHIRP}^*(\omega)S(\omega)$ where CHIRP and S are the Fourier transforms of chirp and s respectively. This is due to the fact that the 'length' of the chirp function is significantly less than that of the signal. Application of a spatial correlator therefore provides greater computational efficiency.

VI. Discussion

The method of digital watermarking discussed here makes specific use of the chirp function. This function is unique in terms of its properties for reconstructing information (via application of the Matched Filter). The watermark f extracted from the host signal n is, in theory, an exact band-limited version of the original watermark.

The approach considered in this paper allows a code to be generated directly from the host signal and that same code used to watermark the signal. The code is therefore self-generating and its reconstruction only requires a correlation process with the watermarked signal to be undertaken. This means that the signal can be authenticated without reference to a known data base. In other words, the method can be seen as a way of authenticating data by extracting a code (the watermark) within a 'code' (the host signal) and is consistent with approaches that attempt to reconstruct information without knowledge of the host data [21].

Audio data watermarking schemes rely on the imperfections of the human audio system. They exploit the fact that the human auditory system is insensitive to small amplitude changes, either in the time or frequency domains, as well as insertion of low amplitude time domain echo's. Spread spectrum techniques augment a low amplitude spreading sequence, which can be detected via correlation techniques. Usually, embedding is performed in high amplitude portions of the signal, either in the time or frequency domains. A common pitfall for both types of watermarking systems is their intolerance to detector de-synchronization and deficiency of adequate methods to address this problem during the decoding process. Although other applications are possible, chirp coding provides a new and novel technique for fragile audio watermarking. In this case, the watermarked signal does not change the perceptual quality of the signal. In order to make the watermark inaudible, the chirp generated is of very low frequency and amplitude. Using audio files with sampling frequencies of over 1000Hz, a logarithmic chirp can be generated in the frequency band of 1-100Hz. Since the human ear has low sensitivity in this band, the embedded watermark will not be perceptible. Depending upon the band and amplitude of the chirp, the signal-to-watermark (chirp stream) ratio can be in excess of 40dB.

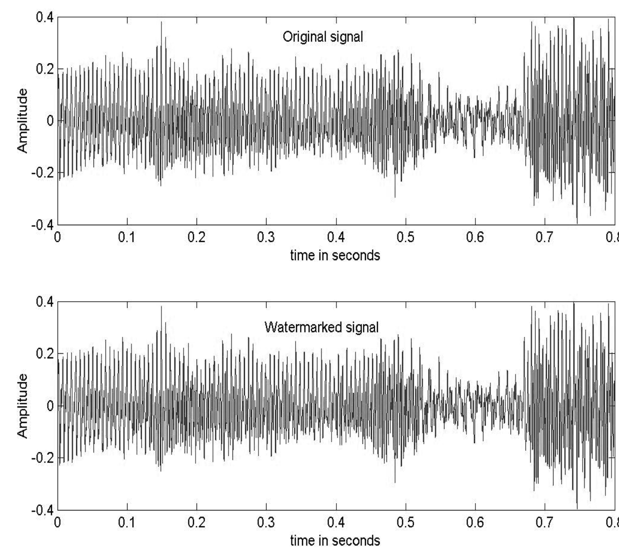

Fig. 2. Original signal (above) and chirp based watermarked signal (below).

Figure 2 is an example of an original and a watermarked audio signal which shows no perceptual difference during a listening test. Various forms of attack can be applied which change the distribution of the percentage sub-band energies originally present in the signal including filtering (both low pass and high pass), cropping and lossy compression (MP3 compression) with both constant and variable bit rates. In each case, the signal and/or the watermark is distorted enough to register the fact that the data has been tampered with. An example of this is given in Figure 3 which shows the power spectral density of an original, watermarked and a (band-pass filtered) tampered audio signal. The filtering is such that there is negligible change in the power spectral density. However, the tampering was easily detected by the proposed technique. Finally, chirp coded watermarks are difficult to remove from the host signal since the initial and the final frequency is at the discretion of the user and its position in the data stream can be varied through application of an offset, all such parameters being combined to form a private key.

