DIGITAL WATERMARKING METHODS FOR DATA SECURITY AND AUTHENTICATION

Mary Lynne Hallot

Department of Computer Science
University of the Western Cape
Republic of South Africa

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy
Department of Computer Science, University of the Western Cape

Supervisors: Professor Bajic and Professor Blackledge

August, 2008

Abstract

Digital Watermarking Methods for Data Security and Authentication

Mary Lynne Hallot
PhD Thesis,
Department of Computer Science,
University of the Western Cape.

Throughout history people have tried to develop methods to secure the exchange of information. Due to the influence of the information technology revolution, protecting data of different types has become as essential requirement. The technological innovations of the military are responsible for a wide range of industrial and commercial advances and the implementation of the protocols and procedures associated with military data processing algorithms have influenced certain industry standards that are constantly being scrutinized and improved upon. Different standards have been developed for particular market sectors. For example, DES (Data Encryption Standard) was originally developed in the early 1970s and, in 1976, was selected by the Federal Information Processing Standard for use in the USA. Since that time, DES has had widespread use internationally and was upgraded to triple DES or DES3 in the 1990s (essentially, but not literally, a triple encryption version of DES in order to compensate for the relatively low key length associated with the original DES).

Specialized areas of information security are in confidentiality, data integrity, access control, identification, authentication and authorization. The implementation associated with these areas depend on the situation and requirement specifications. The design of new approaches, methodology and analysis of the strengths and weaknesses associated with a practical implementation are similar to the development of cryptographic algorithms. Private messaging (encrypting email attachments, for example) information security measures, financial transactions and a host of online services in information protection use specific mathematical techniques and the design of computational methods exclusive to the study of cryptography.

Cryptology is the study of systems that typically originate from a consideration of the ideal circumstances under which secure information exchange is to take place. It involves the

study of cryptographic and other processes that might be introduced for breaking the output of such systems - cryptanalysis. This includes the introduction of formal mathematical methods for the design of a cryptosystem and for estimating its theoretical level of security. A point of note is that the mathematical strength of a cryptographic algorithm is a necessary but not a sufficient requirement for a system to be acceptably secure. In the ideal case, the cryptographic strength of an algorithm and/or implementation method can be checked by means of proving its resistance to various kinds of known attacks. However, in practice, this does not mean that the algorithm and/or its specific application is secure because other unknown attacks may exist. For this reason, the security of a cryptosystem is often based on knowledge of its working legacy and the confidence level that a community has acquired from its continual use over many years often due to various up-grades, additions and modifications as have been considered necessary. Thus, modern systems for securing information exchange are often based to a large degree on past legacies associated with the performance of relatively well established techniques.

Whatever the type, strength and historical legacy of a cryptosystem, all such system suffer from the same fundamental problem. The fact that the data has been encrypted and is thus, not readable in the usual sense of the word, alerts a potential attacker to the fact that the information must be of value to have been encrypted in the first place. In other words, the act of encrypting information 'raises a flag' to the fact that the information has value. This fact can of course be used in the propagation of disinformation whereby the attacker is encouraged to decrypt an encrypted message so that the decrypt is acted upon by the attacker. However, while providing an approach that can provide a strategic advantage, this does not provide a solution to the basic problem of a ciphertext 'giving the game away'. A principal solution to this problem is to 'hide' the plaintext and/or the ciphertext in data that appears to be 'innocent' and unobtrusive and does not, by the very nature of its non-compliance with a cipherext, trigger an attack. This can be accomplished by embedding the ciphertext in images, for example. Another approach is to try a camouflage the ciphertext in noise such as in transmission noise, especially when the transmission noise is characteristic of the transmission environment through which a message is to sent. These approaches to 'hiding' information are concerned with the art of Steganography such as digital watermarking, for example.

In this thesis, although a background to encryption methods are presented, the focus of the research has been on steganography and in particular, the use of steganographic techniques for document security. The complexity in replicating digital media which has quality has necessitated novel techniques to restrain the forbidden reproduction, forgery and dissemination of digital data. We deem an embedding process to expose the process using

Independent Component Analysis (ICA) and presents a method that demonstrates how ICA can be used to detect forged documents. We show how robust digital watermarks used for copyright protection can be explained as undetectable information concealed in a digital media. This information is measurable as long as the quality of the content is considered as being acceptable. Despite many algorithms, methods, techniques and fully functional systems being used to hide information, the main dilemma is that the bulk of these methods use a symmetric key. The key becomes the'weakest link. In order to decipher or decode the hidden data it is necessary to know the secret key. Once the secret key is known, the watermark can not only be decoded, but also easily estimated and removed from the content. In such a state of affairs, a decoder used for copy protection has to be either put into operation as a tamper-proof device, or located in a trusted third party. These solutions are expensive and not necessarily plausible. We used ICA to separate our 2D DataGlyph from the watermark itself. Of interest is the fact that the 2D DataGlyph plays the role of a watermark and is created to mislead the attacker who would try to crack the watermark typically as a watermark is normally broken. This is done in the view that an attacker, not realizing that a different element had been introduced, will be foiled.

We embedded a 2D Dataglyph as a watermark signature into a watermark. In case of symmetric watermarks, the management of the keys is another challenge. This problem is rectified by ensuring that a watermarker implements the key and only they alone know the key. The 2D DataGlyph has its own key, whist the image maker has the original image. Each are separate and not shared amongst the others. The chances of all three keys being found are then minimal. Within these parameters we used ICA to see if it is possible to break the watermark. The ICA technique extracts linearly independent distinctive parts from a data-set. Unlike decorrelating data methods, ICA searches for directions in data-space which are independent across all statistical orders. It is capable of finding the underlying factors or sources when classic methods fail completely. The approach reported here has the advantage of including a learning probabilistic model (as opposed to projections in data-space), while remaining computationally efficient in high dimensions. A number of methods are explored with regard to this focus including, diffusion of information with noise and the use of DataGlyphs for the copy protection of documents. Moreover, we explore methods that can be used for both audio and image data and, in the latter case, consider both electronic image watermarking and similar techniques that can be applied to hardcopy images to secure paper-based documents with regard to authentication and anti-counterfeiting measures.

June, 2008

Acknowledgements

'Asante sana kwa wema ulionitendea'
'Thank you for your good deeds to me'

'Unyawo-alunampumlo' is a Xhosa proverb which relates to the spirit of Ubuntu. When living amongst strangers, one is defenseless and at the mercy of the local inhabitants. 'Unyawo-alunampumlo' refers to such persons of whom I have been one. I am in awe at the many diversions created which helped me out of my 'intellectual colonoscopy'.

It is a myth that a dissertation is the soul-wrenching creation solely of its author's time, toil and tenacity only. I have to salute my supervisors Professor Bajic, Professor Blackledge who showed me that a candle loses nothing by lighting another candle.

I have been fortunate for my many helpers. I risk doing them a disservice by not mentioning all of them here, but plead paucity of space. To mention a few: Professor E. Blake, Professor C. Omlin, Dr David Hecht, Professor I. Cox, Matt Miller, Matt Damien, P. Mulamba, Dr Lorna Holtman, Professor Isabelle Venter, Mr Reg Dodds's, Professor Jan van Bever-Donker Professor Kotze, the Watermarking Society and Professor Nelleke Bak, Daniel Leendarts and my friends, H. Voutsas, D. Dixon and my beloved family to whom total gratitude is mine. I also very grateful to everyone in the Department's of Computer Science, Mathematics and Statistics at the University of the Western Cape.

Finally, thanks go to Jennifer Hoffman, Coral Bell, Patricia D. Cota-Robles, Rhonda Byrnes, Catherine Moore, Dr S. Sands and the staff at the Sedona Journal for their kindness and constant support.

Thank you.

Declaration

I declare that *Digital Watermarking Methods for Applications to Document Security* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full name: M. L. Hallot
Date June, 2008

Signed

Notation and Glossary of Terms

Mathematical Notation

Alphabetic

a	Step length
A	Resultant step length
$A(t)$	Amplitude modulation (amplitude envelope)
$A(\omega)$	Amplitude spectrum
chirp(t)	Unit chirp function [with complex form $\exp(-i\alpha t^2)$]
D	Diffusivity
D_F	Fractal Dimension
D_T	Topological Dimension
E	Entropy
$f(t)$	Real or complex function - typically a system input
$\mid f \mid$	modulus of real or complex function f
$\|\|f(t)\|\|$	Norm (e.g. a Euclidean norm) of a function $f(t)$
$\|\|f(t)\|\|_\infty$	Uniform norm - maximum value of $f(t)$
$G(\mathbf{r} \mid \mathbf{r}_0, t \mid t_0)$	Time depedndent Green's function
$g(\mathbf{r} \mid \mathbf{r}_0, \omega)$	Time independent Green's function
H	Hurst exponent or dimension
$H(t)$	Tophat function

k	Wavenumber ($= 2\pi/\lambda$)
$n(t)$	Noise function
$N(k)$	Noise spectrum
n_i	Discrete noise function
N_i	Discrete Noise spectrum
$p(t)$	Impulse Response Function
p_i	Discrete Impulse Response Function
$P(\omega)$	Transfer Function (Fourier transform of $p(t)$)
P_i	Discrete Transfer Function (DFT of p_i
$P(x)$	Probability density function also denoted by $\Pr[x(t)]$
$P(\omega)$	Power spectrum ($=\mid F(\omega) \mid^2$) where $F(\omega)$ is the Fourier transform of $f(t)$
P_i	Discrete power spectrum
$\Pr(x)$	Probability occurrence of x
q	Fourier dimension
$s(t)$	Real or complex signal
s_i	Discrete real or complex signal
sgn(t)	Sign function
$\mathrm{sinc}(t)$	Sinc function ($= \sin(t)/t$)
t	Time
$u(\mathbf{r}, t)$	Field solution to a partial differential equation
u_0	$u(\mathbf{r}, t)$ at $t = 0$
$\in$	In (e.g. $x \in [a, b)$ is equivalent to $a \leq x < b$)
$\forall$	Forall (e.g. $f(t) = 0, \ \forall t \in (a, b]$)

Greek

α	Chirping parameter
$\Gamma(q)$	Gamma function
$\delta(t)$	Dirac delta function
δ^n	n-dimensional delta function
$\theta(t)$	Instantaneous phase
λ	Wavelength or scale length
σ	coeeficient of diffusion D^{-1} or standard deviation
ω	Angular frequency
Ω	Bandwidth of a spectrum

Operators

$\hat{D}^q$	Fractional differential operator
$\hat{\mathcal{F}}_1$	One dimensional Fourier transform
$\hat{\mathcal{F}}_1^{-1}$	One dimensional inverse Fourier transform
$\hat{\mathcal{F}}_n$	n-dimensional Fourier transform
$\hat{\mathcal{F}}_n^{-1}$	n-dimensional inverse Fourier transform
$\hat{H}$	Hilbert transform operator
$\hat{I}^q$	Fractional integral operator (e.g. Riemann-Liouville fractional integral)
$\hat{W}$	Wavelet transform
$\hat{W}^{-1}$	Inverse wavelet transform
$\otimes$	Convolution operation (continuous or discrete and causal or otherwise, depending on the context specified)
$\odot$	Correlation operation (continuous or discrete and causal or otherwise, depending on the context specified)
$\otimes\otimes$	two-dimensional convolution
$\otimes_{\mathbf{r}}$	convlution over $\mathbf{r}$
$\otimes_t$	convolution over t
$\Longleftrightarrow$	Transformation into Fourier space
$\longleftrightarrow$	Transformation into some transform space (as defined)

Terms

Mathematical and Statistical

DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Frequency Modulation or Fractal Modulation
FIR	Finite Impulse Response
GUI	Graphical User Interface
IRF	Impulse Response Function
IV	Initialization Vector
PDE	Partial Differential Equation
PDF	Probability Distribution or Density Function
PSDF	Power Spectral Distribution or Density Function
PSF	Point Spread Function
RSRA	ReScaled Range Analysis

Computer Science

ASCII	American Standard Code for Information Interchange
BIN	Binary
CCD	Charge Couple Device
CHAR	Character
DEC	Decimal
ICT	Information and Communications Technology
I/O	Input/Output
MOD	Modular
PCNG	Pseudo Chaotic Number Generator
PRNG	Pseudo Random Number Generator
XOR	Bit addition
USB	Universal Serial Bus

Organizational and Standards

AES	Advanced Encryption Standard
COTS	Commercial-Off-The-Shelf
DES	Digital Encryption Standard
DES3	Triple DES
GCHQ	Government Communications Head Quarters
JPEG	Joint Photographic Expert Group
NDA	Non Diclosure Agreement
OTP	One-Time Pad
PIN	Personal Identity Number
PKI	Public Key Infrastructure
RIP	Regulation of Investigatory Powers
RSA	Rivest, Shamir and Adleman

Terms Specific to Cryptology

Asymmetric Encryption
A cryptosystem in which encryption and decryption are performed using two different keys, one of which is referred to as the public key and the other as a private key. Also known as public key encryption.

Authentication
A process used to verify the integrity of transmitted data, especially a message.

Block Cipher
A symmetric encryption algorithm in which a large block of plaintext bits is transformed as a whole into a ciphertext block of the same length.

Cipher
An algorithm or data used for encryption and decryption. A cipher is that data which is used to replace a piece of information (an element in plaintext) with another object, with the intent of concealling its meaning. Typically, the replacement rule is governed by a secret key.

Ciphertext
The output of an encryption algorithm; the encrypted form of message data.

Code
A procedure for replacing a piece of information (e.g. letter, word, phrase) with another object, not necessarily of the same sort. Generally, there is no intent to conceal meaning. Examples include the 7-bit ASCII character code (each character is represented by 7 bits) and frequency-shift keying (each binary value is represented by a particular frequency).

Confusion
A cryptographic technique that seeks to make the relationship between the plaintext and the ciphertext as complex as possible. This is achieved by the use of a complex scrambling algorithm that depends on a key and the input with the aim of maximising the Entropy of the ciphertext.

Covertext
The output associated with the application of a steganographic technique.

Cryptanalysis
The branch of cryptology dealing with the breaking of a cipher to recover information, or forging encrypted information that will be accepted as authentic.

Cryptography
The branch of cryptology dealing with the design of algorithms for encryption and decryption, intended to ensure the secrecy and authenticity of messages.

Cryptology
The study of secure communications which encompasses both cryptography and cryptanalysis.

Cryptosystem
Any system that is concerned with the encryption and decryption of information.

Decryption
The translation of encrypted text or data (called ciphertext) into original text or data (called plaintext). Also called deciphering.

Differential Cryptanalysis
A technique in which chosen plaintexts with particular operational difference patterns are

encrypted. The difference patterns of the resulting ciphertext provide information that can be used to determine the encryption key.

Diffusion
A cryptographic technique that seeks to obscure the statistical characteristics of the plaintext by spreading out the influence of each individual plaintext elements over many ciphertext elements.

Digital Signature
An authentication mechanism that enables the creator of a message to attach a code which acts as a signature. The signature guarantees the source and integrity of the message.

Encryption
The conversion of plaintext or data into an unintelligible form by means of a public (or private) algorithm from which the plaintext can be recovered by decryption. Also called enciphering.

Hash Function
A function that maps a variable length data block or message into a fixed length value called a hash code. The function is designed in such a way that, when protected, it provides an authenticator for the data or message. Also referred to as a message digest.

Initialisation Vector (IV)
A random block of data that is used to begin the encryption of multiple blocks of plaintext, when a block-chaining encryption technique is used. The IV serves to foil known-plaintext attacks.

One-Way Function
A function that is easily computed, but where the calculation of its inverse is not feasible.

Plaintext
The input to an encryption function or the output to a decryption function.

Private Key
One of the two keys used in an asymmetric encryption system. For secure communication, the private key should only be known to its user.

Pseudo Chaotic Number Generator (PCNG)
A function that deterministically produces a sequence of numbers that are apparently chaotic.

Pseudo Random Number Generator (PRNG)
A function that deterministically produces a sequence of numbers that are apparently random.

Public Key
One of the two keys used in an asymmetric encryption system. The public key is made public and is used in conjunction with a corresponding private key.

Secret Key
The key is used in a symmetric encryption system. Both participants must share the same key, this key must remain secret to protect the communication.

Steganography
The technique associated with the hiding encrypted data in host data.

Stegotext The host data used to hide plaintext and/or ciphertext.

Stream Cipher
A symmetric encryption algorithm in which ciphertext output is produced bit-by-bit or byte-by-byte from a stream of plaintext input.

Symmetric Encryption
A cryptosystem in which encryption and decryption are performed using the same key. Also known as conventional encryption.

Contents

Chapter 1

Background to the Research Thesis

1.1 Introduction

In this thesis we concentrate on the functionality of watermarking and steganography and present how these cases in history have used applications that can surrender new technologies for secure information exchange invalid. Where we diverge in our line of attack is that cryptography and watermarking techniques just keep repeating the same mistakes. This Chapter discusses the various historical accounts encountered, the pitfalls, the patterns which emerge and their weak points. We illustrate what should have been done to secure the information henceforth. In Section 1.2, we discuss the history surrounding the area of investigation. Section 1.3 is based on Knowledge Management whereby we give a detailed discussion of how information is managed and protected and the pitfalls thereof. We discuss the issue of Privacy and the use of Disinformation and how it is disseminated, illustrating some of the managerial problems associated with Obfuscation for example. Section 1.4 explains the basic concepts where we illustrate the fundamental premises through which some of the theoretical ideas are based. In this chapter steganographic encryption is discussed together with the processes and methods which are explained in context of the main consensus of our thinking from Symmetric Encryption to Cryptanalysis. Finally in Section 1.8, we explain the unique contributions of Steganography in terms of some of the historical achievements and the improvements to be initiated.

1.2 Information and Communications Security

The quest for inventing innovative techniques which allow only authorized users to transfer information that is impervious to attack by others has, and continues to be, an essential

requirement in the communications industry. This requirement is based on the importance of keeping certain information secure, obvious examples being military communications and financial transactions, the former example being a common theme in the history and development of cryptology [1].

The Information and Communications Technology (ICT) revolution associated with the latter part of the Twentieth Century has brought about a number of significant changes in the way we operate on a routine basis. One of the most significant of these changes is the impact ICT has had upon basic human activities such as decision making, information processing and knowledge management. Business communities and government organisations rely heavily on exchanging, sharing and processing information to assist them in making a variety of strategic decisions and a wide range of security infrastructures have been established to help protect and preserve the integrity of information flowing across different channels.

At a government level, knowledge of public opinion allows politicians to react rapidly in their policies or programmes (at least in those cases where the infrastructure of society is based on democratic principles). In the commercial sector, 'know how' contributes considerably to company market value because information is a primary competitive advantage. Information is now a key factor in decision making for all organisations, whatever their size and complexity, and is arguably the most important asset of an organisation. The data transferred between different locations and recipients can therefore become vulnerable to being intercepted and altered by a capable and interested eavesdropper, and information exchange through the application of a secure infrastructure has, therefore, become an essential component in all forms of knowledge management. In other words, 'knowledge is power' and since all power must be contained, it is necessary to provide continuous improvements to securing communications in order to maintain parity with the pace and growth of ICT in general (e.g. [2] - [6]). This thesis focuses on the applications of watermarking and steganography in cryptology and provides examples of how this application can yield new technologies for secure information exchange.

1.3 Cryptology

Cryptography is the study of mathematical and computational techniques related to aspects of information security (e.g. [7]-[9]). The word is derived from the Greek *Kryptos*, meaning hidden, and is related to disciplines such *Cryptanalysis* and *Cryptology*. Cryptanalysis is the art of breaking cryptosystems by developing techniques for the retrieval of information from encrypted data [10]. Cryptology is the science that underpins cryptography and cryptanalysis and can include a broad range of mathematical concepts, computational algorithms

and technologies. In other words, Cryptology is a multi-disciplinary subject that covers a wide spectrum of different disciplines and increasingly involves using a range of engineering concepts and technologies through the innovation associated with term 'technology transfer'. Figure 1.1 shows some example subject areas associated with modern Cryptology [11]. These include areas such as Synergetics, which is an interdisciplinary science explaining the formation and self-organization of patterns and structures in non-equilibrium open systems and Semiotics, which is the study of signs and symbols, both individually and grouped in sign systems including the study of how meaning is constructed and understood.

Cryptology is often concerned with the application of formal mathematical techniques to design a cryptosystem and to estimate its theoretical security. This can include the use of formal methods for the design of security software which should ideally be a 'safety critical system' [12]. Although the mathematically defined and provable strength of a cryptographic algorithm or cryptosystem is necessary, it is not a sufficient requirement for a system to be acceptably secure. This is because it is difficult to estimate the security of a cryptosystem in any formal sense when it is implemented in the field under conditions that can not always be predicted and thus, simulated. The security associated with a cryptosystem can be checked only by means of proving its resistance to various kinds of known attack that are likely to be implemented. However, in practice, this does not mean that the system is secure since other attacks may exist that are not included in simulated or test conditions. The reason for this is that humans possess a broad range of abilities from unbelievable ineptitude to astonishing brilliance which can not be formalised in a mathematical sense or on a case by case basis.

The practical realities associated with Cryptology are indicative of the fact that 'Security is a process, not a product' [13]. Whatever the sophistication of the security product (e.g. the encryption and/or key exchange algorithm(s), for example), unless the user adheres strictly to the procedures and protocols designed for its use, the 'product' can be severely compromised.

A good example of this is the use of the Enigma [14] cipher by Germany during the second world war. It was not just the 'intelligence' of the 'code breakers' at Bletchley Park in England that allowed the allies to break many of the Enigma codes but the 'irresponsibility' and, in many cases, the sheer stupidity of the way in which the system was used by the German armed and intelligence services at the time. The basic mechanism for the Enigma cipher, which had been developed as early as 1923 by Artur Schubius for securing financial transactions, was well known to the allies (thanks to the efforts of the Polish Cipher Office at Poznan in the 1930s) and the distribution of some 10000 similar machines (with relatively minor modifications) to the German army, navy and air force was a disaster waiting to happen. The solution would have been to design a brand new encryption engine or better

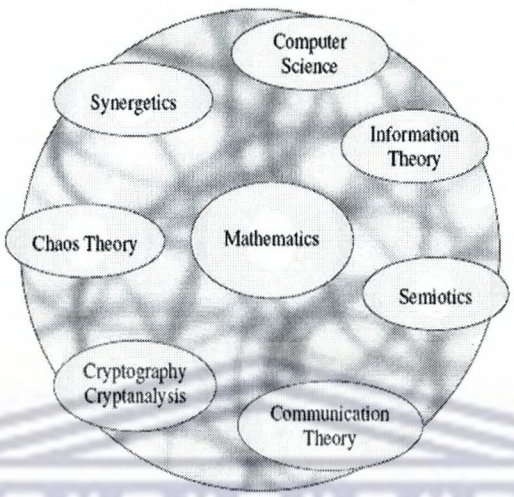


Figure 1.1: Example subject areas that contribute to developments in modern Cryptology [11].

still, a range of different encryption engines given the technology of the time, and use the Enigma machine to propagate disinformation. Indeed, some of the new encryption engines introduced by the Germans towards the end of the Second World War were not broken by the allies. These historically intriguing insights are easy to contemplate in hindsight, but they can also help to focus on the methodologies associated with developing new technologies for knowledge management.

1.4 Knowledge Management

With regard to information security and knowledge management in general, there are some basic concepts that are easy to grasp and sometimes tend to get lost in the detail. The first of these is that the recipient of any encrypted message must have some form of *a priori* knowledge on the method (the algorithm for example) and the operational conditions (e.g. the key) used to encrypt a message. Otherwise, the recipient is in no better 'state of preparation' than the potential attacker. The conventional approach is to keep this *a priori* information to a minimum but in such a way that it is critical to the decryption process. Another important reality is that in an attack, if the information transmitted is not deciphered in good time, then it may become redundant. Coupled with the fact that an attack usually has to focus on

a particular criterion (a specific algorithm for example), one way to enhance the security of a communications channel is to continually change the encryption algorithm and/or process offered by the technology currently available.

Another approach to knowledge management is to disguise or camouflage the encrypted message in what would appear to be 'innocent' or 'insignificant' data such as a digital photograph of a holiday snap-shot, a music file or both, for example[1]. This is known as *Steganography* [19]-[21]. Further, the information security products themselves should be introduced and 'organised' in such a way as to reflect their apparent insignificance in terms of both public awareness and financial reward. This is of course contrary to the dissemination of many encryption systems, a process that is commonly perceived as being necessary for business development through the establishment of a commercial organisation, international patents, distribution of marketing material, elaborate and sophisticated Web sites, authoritative statements on the strength of a system to impress customers, publications and so on. Thus, a relatively simple but often effective way of maintaining security with regard to the use of an encryption system is to not tell anyone about it. The effect of this can be enhanced by publishing other systems and products that are designed to mislead the potential attacker. In this sense, ICT security products should be treated in the same way as many organisations treat a breach of security, i.e. not to publish the breach in order to avoid embarrassment and loss of faith by the client base.

1.4.1 Keeping it Quiet

A classic mistake (of historical importance) of not 'keeping it quiet', in particular, not maintaining 'silent warfare' [22], was made by Winston Churchill when he published his analysis of World War I. In his book *The World Crisis 1911-1918* published in 1923, he stated that the British had deciphered the German Naval codes for much of the war as a result of the Russians salvaging a code book from the small cruiser *Magdeburg* that had ran aground off Estonia on August 27, 1914. The code book was passed on to Churchill who was, at the time, the First Sea Lord. This helped the British maintain their defences with regard to the German navy before and after the Battle of Jutland in May, 1916. The German navy became impotent and forced the Germans to turn their attention to unrestricted submarine warfare. In turn, this led to an event (the sinking on May 7, 1915 of the Lusitania, torpedoed by a German submarine, the U-20) that galvanized American opinion against Germany and played a key role in the United States' later entry into World War I on April 17, 1917 and the defeat of Germany [23], [24].

[1]By encoding the encrypted message in the least significant bit or bit-pair of the host data, for example.

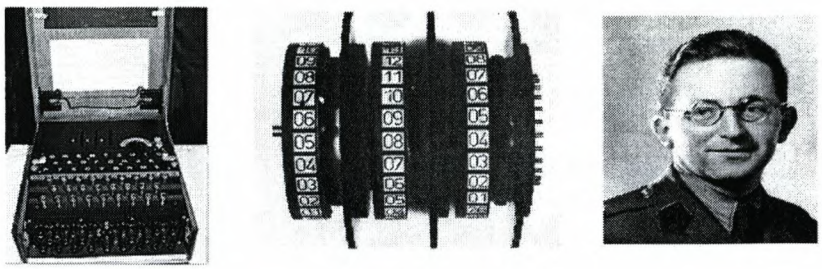


Figure 1.2: A military grade Enigma cipher (left), the three cipher rotors (centre) and a photograph of Marian Rejewski who invented the *Bomba kryptologiczna*, the basis for the deciphering machines constructed at Bletchley Park.

Churchill's publication did not go un-noticed by the German military between the First and Second World wars. Consequently significant efforts were made to develop new encryption devices for military communications. This resulted in the famous Enigma machine, named after Sir Edward Elgar's masterpiece, the *Enigma Variations* [25]. This was an electro-mechanical machine about the size of a portable typewriter (see Figure 1.2), with a standard keyboard, three (Figure 1.2) and later, four interchangeable rotors, and a number of plug connectors. The rotors and plugs offered 200 quintillion permutations. The machine could be used without difficulty by semi-skilled operators under the most extreme battle conditions. The rotor settings could be changed daily or several times a day according to the number of messages transmitted, after which the rotors returned to their original setting. The interest in cryptology by Germany that was undoubtedly stimulated by Churchill's indiscretions included establishing a specialist cipher school in Berlin. Ironically, it was at this School that some of the Polish mathematicians were trained who later worked for the Polish Cipher Office, opened in utmost secrecy at Poznan in 1930 [26], [27]. In January 1929, the Dean of the Department of Mathematics, Professor Zdzislaw Krygowski from the University of Poznan, provided a list of his best graduates to start working at this office. One of these graduates was the brilliant young logician, Marian Rejewski (see Figure 1.2) who pioneered the design of the *Bomba kryptologiczna*, an electro-mechanical device used for eliminating combinations that had not been used to encrypt a message with the Enigma cipher [28]. However, the design of the *Bomba kryptologiczna* was only made possible through the Poles gaining access to the Enigma machine and obtaining knowledge of its mechanism without alerting the Germans to their activities. In modern terms, this is equivalent to obtaining information on the type of encryption algorithm used in a cryptosystem.