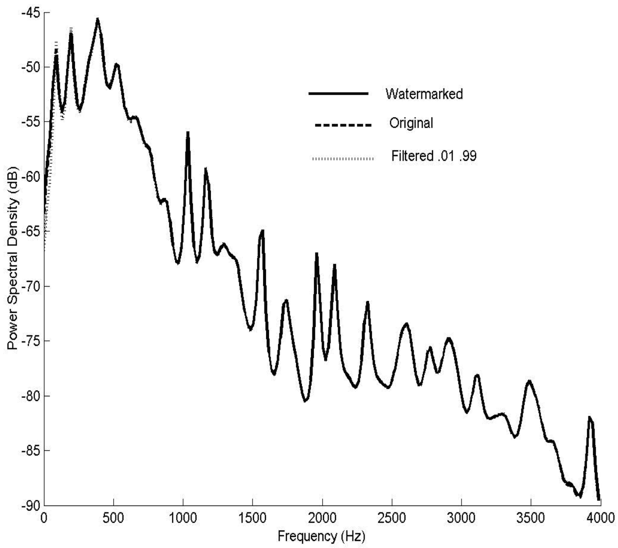

Fig. 3. Difference in the power spectral density of the original, watermarked and tampered signal. The tampering has been undertaken using a band pass filter with a normalised lower cut-off frequency of 0.01 and higher cut-off frequency 0.99.

Chirp coding is generic in the sense that it can be used to watermark any (user defined) bit stream in a signal. For watermarking with plaintexts, the bit stream can be generated using a standard ASCII (7-bit) code. Thus, the use of this method for self-authenticating signals, as discussed in this paper, is just one approach, albeit a useful one. However, in terms of sending and receiving data through some communications channel, the most important feature of chirp coding is the facility it provides for transmitting information through environments with significant amounts of noise, recovery of this information being based on knowledge of the exact chirp function used to 'chirp code'.

The radio frequency spectrum of the universe is relatively quiet when compared to other parts of the electromagnetic spectrum such as the microwave spectrum. Nevertheless, radio wave emissions will acquire a significant amount of noise if transmitted over distances of many light years. Chirp coding may provide a way of preserving such information when it is known that the final SNR is likely to be very small. In the search for extraterrestrial intelligence, the radio spectrum is considered to be the most likely frequency range in which 'intelligent signals' might exist. In light of the above, it may be of value to analyse such radio signals by correlating them with a range of different chirp functions, focusing on those outputs (i.e. the correlation functions) that provide some minimum Entropy measure.

APPENDIX I
DERIVATION OF THE MATCHED FILTER

Given equation (2), the matched filter is essentially a 'by-product' of the 'Schwarz inequality', i.e. the result

$$\left| \int Q(\omega)P(\omega)d\omega \right|^2 \leq \int \mid Q(\omega) \mid^2 d\omega \int \mid P(\omega) \mid^2 d\omega.$$

The principal trick is to write

$$Q(\omega)P(\omega) = \mid N(\omega) \mid Q(\omega) \times \frac{P(\omega)}{\mid N(\omega) \mid}$$

so that the above inequality becomes

$$\left| \int Q(\omega)P(\omega)d\omega \right|^2 = \left| \int \mid N(\omega) \mid Q(\omega) \frac{P(\omega)}{\mid N(\omega) \mid} d\omega \right|^2$$

$$\leq \int \mid N(\omega) \mid^2 \mid Q(\omega) \mid^2 d\omega \int \frac{\mid P(\omega) \mid^2}{\mid N(\omega) \mid^2} d\omega.$$

From this result, using the definition of r given in equation (2), we see that

$$r \leq \int \frac{\mid P(\omega) \mid^2}{\mid N(\omega) \mid^2} d\omega.$$

Now, if r is to be a maximum, then we require that

$$r = \int \frac{\mid P(\omega) \mid^2}{\mid N(\omega) \mid^2} d\omega$$

or

$$\left| \int \mid N(\omega) \mid Q(\omega) \frac{P(\omega)}{\mid N(\omega) \mid} d\omega \right|^2$$

$$= \int \mid N(\omega) \mid^2 \mid Q(\omega) \mid^2 d\omega \int \frac{\mid P(\omega) \mid^2}{\mid N(\omega) \mid^2} d\omega.$$