The *Bomba kryptologiczna* helped the Poles to decipher some 100,000 Enigma messages

from as early January 1933 to September 1939 including details associated with the remilitarization of the Rhine Province, Anschluss of Austria and seizure of the Sudetenland. It was Rejewski's original work that formed the basis for designing the advanced electro-mechanical and later, the electronic decipher machines (including 'Colossus' - the world's first programmable computer) constructed and utilized at Bletchley Park between 1943 and 1945 [29], [30].

After the Second World War, Winston Churchill made sure that he did not repeat his mistake, and what he referred to as his 'Ultra-secret' - the code breaking activities undertaken at Station X in Bletchley Park, England - was ordered by him to be closed down and the technology destroyed soon after the end of the war. Further, Churchill never referred to his Ultra-secret in any of his publications after the war. Those personnel who worked at Bletchley Park were required to maintain their silence for some fifty years afterwards and some of the activities at Bletchley Park remain classified to this day. Bletchley park is now a museum which includes a reconstruction of 'Colossus' (see Figure 1.3) undertaken in the mid-1990s. However, the type of work undertaken there in the early 1940s continues in many organisations throughout the world such as the Government Communications Head Quarters (GCHQ) based at Cheltenham in England [31] where a range of 'code making' and 'code breaking' activities continue to be developed.

The historical example given above clearly illustrates the importance of maintaining a level of secrecy when undertaking cryptographic activities. It also demonstrates the importance of not publishing new algorithms, a principle that is at odds with the academic community; namely, that the security of a cryptosystem should not depend upon algorithm secrecy. However, this has to be balanced with regard to the dissemination of information in order to advance a concept through peer review, national and international collaboration. Taken to an extreme, the secrecy factor can produce a psychological imbalance that is detrimental to progress. Some individuals like to use confidential information to enhance their status. In business, this often leads to issues over the signing of Non-Disclosure Agreements or NDAs, for example, leading to delays that are of little value, especially when it turns out that there is nothing worth disclosing. Thus, the whole issue of 'keeping it quiet' has to be implemented in a way that is balanced, such that confidentiality does not lead to stagnation in the technical development of a cryptosystem. However, used correctly and through the appropriate personality, issues over confidentiality coupled with the 'feel important' factor can be used to good effect in the dissemination of disinformation.

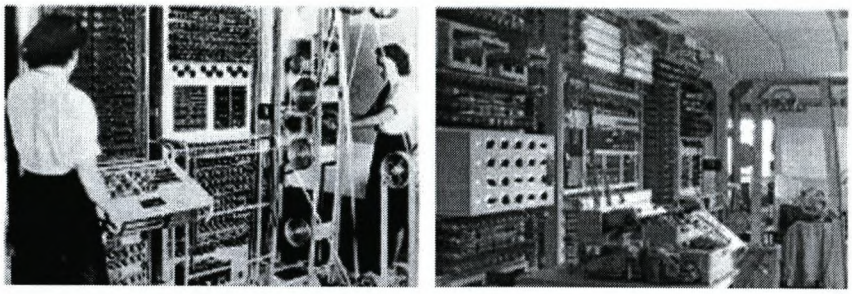


Figure 1.3: The Colossus Mark II computer (left) design by T Flowers at the Post Office Research Station was first installed at Bletchley Park in June 1944. The machine - which was, in effect, the worlds first electronic digital computer - was reconstructed at Bletchley Park in the mid-1990 (right).

1.4.2 Disinformation

Disinformation is used to tempt the enemy into believing certain kinds of information. The information may not be true or contain aspects that are designed to cause the enemy to react in an identifiable way that provides a strategic advantage [32], [33]. Camouflage, for example, is a simple example of disinformation [34]. This includes techniques for transforming encrypted data into forms that resemble the environments through which an encrypted message is to be sent. At a more sophisticated level, disinformation can include encrypted messages that are created with the sole purpose of being broken in order to reveal information that the enemy will react to by design.

Disinformation includes arranging events and processes that are designed to protect against an enemy acquiring knowledge of a successful encryption technology and/or a successful attack strategy. A historically significant example of this involved the Battle of Crete which began on the morning of 20 May 1941 when Nazi Germany launched an airborne invasion of Crete under the code-name Unternehmen Merkur (operation mercury) [35]. During the next day, through miscommunication and the failure of Allied commanders to grasp the situation, the Maleme airfield in western Crete fell to the Germans which enabled them to fly in heavy reinforcements and overwhelm the Allied forces. This battle was unique in two respects: it was the first airborne invasion in history [2]; it was the first time the Allies made significant use of their ability to read Enigma codes. The British had known for some weeks prior to the invasion of Crete that an invasion was likely because of the work being

[2]Illustrating the potential of paratroopers and so initiating the Allied development of their own airborne divisions.

undertaken at Bletchley Park. They faced a problem because of this. If Crete was reinforced in order to repel the invasion then Germany would suspect that their encrypted communications were being compromised. But this would also be the case if the British and other Allied troops stationed on Crete were evacuated. The decision was, therefore, taken by Churchill to let the German invasion proceed with success but not without giving the invaders a 'bloody nose'. Indeed, in light of the heavy casualties suffered by the parachutists, Hitler forbade further airborne operations and Crete was dubbed 'the graveyard of the German parachutists'. The graveyard for German, British, Greek and Allied soldiers alike was not a product of a fight over desirable and strategically important territory (at least for the British). It was a product of the need to secure Churchill's 'Ultra-secret'. In other words, the Allied efforts to repulse the German invasion of Crete was, in reality, a form of disinformation, designed to secure a secret that was, in the bigger picture, more important than the estimated 16,800 dead and wounded that the battle cost.

1.4.3 Plausible Deniability

Deniable encryption allows an encrypted message to be decrypted in such a way that different and plausible plaintexts can be obtained using different keys [36]. The idea is to make it impossible for an attacker to prove the existence of the real message, a message that requires a specific key. This approach provides the user with a solution to the 'gun to the head problem' as it allows the sender to have plausible deniability if compelled to give up the encryption key. There are a range of different methods that can be designed to implement such a scheme. For example, a single ciphertext can be generated that is composed of randomised segments or blocks of data which correlate to blocks of different plaintexts encrypted using different keys. A further key is then required to assemble the appropriate blocks in order to generate the desired decrypt. This approach, however, leads to ciphertext files that are significantly larger than the plaintexts they contain. On the other hand, a ciphertext file should not necessarily be the same size as the plaintext file and padding out the plaintext before encryption can be used to increase the Entropy of the ciphertext.

Other methods used for deniable encryption involve establishing a number of abstract 'layers' that are decrypted to yield different plainexts for different keys. Some of these layers are designed to include so-called 'chaff layers'. These are layers that are composed of random data which provide the owner of the data to have plausible deniability of the existence of layers containing the real ciphertext data. The user can store 'decoy files' on one or more layers while denying the existence of others, identifying the existence of chaff layers as required. The layers are based on file systems that are typically stored in a single

directory consisting of files with filenames that are either randomized (in the case where they belong to chaff layers), or are based on strings that identify cryptographic data, the timestamps of all files being randomized throughout.

1.4.4 Obfuscation

In a standard computing (windows) environment, a simple form of camouflage can be implemented by renaming files to be of a different type; for example, storing an encrypted data file as a .exe or .dll file. Some cryptosystems output files with identifiable extensions such as .enc which can then be simply filtered by a firewall. Another example includes renaming files in order to access data and/or execute an encryption engine. For example, by storing an executable file as a .dll (dynamic link library) file (which has a similar structure to a .exe file) in a directory full of real .dll files associated with some complex applications package, the encryption engine can be obfuscated, especially if it has a name that is similar to the environment of files in which it is placed. By renaming the file back to its 'former self', execution of cryptosystem can be undertaken in the usual way.

1.4.5 Steganographic Encryption

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

1.5 Basic Concepts

Irrespective of the wealth of computational techniques that can be invented to encrypt data, there are some basic concepts that are a common theme in modern cryptography. The application of these concepts typically involves the use of random number generators and/or the use of algorithms that originally evolved for the generation of random number streams, algorithms that are dominated by two fundamental and interrelated themes [4], [5], [6]:

- the use of modular arithmetic;
- the application of prime numbers.

The application of prime numbers is absolutely fundamental to a large range of encryption processes and international standards such as PKI (Public Key Infrastructure) details of which are discussed in Chapter 2.

Using a traditional paradigm, we consider the problem of how Alice (A) and Bob (B) can pass a message to and from each other without it being compromised or 'attacked' by an intercept. As illustrated in Figure 1.4, we consider a simple box and combination lock scenario. Alice and Bob can write a message, place it in the box, lock the box and then send it through an open 'channel' - the postal services, for example. In cryptography, the

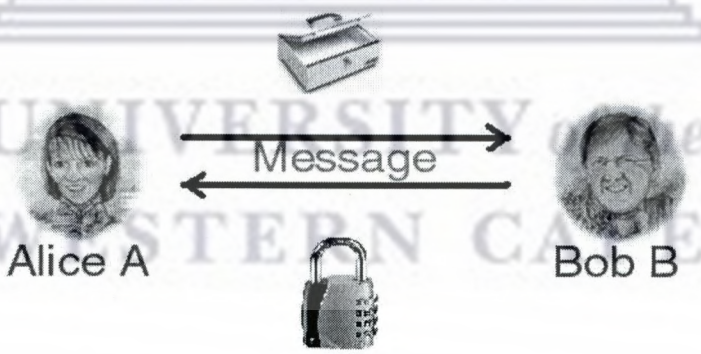


Figure 1.4: Alice and Bob can place a message in a box which can be secured using a combination lock and sent via a public network - the postal service, for example.

strength of the box is analogous to the strength of the cipher. If the box is 'weak' enough to be opened by brute force, then the strength of the lock is relatively insignificant. This is analogous to a cipher whose statistical properties are poor, i.e. whose PDF is narrow and whose information Entropy is relatively low, with a similar value to the plaintext. The strength of the lock is analogous to the strength of the key in a real cryptographic system.

This includes the size of the combination number which is equivalent to the length of the key that is used. Clearly a four rotor combination lock as illustrated in Figure 1.4 represents a very weak key since the number of ordered combinations required to attempt a brute force attack to open the lock are relatively low, i.e. for a 4-digit combination lock where each rotor has ten digits 0-9, the number of possible combinations is 10000 (including 0000). However, the box-and-lock paradigm being used here is for illustrative purposes only.

1.5.1 Symmetric Encryption

Symmetric encryption is the simplest and most obvious approach to Alice and Bob sending their messages. Alice and Bob agree on a combination number *a priori*. Alice writes a message, puts it in the box, locks it and sends it off. Upon receipt, Bob unlocks the box using the combination number that has been agreed and recovers the message. Similarly, Bob can send a message to Alice using exactly the same approach or 'protocol'. Since this protocol is exactly the same for Alice and Bob it has a symmetry and thus, encryption methods that adopt this protocol are referred to as symmetric encryption methods. Given that the box and the lock have been designed to be strong, the principal weakness associated with this method is its vulnerability to attack if a third party obtains the combination number at the point when Alice and Bob invent it and agree upon it. Thus, the principal problem in symmetric encryption is how Alice and Bob exchange the key. Irrespective of how strong the cipher and key are, unless the key exchange problem can be solved in an appropriate and a practicable way, symmetric encryption always suffers from the same fundamental problem - key exchange!

If E denotes the encryption algorithm that is used which depends upon a key K to encrypt plaintext P, then we can consider the ciphertext C to be given by

$$C = E_K(P)$$

Decryption can then be denoted by the equation

$$P = E_K(C)$$

Note that it is possible to encrypt a number of times using different keys $K_1, K_2, \ldots$ with the same encryption algorithm to give a double encrypted cipher text

$$C = E_{K_2}(E_{K_1}(P))$$

or a triple encrypted ciphertext

$$C = E_{K_3}(E_{K_2}(E_{K_1}(P)))$$

Decryption, is then undertaken using the same keys in the reverse order to which they have been applied, i.e.

$$P = E_{K_1}(E_{K_2}(E_{K_3}(C)))$$

Symmetric encryption systems, which are also referred to as shared secret systems or private key systems, are usually significantly easier to use than systems that employ different protocols (such as asymmetric encryption). However, the requirements and methods associated with key exchange sometimes make symmetric system difficult to use. Examples of symmetric encryption systems include the Digital Encryption Standard DES and DES3 (essentially, but not literally, the Digital Encryption Standard with triple encryption) and the Advanced Encryption Standard (AES). Symmetric systems are commonly used in many banking and other financial institutes and in some military applications. A well known historical example of a symmetric encryption engine, originally designed for securing financial transactions and later used for military communications was the Enigma used by the German military during World War II.

1.5.2 Asymmetric Encryption

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

One of the principal weaknesses of this approach relates to the lock being obtained by a third party whose interest is in sending bogus or disinformation to Alice. The problem for Alice is to find a way of validating that a message sent from Bob (or anyone else who is entitled to send messages to her) is genuine, i.e. that the message is authentic. Thus, data authentication becomes of particular importance when implementing asymmetric encryption systems.