But this is only true if

$$\mid N(\omega) \mid Q(\omega) = \frac{P^*(\omega)}{\mid N(\omega) \mid}.$$

Hence, r is a maximum when

$$Q(\omega) = \frac{P^*(\omega)}{\mid N(\omega) \mid^2}.$$

Noise is usually characterised by: (i) the Probability Density Function (PDF) or the Characteristic Function (i.e. the Fourier transform of the PDF); (ii) the Power Spectral Density Function (PSDF). To apply the Matched Filter, the function $\mid N(\omega) \mid^2$ (i.e. the power spectrum of the noise), in addition to $P(\omega)$, is required to be known *a priori*. In some practical systems, this is possible if the Impulse Response Function is zero so that the output of the system is 'noise driven'. In general however, it is often necessary to develop a suitable model for the PSDF. Such models may include uniform, Gaussian, Poisson or random fractal noise, for example, which may be suitable in many cases [3]. However, if we consider the case when the PSDF is uniform or 'white' and of unit amplitude then we can write

$$\mid N(\omega) \mid^2 = 1 \forall \omega$$

so that the Matched Filter reduces to the simple result

$$Q(\omega) = P^*(\omega).$$

APPENDIX II
PROTOTYPE MATLAB CODING ALGORITHM

```
% Read a (.wav) audio file
  [au2,fs,nbit]=wavread('file');

% Clear the screen
  clc

% Compute the size of the data
  size(au2);

% Extract a single set of data composed of
% 1500150 (arbitrary) elements
  au1=au2(1:1500150,1);

% Set the watermarking scaling factor
% (user defined)
  div_fac=270;

% Extract data segment from 300031 to
% 1500150 (arbitrary) and compute the
% maximum value
  au=au1(300031:1500150,1);
  au_max1=max(au1(300031:1500150,1));

% Apply wavelet decomposition using
% Daubechies wavelets with 7 levels
  [ca cl]=wavedec(au(:,1),7,'db4');

% Compute the approximation
% coefficients at level 7
  appco=appcoef(ca,cl,'db4',7);

% Extract the 'detail coefficients'
% at each level
  detco7=detcoef(ca,cl,7);
  detco6=detcoef(ca,cl,6);
  detco5=detcoef(ca,cl,5);
  detco4=detcoef(ca,cl,4);
  detco3=detcoef(ca,cl,3);
  detco2=detcoef(ca,cl,2);
  detco1=detcoef(ca,cl,1);

% Compute the energy for each set
% of coefficients
  ene_appco=sum(appco.^2);
  ene_detco7=sum(detco7.^2);
  ene_detco6=sum(detco6.^2);
  ene_detco5=sum(detco5.^2);
  ene_detco4=sum(detco4.^2);
  ene_detco3=sum(detco3.^2);
  ene_detco2=sum(detco2.^2);
  ene_detco1=sum(detco1.^2);

% Compute the total enegy of all
% the coefficients
  tot_ene=round(ene_detco7...
  +ene_detco6+ene_detco5+...
  ene_detco4+ene_detco3+...
  ene_detco2+ene_detco1);

% Round towards the nearest
% integer the percentage energy of
% each set
  pene_hp7=round(ene_detco7*100/tot_ene);
  pene_hp6=round(ene_detco6*100/tot_ene);
  pene_hp5=round(ene_detco5*100/tot_ene);
  pene_hp4=round(ene_detco4*100/tot_ene);
  pene_hp3=round(ene_detco3*100/tot_ene);
  pene_hp2=round(ene_detco2*100/tot_ene);
  pene_hp1=round(ene_detco1*100/tot_ene);

% Do decimal integer to binary conversion
% with at least 17 bits
  tot_ene_bin=dec2bin(tot_ene,31);
  f7=dec2bin(pene_hp7,17);
  f6=dec2bin(pene_hp6,17);
  f5=dec2bin(pene_hp5,17);
  f4=dec2bin(pene_hp4,17);
  f3=dec2bin(pene_hp3,17);
  f2=dec2bin(pene_hp2,17);
  f1=dec2bin(pene_hp1,17);

% Concatenate the arrays f1,f2,...
% along dimension 2 to produce a binary
% sequence (watermark code)
  wmark=cat(2,tot_ene_bin,f7,f6,f5,f4,...
            f3,f2,f1);

% Concatenate decimal integer array
  per_ce=cat(2,tot_ene,pene_hp7,...
  pene_hp6,pene_hp5,pene_hp4,...
  pene_hp3,pene_hp2,pene_hp1);

% Write out decimal integer and binary
% codes for analysis
  d_string=per_ce
  b_string=wmark

% Assign -1 to 0 and +1 to 1
  for j=1:150
    if str2num(wmark(j))==0
       x(j)=-1;
    else
       x(j)=1;
    end
  end

% Initialise a compute chirp
% function using a log sweep
  t=0:1/44100:10000/44100;
  y=chirp(t,00,10000/44100,100,'log');

% Compute +chirp for 1 and -chirp for 0,
% scale by div_fac and concatentate.
```

```
  znew=0;
  for j=1:150
    z=x(j)*y/div_fac;
    znew=cat(2,znew,z);
  end

% Compute length of znew and
% watermark signal
  znew=znew(2:length(znew));
  wmark_sig=znew'+au1;

% Compute power of watermark and
% power of signal
  w_mark_pow=(sum(znew.^2));
  sig_pow=(sum(au1.^2));

% Rescale watermarked signal
  wmark_sig1...
  =wmark_sig*au_max1/max(wmark_sig);

% Concatenate and write to file
  wmark_sig...
  =cat(2,wmark_sig1,au2(1:1500150,2));
  wavwrite(wmark_sig,fs,nbit,'file');
```