Asymmetric encryption relies on both parties having two keys. The first key (the public key) is shared publicly. The second key is private, and is kept secret. When working with asymmetric cryptography, the message is encrypted using the recipients' public key. The recipient then decrypts the message using the private key. Because asymmetric ciphers tend to be computationally intensive (compared to symmetric encryption), they are usually used in combination with symmetric systems to implement public key cryptography. Asymmetric encryption is often used to transfer a session key rather than information proper - plaintext. This session key is then used to encrypt information using a symmetric encryption system. This gives the key exchange benefits of asymmetric encryption with the speed of symmetric encryption. A well known example of asymmetric encryption - also known as public key cryptography - is the RSA algorithm which is discussed in Chapter 2. This algorithm uses specific prime numbers (from which the private and public keys are composed) in order to realize the protocol. To provide users which such prime numbers, an infrastructure needs to be established by a third party whose 'business' is to distribute the public/private key pairs. This infrastructure is known as the Public Key Infrastructure or PKI. The use of a public key is convenient for those who wish to communicate with more than one individual and is thus, a many-to-one protocol that avoids multiple key-exchange. On the other hand, a public key provides a basis for cryptanalysis. Given that $C = E_K(P)$ where K is the public key, the analyst can guess P and check the answer by comparing C with the intercepted ciphertext, a guess that is made easier if it is based on a known Crib - i.e. information that can be assumed to be a likely component of the plaintext. Public key algorithms are therefore often designed to resist chosen-plaintext attack. Nevertheless, analysis of public key and asymmetric systems in general, reveals that the level of security is not as significant as that which can be achieved using a well-designed symmetric system. One obvious and fundamental issue relates to the third party responsible for the PKI and how much trust should be assumed, especially with regard to legislation concerning issues associated with the use of encrypted material.

1.5.3 Three-Way Pass Protocol

The three-way pass protocol, at first sight, provides a solution to the weaknesses associated with symmetric and asymmetric encryption. Suppose that Alice writes a message, puts it in the box, locks the box with a lock whose combination number is known only to her and sends it onto Bob. Upon receipt Bob cannot open the box, so Bob locks the box with another lock whose combination number is known only to himself and sends it back to Alice. Upon receipt, Alice can remove her lock and send the box back to Bob (secured with his lock

only) who is then able to remove his lock and recover the message. Note that by using this protocol, Alice and Bob do not need to agree upon a combination number; this avoids the weakness of symmetric encryption. Further, Alice and Bob do not need to send each other open locks which is a weakness of asymmetric encryption.

The problem with this protocol relates to the fact that it requires the message (secured in the locked box) to be exchanged three times. To explain this, suppose we have plaintext in the form of an ASCII-value array $p[i]$ say. Alice generates a cipher $n_1[i]$ using some appropriate strength random number generator and an initial condition based on some long integer - the key. Let the ciphertext $c[i]$ be generated by adding the cipher to the plaintext (process of confusion), i.e.

$$c_1[i] = p[i] + n_1[i]$$

which is transmitted to Bob. This is a substitution-based encryption process and is equivalent to Alice securing the message in the box with her lock - the first pass. Bob generates a new cipher $n_2[i]$ using the same (or possibly a different) random number generator with a different key and generates the ciphertext

$$c_2[i] = c_1[i] + n_2[i] = p[i] + n_1[i] + n_2[i]$$

which is transmitted back to Alice - the second pass. Alice now uses her cipher to generate

$$c_3[i] = c_2[i] - n_1[i] = p[i] + n_2[i]$$

which is equivalent to her taking off her lock from the box and sending the result back to Bob - the third pass. Bob then uses his cipher to recover the message, i.e.

$$c_3[i] - n_2[i] = p[i].$$

However, suppose that the cipher texts c_1, c_2 and c_3 are intercepted, then the plaintext array can be recovered since

$$p[i] = c_3[i] + c_1[i] - c_2[i].$$

This is the case for any encryption process that is commutative and associative. For example, if the arrays are considered to be bit streams and the encryption process undertaken using the XOR process (denoted by $\oplus$), then

$$\mathbf{c}_1 = \mathbf{n}_1 \oplus \mathbf{p}$$

$$\mathbf{c}_2 = \mathbf{n}_2 \oplus \mathbf{c}_1 = \mathbf{n}_2 \oplus \mathbf{n}_1 \oplus \mathbf{p}$$

$$\mathbf{c}_3 = \mathbf{n}_1 \oplus \mathbf{c}_2 = \mathbf{n}_2 \oplus \mathbf{p}$$

and

$$\mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \mathbf{c}_3 = \mathbf{p}$$

This is because for any bit stream **a**, **b** and **c**

$$\mathbf{a} \oplus \mathbf{a} \oplus \mathbf{b} = \mathbf{b}$$

and because the XOR operation is both commutative and associative i.e.

$$\mathbf{a} \oplus \mathbf{b} = \mathbf{b} \oplus \mathbf{a}$$

and

$$\mathbf{a} \oplus (\mathbf{b} \oplus \mathbf{c}) = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c}$$

These properties are equivalent to the fact that when Alice receives the box at the second pass with both locks on it, she can, in principle, remove the locks in any order. If, however, she had to remove Bob's lock before her own, then the protocol would become redundant.

1.5.4 Public-Private Key Encryption

Public-Private Key Encryption [40], [41] is fundamentally asymmetric and in terms of the box and combination-lock paradigm is based on considering a lock which has two combinations, one to open the lock and another to lock it. The second constraint is the essential feature because one of the basic assumptions in the use of combination locks is that they can be locked irrespective of the rotor positions. Thus, after writing a message, Alice uses one of Bobs specially designed locks to lock the box using a combination number that is unique to Bob but is openly accessible to Alice and others who want to send Bob a message. This combination number is equivalent to the public key. Upon reception, Bob can open the lock using a combination number that is known only to himself - equivalent to a private key. However, to design such a lock, there must be some mechanical 'property' linking the combination numbers required to first lock it and then unlock it. It is this property that is the principal vulnerability associated with public/private key encryption, a property that is concerned with certain precise and exact relationships that are unique to the use of prime numbers and their applications with regard to generating pseudo random number streams and stochastic functions in general [42].

1.6 Cryptanalysis

Any cryptographic system must be able to withstand cryptanalysis [43]. Cryptanalysis methods depend critically on the encryption techniques which have been developed and are,

therefore, subject to delays in publication. Cryptanalysts work on 'attacks' to try and break a cryptosystem. In many cases, the cryptanalysts are aware of the algorithm used and will attempt to break the algorithm in order to compromise the keys or gain access to the actual plaintext. It is worth noting that even though a number of algorithms are freely published, this does not in any way mean that they are the most secure. Most government institutions and the military do not reveal the type of algorithm used in the design of a cryptosystem. The rationale for this is that, if we find it difficult to break a code with knowledge of the algorithm then how difficult is it to break a code if the algorithm is unknown? On the other hand, within the academic community, security in terms of algorithm secrecy is not considered to be of high merit and publication of the algorithm(s) is always recommended. It remains to be understood whether this is a misconception within the academic world (due in part to the innocence associated with academic culture) or a covertly induced government policy (by those who are less innocent!). For example, in 2003, it was reported that the Americans had broken ciphers used by the Iranian intelligence services. What was not mentioned, was the fact that the Iranian ciphers were based on systems purchased indirectly from the USA and thus, based on USA designed algorithms [44].

The 'known algorithm' approach originally comes from the work of Auguste Kerchhoff. Kerchhoff's Principle states that: *A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.* This principle was reformulated by Claude Shannon as *the enemy knows the system* and is widely embraced by cryptographers world wide. In accordance with the Kerchhoff-Shannon principle, the majority of civilian cryptosystems make use of publicly known algorithms. The principle is that of 'security through transparency' in which open-source software is considered to be inherently more secure than closed source software. On this basis there are several methods by which a system can be attacked where, in each case, it is assumed that the cryptanalyst has full knowledge of the algorithm(s) used.

1.6.1 Basic Attacks

We provide a brief overview of the basic attack strategies associated with cryptanalysis.

Ciphertext-only Attack is where the cryptanalyst has a ciphertext of several messages at their disposal. All messages are assumed to have been encrypted using the same algorithm. The challenge for the cryptanalyst is to try and recover the plaintext from these messages. Clearly a cryptanalyst will be in a valuable position if they can recover the actual keys used for encryption.

Known-plaintext Attack makes the task of the cryptanalysis simpler because, in this case, access is available to both the plaintext and the corresponding ciphertext. It is then necessary to deduce the key used for encrypting the messages, or design an algorithm to decrypt any new messages encrypted with the same key.

Chosen-plaintext Attack involves the cryptanalyst possessing both the plaintext and the ciphertext. In addition, the analyst has the ability to encrypt plaintext and see the ciphertext produced. This provides a powerful tool from which the keys can be deduced.

Adaptive-chosen-plaintext Attack is an improved version of the chosen-plaintext attack. In this version, the cryptanalyst has the ability to modify the results based on the previous encryption. This version allows the cryptanalyst to choose a smaller block for encryption.

Chosen-ciphertext Attack can be applied when the cryptanalyst has access to several decrypted texts. In addition, the cryptanalyst is able to use the text and pass it though a 'black box' for an attempted decrypt. The cryptanalyst has to guess the keys in order to use this method which is performed on an iterative basis (for different keys), until a decrypt is obtained.

Chosen-key attack is based on some knowledge on the relationship between different keys and is not of practical significance except in special circumstances.

Rubber-hose Cryptanalysis is based on the use of human factors such as blackmail and physical threat for example. It is often a very powerful attack and sometimes very effective.

Differential Cryptanalysis is a more general form of cryptanalysis. It is the study of how differences in an input can affect differences in the output. This method of attack is usually based on a chosen plaintext, meaning that the attacker must be able to obtain encrypted ciphertexts for some set of plaintexts of their own choosing. This typically involves acquiring a Crib of some type as discussed in the following section.

Linear Cryptanalysis is a known plaintext attack which uses linear relations between inputs and outputs of an encryption algorithm which holds with a certain probability. This approximation can be used to assign probabilities to the possible keys and locate the one that is most probable.

1.6.2 Cribs

The problem with any form of plaintext attack is, of course, how to obtain part or all of the plaintext in the first place. One method that can be used is to obtain a Crib. A Crib, a term that originated at Bletchley Park during the Second World War, is a plaintext which is known or suspected of being part of a ciphertext. If it is possible to compare part of the ciphertext that is known to correspond with the plaintext then, with the encryption algorithm known, one can attempt to identify which key has been used to generate the cipherext as a whole and thus decrypt an entire message. But how is it possible to obtain any plaintext on the assumption that all plaintexts are encrypted in their entirety? One way is to analyse whether or not there is any bad practice being undertaken by the user, e.g. sending stereotyped (encrypted) messages. Analysing any repetitive features that can be expected is another way of obtaining a Crib. For example, suppose that a user was writing letters using Microsoft word, for example, having established an electronic letter template with his/her name, address, company reference number etc. Suppose we assume that each time a new letter is written, the entire document is encrypted using a known algorithm. If it is possible to obtain the letter template then a Crib has been found. Assuming that the user is not prepared to share the electronic template (which would be a strange thing to ask for), a simple way of obtaining the Crib could be to write to the user in hardcopy and ask that the response from the same user is of the same type, pleading ignorance of all forms of ICT or some other excuse. This is typical of methods that are designed to seed a response that includes a useful Crib. Further, there are a number of passive cribs with regard to letter writing that can be assumed, the use of *Dear* and *Yours sincerely*, for example.

During the Second World War, when passive cribs such as daily weather reports became rare through improvements in the protocols associated with the use of Enigma and/or operators who took their work seriously, Bletchley Park would ask the Royal Air Force to create some 'trouble' that was of little military value. This included seeding a particular area in the North Sea with mines, dropping some bombs on the French coast or, for a more rapid response, asking fighter pilots to go over to France and 'shoot up' targets of opportunity[3], processes that came to be known as 'gardening'. The Enigma encrypted ciphertexts that were used to report the 'trouble' could then be assumed to contain information such as the name of the area where the mines had been dropped and/or the harbour(s) threatened by the mines. It is worth noting that the ability to obtain cribs by gardening was made relatively easy because of the war in which 'trouble' was to be expected and to be reported. Coupled

[3]Using targets of opportunity became very popular towards the end of the war. Fighter pilots were encouraged to 'get them in the air, get them on the ground, just get them'.

with the efficiency of the German war machine with regard to its emphasis on accurate and timely reports, the British were in a privileged position in which they could create cribs at will and have some fun doing it.

When a captured and interrogated German stated that Enigma operators had been instructed to encode numbers by spelling them out, Alan Turing reviewed decrypted messages, and determined that the number 'eins' appeared in 90% of the messages. He automated the crib process, creating an 'Eins Catalogue', which assumed that 'eins' was encoded at all positions in the plaintext. The catalogue included every possible position of the various rotors and plug board settings of the Enigma. This provided a very simple and effective way of recovering the key and is a good example of how the statistics (of a word or phrase) can be used in cryptanalysis.

The use of Enigma by the German naval forces (in particular, the U-boat fleet) was, compared to the German army and air force, made secure by using a password from one day to the next. This was based on a code book provided to the operator prior to departure from base. No transmission of the daily passwords was required, passive cribs were rare and seeding activities were difficult to arrange. Thus, if not for a lucky break, in which one of these code books (which were printed in ink that disappeared if they were dropped in seawater) was recovered in tact by a British destroyer (HMS Bulldog) from a damaged U-boat (U-110) on May 9, 1941, breaking the Enigma naval transmissions under their time-variant code-book protocol would have been very difficult. A British Naval message dated May 10, 1941 reads: '1. Capture of U Boat 110 is to be referred to as operation Primrose; 2. Operation Primrose is to be treated with greatest secrecy and as few people allowed to know as possible...' Clearly, and for obvious reasons, the British were anxious to make sure that the Germans did not find out that U-110 and its codebooks had been captured and all the sailors who took part in the operation were sworn to secrecy. On HMS Bulldog's arrival back in Britain a representative from Bletchley Park, photographed every page of every book. The 'interesting piece of equipment' turned out to be an Enigma machine, and the books contained the Enigma codes being used by the German navy.

The U-boat losses that increased significantly through the decryption of U-boat Enigma ciphers led Admiral Carl Doenitz to suspect that his communications protocol had been compromised. He had no firm evidence, just a 'gut feeling' that something was wrong. His mistake was not to do anything about it[4], an attitude that was typical of the German High Command who were certifiable with regard to their confidence in the Enigma system. However, they were not uniquely certifiable. For example, on April 18, 1943, Admiral Yamamoto (the victor of Pearl Harbour) was killed when his plane was shot down while

[4]An instinct can be worth a thousand ciphers, ten-thousand if you like.

he was attempting to visit his forces in the Solomon Islands. Notification of his visit from Rabaul to the Solomon's was broadcast as Morse coded ciphertext over the radio, information that was being routinely decrypted by the Americans. At this point in the Pacific War, the Japanese were using a code book protocol similar to that used by the German Navy, in which the keys were changed on a daily basis, keys that the Americans had 'generated' copies of. Some weeks before his visit, Yamamoto had been given the option of ordering a new set of code books to be issued. He had refused to give the order on the grounds that the logistics associated with transferring new code books over Japanese held territory was incompatible with the time scale of his visit and the possible breach of security that could arise through a new code book being delivered into the hands of the enemy. This decision cost him his life. However, it is a decision that reflects the problems associated with the distribution of keys for symmetric cryptosystems especially when a multi-user protocol needs to be established for execution over a wide communications area. In light of this problem, Yamamoto's decision was entirely rational but, nevertheless, a decision based on the assumption that the cryptosystem had not already been compromised. Perhaps it was his 'faith in the system' and thereby his refusal to think the 'unthinkable' that cost him his life!

The principles associated with cryptanalysis that have been briefly introduced here illustrate the importance of using a dynamic approach to cryptology. Any feature of a security infrastructure that has any degree of consistency is vulnerable to attack. This can include plaintexts that have routine phrases such as those used in letters, the key(s) used to encrypt the plaintext and the algorithm(s) used for encryption.

1.7 Steganography

One of the principal weaknesses of all encryption systems is that the form of the output data (the ciphertext), if intercepted, alerts the intruder to the fact that the information being transmitted may have some importance and that it is therefore worth attacking and attempting to decrypt it. In Figure 1.4, for example, if a postal worker observed a locked box passing through the post office, it would be natural for them to wonder what might be inside. It would also be natural to assume that the contents of the box would have a value in proportion with the strength of the box/lock. These aspects of ciphertext transmission can be used to propagate disinformation, achieved by encrypting information that is specifically designed to be intercepted and decrypted. In this case, we assume that the intercept will be attacked, decrypted and the information retrieved. The key to this approach is to make sure that the ciphertext is relatively strong (but not too strong!) and that the information ex-

tracted is good quality in terms of providing the attacker with 'intelligence' that is perceived to be valuable and compatible with their expectations, i.e. information that reflects the concerns/interests of the individual(s) and/or organisation(s) that encrypted the data. This approach provides the interceptor with a 'honey pot' designed to maximize their confidence especially when they have had to put a significant amount of work in to 'extracting it'. The trick is to make sure that this process is not too hard or too easy. 'Too hard' will defeat the object of the exercise as the attacker might give up; 'too easy', and the attacker will suspect a set-up!

In addition to providing an attacker with a honey-pot for the dissemination of disinformation it is of significant value if a method can be found that allows the real information to be transmitted by embedding it in non-sensitive information after (or otherwise) it has been encrypted, e.g. camouflaging the ciphertext. This is known as *Steganography* which is concerned with developing methods of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message in contrast to cryptography in which the existence of the message itself is not disguised but the content is obscured [19], [20]. This provides a significant advantage over cryptography alone is that messages do not attract attention to themselves, to messengers, or to recipients. No matter how well plaintext is encrypted (i.e. how unbreakable it is), by default, a ciphertext will arouse suspicion and may in itself be incriminating, as in some countries encryption is illegal. With reference to Figure 1.4, *Steganography* is equivalent to transforming the 'strong box' into some other object that will pass through without being noticed - an 'egg-box', for example.

The word 'Steganography' is of Greek origin and means 'covered', or 'hidden writing'. In general, a steganographic message appears as something else known as a covertext. By way of a simple illustrative example, suppose we want to transmit the phrase

```
The Queen likes horses
```

which is encrypted to produce the cipher stream

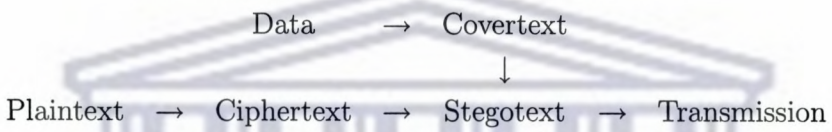
```
syoahfsuyTebhsiaulemNG
```

This is clearly a scrambled version of a message with no apparent meaning to the order of the letters from which it is composed. Thus, it is typical of an intercept that might be attacked because of the very nature of its incomprehensibility. However, suppose that the cipher stream above could be re-cast to produce the phrase

```
Beware of Greeks bearing gifts
```

If this phrase is intercepted it may not be immediately obvious that there is alternative information associated with such an apparently innocuous message, i.e. if intercepted, it is not clear whether or not it is worth initiating an attack.

The conversion of a ciphertext to another plaintext form is called *Stegotext* conversion and is based on the use of *Covertext*. Some covertext must first be invented and the ciphertext mapped on to it in some way to produce the stegotext. This can involve the use of any attribute that is readily available such as letter size, spacing, typeface, or other characteristics of a covertext, manipulated in such a way as to carry a hidden message. The basic principle is given below:



Note that this approach does not necessarily require the use of plaintext to ciphertext conversion as illustrated above and that plaintext can be converted into stegotext directly. A simple approach to is to use a mask to delete all characters in a message except those that are to be read by the recipient of the message. For example, consider the following message:

```
At what time should I confirm our activities? kindly acknowledge.
```

This seemingly innocent plaintext could be used to hide the message

```
Attack now
```

through application of the following mask:

```
1100001000000000000000000000000011000000000100000100011100000 0
```

where 0 denotes that a character or space is to be ignored and 1 denotes that a character or space is used. Apart from establishing a method of exchanging the mask which is equivalent to the key in cryptography, the principal problem with this approach is that different messages have to be continuously 'invented' in order to accommodate hidden messages and that these 'inventions' must appear to be legitimate. However, the wealth of data that is generated and transmitted in todays environment and the wide variety of formats that are used means that there is much greater potential for exploiting steganographic methods than were available before the IT revolution. In other words, the IT revolution has generated a camouflage rich environment in which to operate and one can attempt to hide plaintext or ciphertext (or both) in a host of data types, including audio and video files and digital images. Moreover, by understanding the characteristics of a transmission environment, it is

possible to conceive techniques in which information can be embedded in the transmission noise, i.e. where natural transmission noise is the covertext. There a some counter measures - steganalysis - that can be implemented in order to detect stegotext. However the technique usually requires access to the covertext which is then compared with the stegotext to see if any modifications have been introduced. The problem is to find ways of obtaining the original stegotext.

1.7.1 Hiding Data in Images

The relatively large amount of data contained in digital images makes them a good medium for undertaking steganography. Consequently digital images can be used to hide messages in other images. A colour image typically has 8 bits to represent the red, green and blue components. Each colour component is composed of 256 colour values and the modification of some of these values in order to hide other data is undetectable by the human eye. This modification is often undertaken by changing the least significant bit in the binary representation of a colour or grey level value (for grey level digital images). For example, the grey level value 128 has the binary representation 10000000. If 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. Further, if this is done for each colour component then a letter of ASCII text can be represented for every three pixels. The larger the host image compared with the hidden message, the more difficult it is to detect the message. Further, it is possible to hide an image in another image for which there are a number of approaches available (including the application of bit modification). For example, Figure 1.5 shows the effect of hiding one image in another through the process of re-quantization and addition. The image to be embedded is re-quantised to just 3-bits or 8 grey levels so that it consists of an array of values between 0 to 7. The result is then added to the host image (an array of values between 0 and 255) on an pixel by pixel basis such that if the output exceeds 255 then it is truncated (i.e. set to 255). The resulting output is slightly brighter with minor distortions in some regions of the image that are homogeneous.

Clearly, knowledge of the original host image allows the hidden image to be recovered (by subtraction) giving a result that is effectively completely black. However, by increasing its brightness, the hidden image can be recovered as shown in Figure 1.5 which, in this example, has been achieved by re-quantising the data from 0-7 back to 0-255 grey levels. The fidelity of this reconstruction is poor compared to the original image but it still conveys the basic

Figure 1.5: Illustration of 'hiding' one image (top left) in another image (top right) through simple re-quantisation and addition (bottom left). By subtracting the bottom left image from the top right image and re-quantising the output, the bottom right reconstruction is obtained.

information, information that could be covertly transmitted through the host image as an email attachment, for example. Note that the host image represents, quite literally, the key to recovering the hidden image. The additive process that has been applied is equivalent to the process of confusion that is the basis for a substitution cipher. Rather than the key being used to generate a random number stream using a pre-defined algorithm from which the stream can be re-generated (for the same key), the digital image is, in effect, being used as the cipher. Note that the distortion generated by re-quantization means that the same method can not be used if the hidden image is encrypted. The degradation in the ciphertext will not allow a decrypt to be accomplished. However, by diffusing the image with a noise field, it is possible to hide the output in a host image without having to resort to quantization.

Steganography is often used for digital watermarking. This is where the plaintext, which acts as a simple identifier containing information such as ownership, copyright and so on, is hidden in an image so that its source can be tracked or verified. This is equivalent to hiding a 2-bit image in a host image as illustrated in Figure 1.6 which uses the same method as discussed above. In this example, a columnar transposition cipher has been used to encrypt this sentence using the keyword: Steganography. This grid is given by

```
11 12 03 04 01 07 08 05 10 02 09 06 13
```

```
I n    t h i s    e x a m p
l e ,    a    c o l u m n a
r    t r a n s p o s i t i
o n    c i p h e r    h a s
b e e n    u s e d    t o
e n c r y p t    t h i s
s e n t a n c e    u s i n
g    t h e    k e y w o r d
:    S t e g a n o g r a p
h y
```

and the ciphertext is

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

As in the previous example, the host image is required to recover the ciphertext information and is thus the key to the process.

The methods discussed above refer to electronic-to-electronic type communications in which there is no loss of information. Steganography and watermarking techniques can be developed for hardcopy data which has a range of applications. These techniques have to be robust to the significant distortions generated by the printing and/or scanning process. A simple approach is to add information to a printed page that is difficult to see. For example, some modern colour laser printers, including those manufactured by HP and Xerox, print tiny yellow dots which are added to each page. The dots are barely visible and contain encoded printer serial numbers, date and time stamps. This facility provides a useful forensics tool for tracking the origins of a printed document which has only relatively recently been disclosed.

1.7.2 Hiding Data in Noise

The art of steganography is to use what ever covertext is readily available to make the detection of plaintext or, ideally, the ciphertext as difficult as possible. This means that the embedding method used to introduce the plaintext/cipherext into the covertext should produce a stegotext that is indistinguishable from the covertext in terms of its statistical

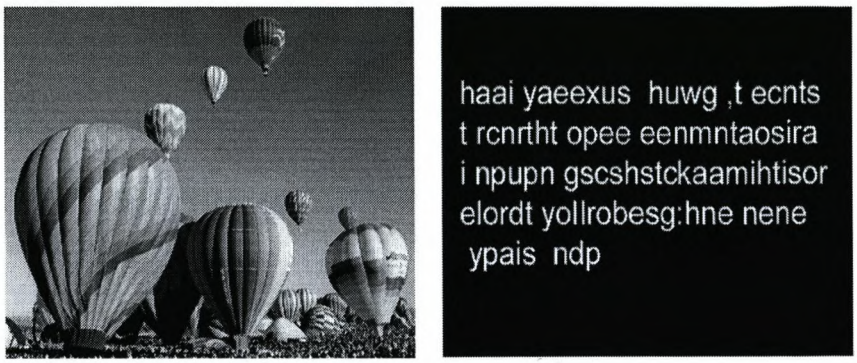


Figure 1.6: Binary image of encrypted information (right), obtained by subtraction of the covertext image from the stegotext image (left).

characteristics and/or the information it conveys. From an information theoretic point of view, this means that the covertext should have significantly more capacity than the ciphertext, i.e. there must be a high level of redundancy. Utilising noisy environments often provides an effective solution to this problem. There are three approaches that can be considered:

- embedding the ciphertext in real noise;
- transforming the ciphertext into noise that is then added to data;
- replacing real noise with ciphertext that has been transformed in to synthetic noise with exactly the same properties as the real noise.

In the first case we can make use of noise sources such as thermal noise, flicker noise, and shot noise associated with electronics that digitize an analogue signal. In digital imaging this may be noise from the imaging charge couplde device (CCD) element; for digital audio, it may be noise associated with the recording techniques used or amplification equipment. Natural noise generated in electronic equipment usually provides enough variation in the captured digital information that it can be exploited as a noise source to 'cover' hidden data. Because such noise is usually a linear combination of different noise types generated by different physical mechanisms, it is usually characterised by a normal or Gaussian distribution as a result of the Central Limit Theorem (see Chapter 3).

In the second case, the ciphertext is transformed into noise whose properties are consistent with the noise that is to be expected in certain data fields. For example, lossy compression schemes (such as JPEG - Joint Photographic Expert Group) always introduce some error

(numerical error) into the decompressed data and this can be exploited for steganographic purposes. By taking a clean image and adding ciphertext noise to it, information can be transmitted covertly providing all users of the image assume that it is the output of a JPEG or some other lossy compressor. Of course, if such an image is JPEG compressed, then the covert information may be badly corrupted.

In the third case, we are required to analyse real noise and derive an algorithm for its synthesis. Here, the noise has to be carefully synthesized because it may be readily observable as it represents the data stream in its entirety rather than data that is 'cloaked' in natural noise. This technique also requires that the reconstruction/decryption method is robust in the presence of real noise that we should assume will be added to the synthesized noise during a transmission phase. In this case, random fractal models are of value because the spectral properties of many noise types found in nature signify fractal properties to a good approximation. This includes transmission noise over a range of radio and microwave spectra, for example, and Internet traffic noise.