Appendix III
Prototype MATLAB Decoding Algorithm

```
% Clear variables and functions
% from memory
  clear

% Read watermarked file and
% clear screen
  [au,fs,nbit]=wavread('file');
  clc

% Extract data
  au1=au(300031:1500150,1);

% Do wavelet decomposition
  [ca cl]=wavedec(au1,7,'db4');

% Extract wavelet coefficients
  appco=appcoef(ca,cl,'db4',7);
  detco7=detcoef(ca,cl,7);
  detco6=detcoef(ca,cl,6);
  detco5=detcoef(ca,cl,5);
  detco4=detcoef(ca,cl,4);
  detco3=detcoef(ca,cl,3);
  detco2=detcoef(ca,cl,2);
  detco1=detcoef(ca,cl,1);

% Compute energy of
% wavelet coefficients
  ene_appco=sum(appco.^2);
  ene_detco7=sum(detco7.^2);
  ene_detco6=sum(detco6.^2);
  ene_detco5=sum(detco5.^2);
  ene_detco4=sum(detco4.^2);
  ene_detco3=sum(detco3.^2);
  ene_detco2=sum(detco2.^2);
  ene_detco1=sum(detco1.^2);

% Compute total energy factor
  tot_ene=round(ene_detco7...
  +ene_detco6+ene_detco5...
  +ene_detco4+ene_detco3...
  +ene_detco2+ene_detco1);

% Express energy values as a
% percentage of thetotal energy
% and round to nearest integer
  pene_hp7=round(ene_detco7*100/tot_ene);
  pene_hp6=round(ene_detco6*100/tot_ene);
  pene_hp5=round(ene_detco5*100/tot_ene);
  pene_hp4=round(ene_detco4*100/tot_ene);
  pene_hp3=round(ene_detco3*100/tot_ene);
  pene_hp2=round(ene_detco2*100/tot_ene);
  pene_hp1=round(ene_detco1*100/tot_ene);

  per_ene=cat(2,tot_ene,pene_hp7,...
  pene_hp6,pene_hp5,pene_hp4,...
  pene_hp3,pene_hp2,pene_hp1);

% Output original decimal integer code
% obtained from signal via wavelet
% decomposition
  original_d_string=per_ene;
  original_d_string
  orig=original_d_string;

% Compute chirp function
  t=0:1/44100:10000/44100;
  y=chirp(t,00,10000/44100,100,'log');

% Correlate input signal with chirp
% and recover sign
  for i=1:150
     yzcorr=xcorr(au(10000*(i-1)...
     +1:10000*i),y,0);

     r(i)=sign(yzcorr);
  end

% Recover bit stream
  for i=1:150
    if r(i)==-1
       recov(i)=0;
    else
       recov(i)=1;
    end
  end

% Convert from number to sring
  recov=(num2str(recov,-8));
```

```
% Covert from binary to decimal
% and concatenate
 rec_ene_dist...
 =cat(2,bin2dec(recov(1:31)),...
 bin2dec(recov(32:48)),...
 bin2dec(recov(49:65)),...
 bin2dec(recov(66:82)),...
 bin2dec(recov(83:99)),...
 bin2dec(recov(100:116)),...
 bin2dec(recov(117:133)),...
 bin2dec(recov(134:150)));

% Write out reconstructed decimal
% integer stream recoverd from
% watermark
  reconstructed_d_string=rec_ene_dist;
  reconstructed_d_string
  rec=reconstructed_d_string;

% Write out error between
% reconsructed and original
% watermark (decimal integer) codes.
   error...
   =sum(abs(rec-orig))/sum(abs(rec+orig))
```

ACKNOWLEDGMENT

The author is grateful for the advice and help of Dr S Datta and Dr O Farooq.