With regard to Internet traffic noise the time series data representing packet size and inter-arrival times shows well defined random fractal properties. There are a range of time-series models that can be used to charactersize Internet traffic noise based on the number of packets (or bytes) as a function of time. Fractal time-series models are applicable when the underlying processes have a similar appearance regardless of the time scale over which they are observed.

An example of the random fractal nature of Internet traffic is given in Figure 1.7 which shows the number of packets per unit time over four different time scales [45]. Compared with a time-series model based on Poisson statistics (i.e. a Poisson random number generator) as shown in Figure 1.7, it is clear that Internet traffic noise has fractal characteristics (i.e. has self-affine behaviour). Like real traffic, Internet traffic changes over a daily cycle according to the number of users. However, much of the traffic 'riding' the Internet can be modelled using fractals and as the Internet has become larger and larger, the fractal nature of the traffic has become more and more pronounced. The basis for the fractal nature of Internet traffic noise can be analysed in terms of information flow through a complex network of sites.

The use of fractal geometry for coding information and/or embedding it in random fractal noise has been considered in a number of publications (e.g. [46]-[50]). In this case, the stegotext is generated directly using a random fractal noise generator which modulates the value of the fractal dimension according to the bit type in an input bit stream. The output is a contiguous stream of fractal noise whose properties are 'tuned' to be consistent with the noisy environment in which a wireless communications system is operating. The technique

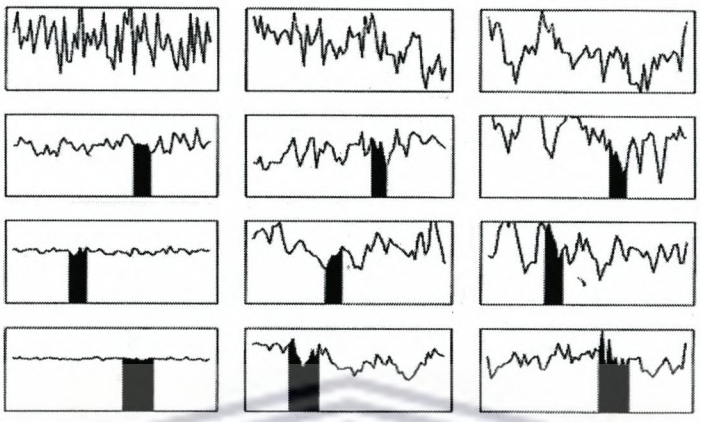


Figure 1.7: Simulated Poisson based time-series model (left), real Internet traffic time-series (centre) and simulated fractal time-series model taken over four different time scales (from top to bottom respectively). The shaded areas highlight the data displayed in each plots above respectively.

is analogous to using Frequency Modulation (FM) and is referred to as Fractal Modulation (also FM) which provides for the reconstruction of a bit stream with acceptable bit-error rates in the presence of expected signal-to-noise ratios. In addition to using random fractal noise to generate covertext and/or stegotext fields, the fractal fields themselves can be used as low resolution printable features (texture maps) that encode information in a way that is consistent with the principles of Fourier optics - convolution based encoding. This provides a method of information recovery that is robust to low resolution image acquisition technology and data degradation [51], [52], thus providing a COTS approach to securing the authenticity of printed materials.

1.8 About This Thesis

This thesis has been prepared in such a way that its focus on watermarking and steganography is presented in the context of modern cryptology. The purpose of this Chapter has been to introduce the reader to the basic themes associated with cryptology from both a historical context and a modern idiom. Irrespective of the details of an encryption algorithm and its implementation, the governing principles associated with cryptography are compounded in an understanding of the processes of diffusion and confusion. The basis of these processes are the focus for material given in Chapter 2 which provides an overview of digital watermarking

Chapter 3 considers the role of wavelets (e.g. the Morlet wavelet) for embedding plaintext or ciphertext in covertext. The background to image based watermarking schemes is the basis for Chapter 4 which provides a survey of the methods currently available. Image based information is the theme for the material considered in Chapter 5 which explores the use of steganography and digital watermarking for e-to-e communications and the application of (fractal) texture coding schemes for low resolution print security based on DataGlyphs. Finally, Chapter 6 provides a conclusion to the work and considers an example of further developments that can be undertaken.

1.9 Research Methodology

The research methodology applied in this thesis follows the principle of Experimental Software Engineering [5] which is a sub-field of software engineering. In this thesis, ESE is used to determine the suitability of algorithms for the development of watermarking methods that are based on the principles of cryptography, in particular, the diffusion and confusion of data.

ESE aims at applying empirical theories and methods for understanding and improving the software development process in organizations. It is a multi-disciplinary field, building upon subsidiary fields such as statistics, sociology, psychology, and computer science. The main objective of ESE is to evaluate tools, techniques, and technologies used in software engineering empirically, and thereby enhancing the state of the art and practice in software engineering. The method employed to achieve the goal is to run empirical investigations with the software development process as the object of study.

There is a five step research method for undertaking software engineering experiments:

1. Experiment definition. In this step the hypothesis is defined, along with the objectives and goals of the experiment. The hypothesis, at this stage, need not be formally stated, but should be stated clearly. The objective and goals are found by asking the following questions: what is studied? (object of study), what is the purpose of the experiment? which effect is studied? (quality focus) whose view is assumed? (perspective of the study) and where is the study conducted? (context).

2. Experiment planning. At this stage, the hypothesis is formalized including a null hypothesis. Input/independent variables and output/dependent variables are determined. A

[5] Wohlin et al., Experimentation in software engineering: An introduction, Kluwer Academic Publishers, London, 2000.

suitable experiment design is chosen, and the potential validity problems with the results are discussed.

3. Experiment operation. Subjects and the materials needed (for data collection forms) for the experiment is prepared, before executing the experiment. The primary objective of this stage is to gather data for the next step.

4. Analysis and interpretation. Descriptive statistics are used to understand the data gathered during step three. A possible reduction of the data set must be considered. After the data has been reduced, a hypothesis test is performed using statistical techniques.

5. Presentation and package. This stage is concerned with presenting and packaging the findings. It is important that the experiment design is clear, to allow for replication as this is an important mechanism for validating the findings.

Experimental software engineering is based on a hypothetic-deductive research model. The primary problem facing any empirical researcher is the wealth and complexity of the real world. The hypothetic-deductive research model 'solves this by reducing the scope of the empirical enquiry. The controlled experiment is a good example of such a reduction, where the empirical world is reduced to a set of independent and dependent variables, subjects, and the treatments applied to the subjects. The ontology of this research philosophy is that the real world can be divided into small bits that can later be reassembled into a complete picture.

1.10 Principal Hypothesis and Original Contributions

The principal hypothesis upon which this thesis is based concerns the use of data diffusion for generating watermarks that are robust to data degradation, and, as such, can be applied to authenticating data that may undergo severe degradation of some form. This includes the use of data diffusion for authenticating audio and image data and for printed materials in which degradation occurs in the print-scan process. In the context of this hypothesis, the principal contributions that are new and innovative as reported in this thesis are compounded in the following:

- Development of a new data authentication technique that is based on the use of data diffusion.
- Application of data diffusion using 'chirp coding' for audio data verification and authentication.

- Application of data diffusion using 'texture coding' for document verification and authentication.
- Design and implementation of a prototype system based on textured DataGlyphs.
- Application of Independent Component Analysis for the evaluation of DataGlyphs.

Chapter 2

Digital Watermarking

2.1 Introduction

The primary aim of digital watermarking is to design methods which provide for the authentication of data (encrypted or otherwise) but in a more universal significance, digital watermarking may be employed to hide information in data. Watermarking is an area of cryptology that is related to the 'art' of hiding secret information in data, or Steganography. In this Chapter, we provide an overview of digital watermarking and consider diversifications to watermarking from the vantage point of a watermarker. We consider our approach over cryptography in that if information can be embedded in a data field that is fully covert, then it is not immediately clear to a potential attacker that there may be information in the data that is worth attacking. We use the method of a watermarker and applied cryptography to confuse the attacker, who will approach our code by breaking it like a watermarker not realizing we have used cryptography to encode information. If we consider the case where this 'secret information' is an encrypted field, then watermarking can provide a method of covert encryption. In this chapter, we divide the area of interest and investigation into three segments. In order to understand the pitfalls of digital watermarking, we explain the principal components and applications, thereby introducing the process of applying of a digital watermark and the flaws that can be taken advantage of. Once the properties are investigated, one is able to understand the approach taken and the problems encountered regarding visibility which is the idea we exploit in terms of the new system developed and its robustness. The process of attacking and breaking a watermark is discussed in Secton 2.3 with regard to resistance to distortions, for example. Finally in Section 2.4, watermarking and cryptography is explained in order to introduce the principal differences associated with each field.

2.2 Digital Watermarking

Digital watermarking is an area of growing importance in information technology security as discussed in Chapter 1. One of the principal aims is to design methods which provide for the authentication of data (encrypted or otherwise) but in a more general sense, digital watermarking may be use to hide information in data.

The principal model that underpins this technology is based on the equation

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

where f is the information that is to embedded in the data (the watermark), $\hat{P}$ is some linear operator and s is the output data (the watermarked image). The function n, which is usually taken to be noise, can indeed be a noise field, but in the context of watermarking it can be any other host data. The operation $\hat{P}f$ can be non-stationary, in most watermarking methods, a stationary model is used. In such cases, the basic watermaking model is of the type

$$s = p \otimes f + n$$

where p is any useful function that is of value to the watermarking process.

Watermarking is an area of cryptology that is related to the 'art' of hiding secret information in data, or Steganography. It has an advantage over cryptography in that, if information can be embedded in a data field that is fully covert, then it is not immediately clear to a potential attacker that there may be information in the data that is worth attacking. If we consider the case where this 'secrete information' is an encrypted field, then watermarking can provide a method of covert encryption.

The operation $\hat{P}f$ describes the processes of 'diffusion' the process of adding noise (i.e. $\hat{P}f + n$) is the process of 'confusion'. For example, one of the processes $\hat{P}f$ that can be used is based on the convolution of f with the kernel

$$p(x) = \sqrt{\frac{1}{4\pi Dt}} \exp\left[-\left(\frac{x^2}{4Dt}\right)\right]$$

which is the Green function for the 1D diffusion equation with diffusivity D at time t, the operation $p \otimes f$ being a solution to the 1D diffusion equation for a source $f(x$ that starts to diffuse at $t = 0$.

This chapter considers approahes to digital watermarking and stegonography with a focus on the application of wavelets (in particular, a frequency modulated chirp). We also consider the use of fractal modulation for camouflaging data in an environment that is assumed to be characterised by random fractal noise.

2.3 Survey of Digital Watermarking Methods

In this section[1], we provide an overview of digital watermarking.

With rapid growth in computer networks and information technology, a large number of copyright works now reside in digital form. Further, electronic publishing is becoming increasingly popular. These developments in computer technology increase the problems associated with copyright protection and enforcement and thus, future developments in networked multimedia systems are conditioned by the development of efficient methods to protect ownership rights against unauthorized copying and redistribution. Digital watermarking has recently emerged as a candidate to solve this difficult problem. The mid-1990s saw the convergence of a number of different information protection technologies, whose theme was the hiding (as opposed to encryption) of information. Hiding can refer to either making the information imperceptible or keeping the existence of the information secret [53]. Important sub-disciplines of information hiding are Steganography and Watermarking. Steganography and watermarking are concerned with techniques that are used to imperceptibly convey information. However, they are two different and distinct disciplines: Watermarking, is the practice of hiding a message (copyright notices or individual serial numbers for example) about an image, audio clip, video clip, or other work of media within that work itself [53] without degrading its quality in such a way that it is expected to be permanently embedded into the data and can be detected later. Steganography, on the other hand, is the study of the techniques used to hide one message inside another, without disclosing the existence of the hidden message or making it apparent to an observer that this message contains a hidden message [134]. From the previous definitions we distinguished them as follows [53],[55]:

- The information hidden by a watermarking system is always associated with the object to be protected or its owner while steganographic systems just hide information.
- As the purpose of steganography is to have a covert communication between two parties whose existence is unknown to a possible attacker, a successful attack consists of detecting the existence of this communication. Watermarking, as opposed to steganography, has the additional requirement of robustness against possible attacks; even if the existence of the hidden information is known it should be hard for an attacker to destroy the embedded watermark. In other words, steganography is mainly concerned with detection of the hidden message while watermarking concerns potential removal by a pirate.

[1]Based on an edited version of K. W. Mahmoud, 'Low Resolution Watermarking for Print Security', PhD Thesis, Loughborough University, 2005, (Chpater 1).

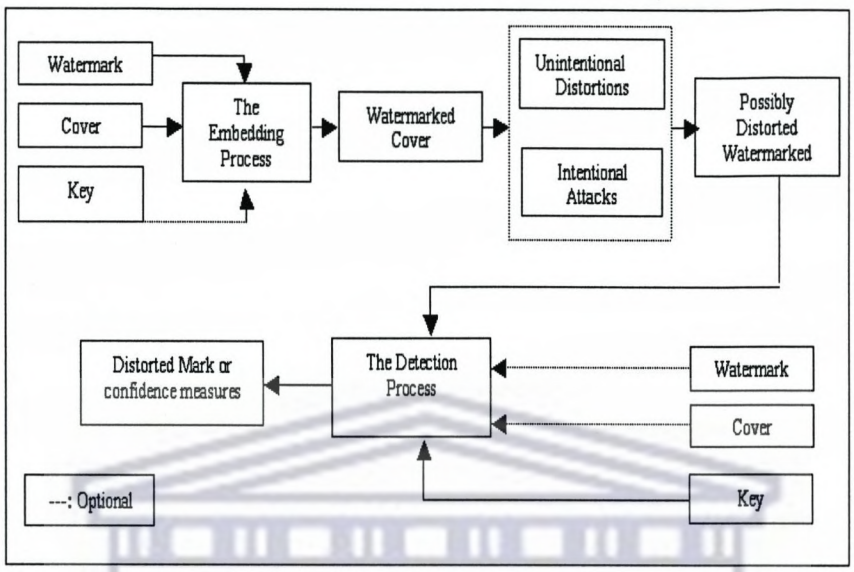


Figure 2.1: The general framework of a watermarking system [58]

- Steganographic communications are usually point-to-point (between sender and receiver) while watermarking techniques are usually one-to-many.

2.3.1 Principal Components

All watermarking schemes share the same generic building blocks (see Figure 2.1). These blocks and their functions are described below [56],[57].

Watermark Embedding System (Signature Casting): The embedded data is the watermark that one wishes to embed. It is usually hidden in a message referred to as a cover (work), producing the watermarked cover. The inputs to the embedding system are the watermark, the cover and an optional key. A key is used to control the embedding process so as to restrict detection and/or recovery of the embedded data to parties who know of it. The watermarked cover may face some intentional and/or unintentional distortion that may affect the existence of the watermark. The resultant outputs are called the 'Possibly Distorted Watermarked Cover'.

Watermark Detection System (Extraction): The inputs to the detection system are the possibly distorted watermarked cover, the key and depending on the method, the original cover or the original watermark. Its output is either the recovered watermark or some kind of confidence measure indicating how likely it is for a given watermark at the

input to be present in the work under inspection (e.g. Correlation). Current watermarking schemes may be viewed as spread-spectrum communications systems [53], whose aim is to send the watermark between two parties with two sources of noise; noise due to the original cover and noise due to processing.

2.3.2 Applications

In this section we discuss some of the scenarios where watermarking is already being used as well as other potential applications. The list given here [53],[55],[56],[59] is by no means complete and is intended to give a perspective of the broad range of possibilities that digital watermarking opens.

Owner identification. Embedding the identity of a work's copyright holder as a watermark in order to prevent other parties from claiming the copyright of the data.

Labeling. The hidden message can contain labels that, for example, allow for annotation of images or audio data. Of course, the annotation may also be included in a separate file, but with watermarking it becomes more difficult to destroy or lose this label, since it becomes closely tied to the object that it annotates. This is especially useful in medical applications since it prevents potentially dangerous errors.

Fingerprinting (Transaction Tracking). This is similar to the previous application and allows acquisition devices (such as video cameras, audio recorders, etc.) to insert information about the specific device (e.g., an ID number and date of creation). This is especially useful for identifying people who obtain content legally but illegally redistribute it. This can involve the embedding of a different watermark into each distributed copy.

Authentication. Embedding signature information in a work that can be later checked to verify if it has not been tampered with.

Copy and Playback Control. The message carried by the watermark may contain information regarding copy and display permissions. A secure module can be added in copy or playback equipment to automatically extract this permission information and block further processing if required. In order to be effective, this protection approach requires agreements between work providers and consumer electronics manufacturers to introduce compliant watermark detectors in their video players and recorders. This approach is being taken in Digital Videodiscs (DVD) for example.

Broadcast monitoring. Identifying when and where works are broadcast by recognizing watermarks embedded in the data.

Additional information. The embedded watermark could be an n-bit index to a database of URLs stored on a known location on the Internet. This index is used to fetch a corresponding URL from the database. The URL is then used to display the related web pages.

2.3.3 Classifications

Watermarking systems can be classified according to several aspects which are described below.

Private/Public Systems

Private Marking Systems (Informed Detector) require at least the original cover. This means that only the copyright holder can detect the watermark. In a private system, we can identify where the distortions were, and invert them before applying the watermark detector (using the original cover to reverse the embedding process or using the original work as a 'hint' to find where the watermark could be in the distorted watermarked cover). These types of systems may also require a copy of the embedded watermark for detection and just yield a 'YES' or 'NO' response to the question: does the distorted marked object contain this watermark? Private systems usually feature increased robustness (greater strength to the embedded bits) not only toward noise-like distortions, but also distortions in the data geometry since it allows the detection and inversion of geometrical distortion [59]. Unfortunately, for these techniques to be applied, the possibility to access the original image must be granted. This mean that the set-up of a watermarking system becomes more complicated, and on the other hand, the owners of the original images are compelled to insecurely share their works with anyone who wants to check the existence of the watermark.

Semi-private Marking Systems. These systems use the original watermark only and check whether it exists in the cover or not.

Public Marking Systems (Blind Marking). These systems remain the most challenging since they require neither the secret original nor the embedded watermark . Blind watermarking techniques are less robust and are therefore more suitable for applications requiring lower security than copyright application, such as authorized copy

distribution in electronic commerce.

Transformation

Another classification criterion distinguishes schemes into spatial domain techniques and transform-domain techniques depending on whether the watermark is encoded by directly modifying pixels (such as simply flipping low-order bits of selected pixels) or by altering some frequency coefficients obtained by transforming the image into the frequency domain. Spatial domain techniques are simple to implement and often require a lower computational cost, although they can be less robust against tampering than methods which place the watermark in the transform domain. Watermarking schemes that operate in a transform space are increasingly common, as this can aid robustness against several attacks and distortions (Transform domain method hide messages in significant area of the cover image which makes them more robust to attacks). However, while they are more robust to various kind of signal processing, they remain imperceptible to the human sensory system. Most schemes operate directly on the components of some transform of the cover like discrete cosine transform, discrete wavelet transforms and discrete Fourier transforms.

2.3.4 Visibility

Copyright marks do not always need to be hidden, as some systems use visible digital watermarks [56], but most of the literature has focused on invisible (or transparent) digital watermarks which have wider applications. Modern visible watermarks may be visual patterns (e.g. a company logo or copyright sign) overlaid on digital images.

2.3.5 Robustness

Fragile Watermarks. Watermarks that have very limited robustness and are destroyed as soon as the object is modified too much. They are applied to detect modifications of the watermarked data, rather than conveying unreadable information [53]. Cryptographic techniques have already made their mark on authentication. However, there are two significant benefits that arise from using watermarking: First, the signature becomes embedded in the message. Second, it is possible to create 'soft authentication' algorithms that offer a multi-valued measure that accounts for different unintentional transformations that the data may have suffered instead of the classical yes/no answer given by cryptography-based authentication.

Robust Watermarks. have the property that it is not feasible to remove them or make them useless without destroying the object at the same time. This usually means that the mark should be embedded in the most robust significant components of the object [59].

Naturalness

Watermarks that range from pseudo-random sequences to small image logo that can be easily recovered and authenticated.

2.3.6 Properties

Watermark system can be characterized by a number of defining properties [53],[59],[56]. The relative importance of each property is dependent on the requirements of the application and the role that the watermark will play. Some important properties are listed below:

Fidelity (watermark imperceptibility) Perceptual similarity between the original and the watermarked versions must be very high (i.e. the difference between the original image and the embedded watermarked work should be invisible). It has been argued that the watermark should not be noticeable to the viewer instead of being imperceptible [59]. Furthermore, if a signal is truly imperceptible, then perceptually base lossy compression algorithms should, in principle, remove such a signal. Current compression algorithms can still leave room for an imperceptible signal to be inserted. This may not be true of the next generation of compression algorithms. Thus, to survive the next generation of lossy compression algorithms, it is necessary for a watermark to be noticeable to a trained observer.

Statistical invisibility The watermark must be statistically invisible to thwart unauthorized removal (i.e. a statistical analysis should not produce any advantage from the attacking point of view). The noise-like watermark is statistically invisible and has good auto-correlation properties.

Readily extracted If the decoder needs to run in real-time, then it is necessary for the decoding process to be significantly simpler than the encoding process [59]. In some applications, this requirement is reversed depending on the purpose of watermarking system.

Data payload This refer to the amount of information that can be carried in a watermarked cover and raises capacity issues in digital watermarking. The length of the watermark

serves as a measure of the capacity. A longer watermark signal means that more coefficients need to be modified; hence, the watermarked images look 'noisier'. The more information one wants to embed, the lower the watermark robustness.

Embedding Effectiveness The probability that the embedder will successfully embed a watermark in a randomly selected work. This property is related to real-time embedding system (which must be high).

False Positive Rate The frequency with which we should expect a watermark to be detected in a non-watermarked object (which must be low).

Robustness (Security) The watermark should be resilient to standard manipulations which are both intentional and unintentional in nature. Some authors [53] distinguish between resistance to intentional and unintentional attacks. They use 'security' when dealing with the ability of the watermark to resist hostile attacks while using the 'robustness' when dealing with the ability of the watermark to survive normal processing of the content such as spatial filtering, lossy compression, printing and scanning, geometric distortions (such as rotation, translation and scaling). Note, that robustness actually comprises two separate issues: (i) whether or not the watermark is still present in the data after distortion; (ii) whether the watermark detector can detect it. For example, watermarks inserted by many algorithms remain in the data after geometric distortions but the corresponding detection algorithm can only detect the watermark if the distortion is first removed, otherwise the detector can not detect the watermark [59]. In general, any increase in robustness comes at the expense of increased watermark visibility. Also the presence of the original cover increases the robustness. For example, the use of the original image permits some pre-processing to be carried out before the watermark checking such as: rotation angles, translation and scaling factors which can usually be estimated together with missing parts of the image which can be replaced by corresponding parts of the original one. It is possible to undertake and exhaustive search on different rotation angles and scaling factors until a watermark is found, but this is prohibitively computationally intensive.

2.4 Distortions and Attack

In practice, a watermarked cover may be altered either intentionally or unintentionally, so the watermarking system should still be able to detect and extract the watermark. The distortions are limited to those that do not produce excessive degradations, since otherwise

the transformed object would be unusable. Several authors have classified attacks based on a range of aspects. One famous classification has been carried out by Craver et al. [60],[61].

2.4.1 Attack Classifications

There are four general classes of attacks [60], organized by the way in which an attack attempts to defeat the watermarking technology. These classes are illustrated below together with some examples for each class. Some of them may be intentional or unintentional, depending on the application.

Robustness (Unauthorized removal)

This type of attacks aim to diminish or remove the presence of a digital watermark from its associated content, while preserving the content so that it is not useless after the attack is over. Some examples of robustness attacks are discussed below.

Additive Noise. This may happen (unintentionally) in certain applications such as D/A (printing) and A/D (scanning) converters or from transmission errors. It could happen intentionally by an attacker who is trying to destroy the watermark (or make it undetectable) by adding noise to the watermarked cover.

Filtering. Linear filtering such as low-pass filtering or non-linear filtering such as median filtering.

Collusion attack. In some watermarking schemes, if an image has been watermarked many times under different secret keys, it is possible to collect many such copies and 'average' them into a composite image that closely resembles the original image and does not contain any useful watermarking data [62].

Inversion attack (elimination attack). An attacker may try to estimate the watermark and then remove the watermark by subtracting the estimate or reverse the insertion process to perfectly remove the watermark. This means that an attacked object can not be considered to contain a watermark at all (even using a more sophisticated detector). Note, that with different watermarked objects, it is possible to improve the estimate of the watermark by simple averaging.

Lossy Compression. This is generally an unintentional attack, which appears very often in multimedia applications. Practically, nearly all audio, video and digital images that are currently distributed via the Internet are in compressed form. Lossy image

compression algorithms are designed to disregard redundant perceptually-insignificant information in the coding process. Watermarking tries to add invisible information to the image. An optimal image coder would therefore simply remove any embedded watermark information. However, even state-of-the-art image coding such as JPEG 2000 does not achieve optimal coding performance and therefore there is a 'distortion gap' that can be exploited for watermarking. Actually, one can observe that the use of a particular transform provides good results against compression algorithms based on the same transform. For instance, DCT-domain image watermarking is more robust to JPEG compression than spatial-domain watermarking.

Presentation (Masking Attacks)

This attack does not attempt to remove the watermark, but instead, alters the content so that the watermark can no longer be detected or extracted easily. This means that the attacked work can still be considered to contain the watermark, but the watermark is undetectable by an existing detector (such as a detector sensitive to image rotation). Examples of presentation attacka include:

Chopping attack (mosaic attack). Here, an image is 'chopped' into distinct sub-images, which are embedded one after another in a web page. Common web browsers render sub-images together as a single image, so the result is identical to the original image. However, the chopping process distributes the original image's watermark into many pieces and the watermark cannot be recovered unless the original image is reconstructed first.

Rotation and Spatial Scaling. Detection and extraction fail when rotation or scaling is performed on the watermarked image because the embedded watermark and the locally generated version do not share the same spatial pattern anymore. This kind of attack can be unintentional, occurring during the scanning-printing process (copies from printing and/or scanning maybe rotated, scaled, cropped or translated in comparison with the original image).

Cropping. This is a very common attack since in many cases the attacker is interested in a small portion of the watermarked object, such as parts of a certain picture or frames of a video sequence. With this in mind, in order to survive this kind of attack, the watermark needs to be spread over the whole document.

2.4.2 Interpretation

This kind of attack seeks to forge invalid or multiple interpretations from watermark evidence [60] whereby an attacker can devise a situation, which prevents assertion of ownership. Some example are:

- Multiple watermarking: An attacker may watermark an already watermarked object (creating uncertainty about which watermark was inserted first) and later make claims of ownership. The easiest solution is to time-stamp the hidden information by a certification authority.
- Unauthorized embedding (Forgery). Embedding an illegitimate watermark into works that should not contain them or using watermark inversion to remove the original watermark before inserting a new watermark.

Legality

In a legal attack, the attacker uses a legal precedent, the identity or reputation of the object owner, or some other non-technical information to establish doubt in court as to whether a watermark actually constitutes the proof that its owner claims.

Cox Classification for Attacks

Cox et al. [53] classify the attacks into two main categories: active and passive. Active attacks (e.g. changing the cover) include:

- Unauthorized removal (robustness attack).
- Unauthorized embedding (forgery).

Passive attacks (e.g. not changing the cover) include unauthorized detection which can be in three levels according to severity:

- the adversary detects and deciphers an embedded message;
- the adversary detects the watermark and distinguishes one mark from another, but can not decipher what the marks mean;
- the adversary detects the watermark but without distinguishing the decipher.