REFERENCES

[1] J. I. Cox, M. L. Miller and J. A. Bllom, *Digital Watermarking*, Morgan Kaufmann Publishers, Academic Press, 2002
[2] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.
[3] J. M. Blackledge, *Digital Signal Processing*, 2^{nd} Edition, Horwood Publishing, 2006.
[4] J. M. Blackledge, *Digital Image Processing*, Horwood Publishing, 2005.
[5] S. Katzenbaiser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech, 2000.
[6] B. Buck and V. A. Macaulay (Eds.), *Maximum Entropy in Action*, Oxford Science Publications, 1991.
[7] R. J. Anderson and F. A. P. Petitcolas, *On the Limits of Steganography*, IEEE Journal of Selected Areas in Communication (Special issue on Copyright and Privicy Protection), 16(4), 474-481, 1989.
[8] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, *Information Hiding - A Survey*, Proc. IEEE, 87(7), 1062-1078, 1999.
[9] A. Jazinski, *Stochastic Processes and Filtering Theory*, Academic Press, 1970.
[10] A. Papoulis, *Signal Analysis*, McGraw-Hill, 1977.
[11] A. Bateman and W. Yates W, *Digital Signal Processing Design*, Pitman, 1988.
[12] A. W. Rihaczek, *Principles of High Resolution Radar*, McGraw-Hill, 1969.
[13] R. L. Mitchell, *Radar Signal Simulation*, Mark Resources Incorporated, 1985.
[14] J. J. Kovaly, *Synthetic Aperture Radar*, Artech, 1976.
[15] M. Darnell (Ed.), *Cryptography and Coding*, Lecture Notes in Computer Science (1355), Springer, 1997.
[16] D. Kundur and D. Hatzinakos, *Digital Watermarking using Multi-resolution Wavelet Decomposition*, Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '98), IEEE, 2969-2972, 1997.
[17] D. M. A. Lumini, *A Wavelet-based Image Watermarking Scheme*, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '00), IEEE, 122-127, 2000.
[18] D. Kundur and D. Hatzinakos, A Robust Digital Image Watermarking Method using Wavelet based Fusion, Proceedings of the International Conference on Image Processing (ICAP '97), IEEE, 544-547, 1997.
[19] H. Tassignon, *Wavelets in Image Processing*, Image Processing II:Mathematical Methods, Algorithms and Applications (Eds. J M Blackledge and M J Turner), Horwood Publishing, 2000.
[20] M. V. Klein and T. E. Furtak, *Optics*, Wiley, 1986.
[21] J. J. Chae and B. Manjunath, *A Technique for Image Data Hiding and Reconstruction without a Host Image*, Security and Watermarking of Multimedia Contents I, (Eds. P W Wong and E J Delp), SPIE 3657, 386-396, 1999.

Jonathan Blackledge received a BSc in Physics from Imperial College, London University in 1980, a Diploma of Imperial College in Plasma Physics in 1981 and a PhD in Theoretical Physics from Kings College, London University in 1983. As a Research Fellow of Physics at Kings College (London University) from 1984 to 1988, he specialized in information systems engineering undertaking work primarily for the defence industry. This was followed by academic appointments at the Universities of Cranfield (Senior Lecturer in Applied Mathematics) and De Montfort (Professor in Applied Mathematics and Computing) where he established new post-graduate MSc/PhD programmes and research groups in computer aided engineering and informatics. In 1994, he co-founded Management and Personnel Services Limited (http://www.mapstraining.co.uk) where he is currently Executive Director for training and education. His work for Microsharp (Director of R & D, 1998-2002) included the development of manufacturing processes now being used worldwide for digital information display units. In 2002, he founded a group of companies specialising in information security and cryptology for the defence and intelligence communities, actively creating partnerships between industry and academia. He currently holds academic posts in the United Kingdom and South Africa, and in 2007 was awarded a Fellowship of the City and Guilds London Institute for his role in the development of the Higher Level Qualification programmes in engineering and computing, most recently for the nuclear industry.