There are situations in which the watermark has no hostile enemies and need not to be secure, e.g. when the watermark is used to provide enhanced functionality.

2.5 Watermarking and Cryptogarphy

Watermarking is distinguished from other techniques such as placing the mark in the media header, encoding it in a visible bar, or speaking it out loud as an introduction to an audio clip in three ways:

- a watermark is imperceptible;
- a watermark is inseparable from the work (once the digital images is printed on paper, all data in the header is left behind and further, this data may not survive a change in the image format);
- a watermark undergoes same transformations on the work, which can help in authentication and detection based on the kind of alteration that the image has undergone.

Cryptography

Cryptography can be defined as the study of secret writing, i.e. concealing the contents of a secret message by transforming the original message into a form that cannot be easily interpreted by an observer. Thus, the mere discovery of encrypted data suggests that something is illicit, or at least secret, is occurring. Cryptographic techniques can hide a message from plain view during communication, and can also provide auxiliary information that effectively proves the messages. However, traditional cryptosystems suffer from one important drawback, which renders them useless for the purpose of enforcing copyright law [61]: they do not permanently associate cryptographic information with work. Thus, cryptography alone cannot make any guarantees about the redistribution or alteration of content after it has initially passed through the cryptosystem (i.e. cryptography can not help the seller monitor how a legitimate customer handles the content after decryption). Watermarking can fulfil this need; it places information within the content where it is never removed during normal usage. Further, steganography has a distinct advantage over cryptography; it allows for the communication of secret information without alerting an attacker to the presence of the secrets.

Fidelity

It is currently widely accepted that robust/high fidelity watermarking techniques should largely exploit the characteristics of the HVS and human auditory system (HAS) for more effectively hiding watermarks. Perceptual masking techniques exploit the perceptual masking properties of the human auditory system and of the human vision system [75]. A good

watermarking schema has to adapt to the particular image being watermarked in order to exploit specific HVS characteristics and hence amplifying the watermark where the alterations are least likely to be noticed. Local image characteristics that can help determine the visibility of a watermark are listed below:

Texture. It is usually true to say, that the human eyes are not sensitive to the small changes in texture but are very sensitive to small changes in the smooth areas of an image. Hence, it should be possible to incorporate more information into those parts of the image that contain more textures than smooth area. Related methods accomplish this by calculating a value of local contrast and mapping increasing contrast values to increasing watermark magnitudes [64].

Edges. Edge information of an image is the most important factor for perception of the image. This can present a problem though, as directional edges separating two distinct objects in an image may be identified as high contrast areas. This results in the application of a higher strength watermark signal around the connected edges which causes objectionable watermark ringing on connected edges. Methods have also been proposed which identify areas of true high contrast texture while protecting connected directional edges [64],[132] (regions that contain a sudden transition in luminance).

Brightness/Contrast. When the mean value of the square of the noise is the same as the background, the noise tends to be most visible against mid-gray backgrounds. The mid-gray regions have lower noise-capacity compared to other regions[132].

Robustness

The key to making the watermark robust and to preventing the watermark from being easily attacked is to embed the watermark in the perceptually significant regions of the image. These regions do not change much after several signal processing or compression operations. Moreover, if these regions lose their fidelity significantly, the reconstructed image could be perceptually different from the original one (i.e. visual fidelity is only preserved if the perceptually significant regions remain intact). Also, lossy image compression algorithms are designed to disregard redundant information. Information bits placed within textured areas of the image are therefore more vulnerable to attack. The question, therefore, is how much extra watermark information can we add to the perceptually significant regions without any impact on the visual fidelity? There is a compromise to be reached between hiding a large number of information bits where they can least be seen, but where they can be attacked

by image compression algorithms, or placing fewer bits on less textured but safer portions of the image.

Capacity

Every pixel value of an image can be altered only to a certain limit without making a perceptible difference to the image quality. This limit can be called as the 'just noticeable distortion' or JND level [66],[64],[132]. For instance, smooth areas are assigned relatively low JND compared to strongly textured regions (i.e. strongly textured regions have a very high capacity for noise).

Shaping

There is an advantage to shaping the watermark spectrum based on the cover to match currently known human visual systems. Inserting a watermark that is a function of the cover leads to a non-linear embedding procedure. Such a procedure has the advantage that when the image energy in a particular area is small, the watermark energy is also reduced, thereby avoiding artifacts, and when the image energy is large, the watermark energy is increased, thereby improving the robustness of the procedure. Conversely, if simple linear addition to the watermark and image occurs, then the energy of the watermark must be very low in order to avoid worst case scenarios in which the image energy in a particular place is very low and artifacts are created because the watermark energy was too strong relative to the image[59].

Spread Spectrum

Spread-spectrum techniques spread a narrow band signal (watermark) over a much wider band (cover) such that the signal-to-noise ratio in any single band is very low. However, with precise knowledge of the spreading function, the receiver is able to extract the transmitted signal, summing up the signals in each of the bands such that the detector signal-to-noise ratio is strong. Spread spectrum techniques are useful because they have a low probability of interception by an attacker [75]. Embedding a watermark in the high frequency spectrum yield low robustness whereas embedding the watermark in the low frequency spectrum yields visible impacts. Spread spectrum techniques can reconcile these conflicting points by allowing a low-energy signal to be embedded in each of the frequency bands (both high and low).

2.6 Open Problems

Even though watermarking is a fast growing field there are a number of outstanding problems such as the following [67]:

- Optimization between robustness and visibility limiting the capacity.
- Detection speed which is crucial especially in a real time applications.
- Reading the watermark after geometric distortion.
- Adaptability to different printing process, paper and ink which may degrade the watermark. Moreover, printed images do not maintain their quality over time. They are subject to aging, soiling, crumbling, tearing, and deterioration. Designing a watermark scheme to compensate for these kinds of unintentional attacks is another challenge.
- Different input devices (scanner, cameras) introduce different types of distortions. Accounting for these differences in detection is also a major challenge.

Chapter 3

Application of Wavelets to Watermarking

3.1 Introduction

We discuss a new method for 'watermarking' digital signals using wavelets. In particular, we introduce a technique that utilizes Morlet wavelets (with zero characteristic frequency) known as 'chirp coding'. The reason for considering this method is related to 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 use of frequency modulation is the basis for the method presented here and, in the following sections, the theoretical background to the Matched Filter is considered and the result of using chirp functions with matched filtering re-visited. We apply this method in a novel manner by allowing the authentication scheme to be developed in which the watermark is generated from the field in which it is to be hidden. This function is unique in terms of its properties for reconstructing information (via application of the Matched Filter). The approach considered here allows a code to be generated directly from the input signal, the same code being used to watermark the signal. The code used to watermark the signal is therefore self-generating. 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 since the chirp generated is of very low frequency and amplitude. The watermark is 'broken' as soon as any tampering is undertaken on the (watermarked) signal.

It is often desirable to have a watermark extraction process that is 'blind', implying that the original signal is not required to extract the watermark to increase security. We

solve this problem by using a signal dependent watermark sequence. However, to facilitate a blind extraction process, watermark extraction should be transparent to the watermarking scheme (implying that the sequence extracted from the original and watermarked signals are the same). Thus, in this chapter we consider a new multi-level robust audio watermarking scheme based on embedding 'chirps' (i.e. a stream of frequency modulated signals). The method exploits a unique property of a chirp which is that it can be recovered in very noisy environments. The watermarking sequence is derived from the signal spectrum by dividing it into sub-bands using wavelet decomposition.

3.2 Theoretical Concepts

In cryptography and stenography (the process of hiding secret information in data [72]) 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) [73].

Given the equation

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

then being able to recover f from s given n (and the operator $\hat{P}$) provides a way of authenticating the signal. If, in addition, it is possible to determine that a copy and/or modification 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 s; (ii) its fidelity [134], [75].

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 field n to be known *a priori*. If this field has been generated by a pseudo random or a pseudo chaotic number generator, then the initial condition(s) used to generate this field must be known *a priori* in order to recover the data f. In this case, the seed represents the private key required to recover f. However, in principle, n can be any field that is considered appropriate for confusing the information $\hat{P}f$ including a pre-selected signal or image. Further, if the process of confusion is undertaken in which the signal-to-noise ratio is set to be very low (i.e. $\|n\| >> \|\hat{P}f\|$), then the watermark f can be hidden covertly in the data n provided the inverse process $\hat{P}^{-1}$ is well defined and computationally stable. In this case, it is clear that the host signal or image

n must be known in order to recover the watermark f leading to a private watermarking scheme in which the field n in its entirity is a key. This field can of course be (lossless) compressed and encrypted as required. 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 field n is unknown and to consider the problem of extracting the watermark f in the absence of knowledge of this field. In this case, the reconstruction is based on the result

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

where

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

Now, if a process $\hat{P}$ is available in which $\|\hat{P}^{-1}s\| >> \|m\|$, then an approximate (noisy) reconstruction of f can be obtained in which the noise m is determined by the original signal-to-noise ratio of the data s and hence, the level of covertness of the diffused watermark $\hat{P}f$. In this case, it may be possible to post-process the reconstruction (de-noising for example) 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 in this case, only requires knowledge of the operator $\hat{P}$ (and its inverse) and post-processing options as required. The problem here is to find an operator that is able to recover the watermark effectively in the presence of the field n. Ideally, we require an operator $\hat{P}$ with properties such that $\hat{P}^{-1}n \rightarrow 0$.

In this chapter, we consider the case where the operator $\hat{P} = \delta(t)\otimes$ so that $s(t) = f(t) + n(t)$ and where f is contructed from a sequence of Morlet wavelet or chirp function, specifically, a linear Frequency Modulated (FM) chirp of the (complex) types $\exp(\pm i\alpha t^2)$ where α is the chirp parameter and t is the independent variable. 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 in the presence of the field n that is accurate and robust with very low signal-to-noise ratios. Further, we consider a watermark based on a coding scheme in which the field 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 field 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 $s = f + n$ when $\|f\| << \|n\|$. This is an example of a self-generated coding scheme which avoids the use, distribution and application of reference codes. There are numerous applications of this technique in areas where authentication is mandatory. For example, the method can readily be applied to audio data with no detectable differences in the audio quality of the data, providing a tamper proof facility that can be used to validate the autheticity of the data.

3.3 Wavelets

We now discusses an approach to 'watermarking' digital signals using wavelets. In particular, we introduce a technique that utilises Morlet wavelets (with zero characteristic frequency). These wavelets modulate the frequency linearly with time and are consequenctly known as linear frequency modulated wavelets or 'chirps'. The method introduced is therefore known as '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 high signal-to-chirp ratios. 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. The method considered is of value for embedding information in digital signals and for authenticating the data. The approach is generic in the sense that it can can be used for a range of data types and applications (the authentication of speech and audio signals for example).

Fresnel diffusion which is based on the principles associated with Fresnel optics in which the object function f is mapped to an image function u at a distance z via the equation

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

where the Point Spread Function (PSF) p is given by (ignoring scaling)

$$p(x, y) = \exp\left[-i\alpha(x^2 + y^2)\right]$$

and

$$\alpha = \frac{\pi}{z\lambda}$$

Here, λ is the wavelength. By correlating u with p^* we obtain an estimate $\hat{f}$ for f given by

$$\hat{f}(x, y) \simeq XY \operatorname{sinc}(\alpha X x) \operatorname{sinc}(\alpha Y y) \otimes \otimes f(x, y).$$

which in Fourier space becomes

$$\hat{F}(k_x, k_y) = \begin{cases} \frac{\pi^2}{\alpha^2} F(k_x, k_y), & \mid k_x \mid \leq \frac{\pi X}{z\lambda}, \ \mid k_y \mid \leq \frac{\pi Y}{z\lambda}; \\ 0, & \text{otherwise.} \end{cases}$$

Observe that for a fixed values of X, Y and z, as λ decreases, the bandwidth of $\hat{f}(x, y)$ increases. In other words the resolution on the structure of f, as determined by $\hat{f}$, increases as the wavelength decreases. This is of course consistent with the basic expression for resolution R in which

$$R \sim \frac{D}{\lambda}$$

where D is the size of the aperture. The point here, is that in optics, the wavelength governs tha resolution available and in Fresnel optics this is a consequence of the fact that the convolution kernel is given by $\exp(i\pi R^2/z\lambda)$, $R = \sqrt{x^2 + y^2}$. Note that for the one-dimensional case, we can write this result in the form

$$u(x) = \int w\left(\frac{x - y}{L}\right) f(y) dy$$

where

$$w(x) = \exp(i\pi x^2)$$

and

$$L = z\lambda.$$

In principal, by changing the wavelength of a Fresnel imaging system, we can change the resolution of the system and thus explore the structure of the two-dimensional object function $f(x, y)$ on a multi-resolutional basis. In a practical Fresnel imaging system, for example, it is not usually possible to adjust the wavelength in an arbitrary way. However, for signal processing, we can choose to analyse a function $f(t)$ by convolving it with the function $w(t)$ over a range of scales L.

The function $w(t)$ is an example of a wavelet known as the Morlet wavelet which, as a function of time t, is given by (without the correction term and for unit amplitude)

$$w(t) = \exp[i(\omega_0 t - t^2/2)]$$

where ω_0 is the characteristic frequency of the wavelet. However, w is only one of a number of arbitrary functions that may be considered. Thus, from a generic point of view, we are at liberty to generalise and define the wavelet transformation of a signal $f(t)$ as

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

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

$$w_L(t,\tau) = \frac{1}{\sqrt{\mid L \mid}} w\left(\frac{\tau - t}{L}\right), \quad L \neq 0.$$

The parameters L and τ are continuous dilation and translation parameters respectively.

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 where the convolution integral is now a function of L. In general, a multi-resolution signal analysis is a framework for analysing signals based on isolating variations in the signal 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.

The multi-resolution properties of the wavelet transform have been crucial to their development and success in the analysis and processing of signal and images. Wavelet transformations play a central role in the study of self-similar or fractal signals and images. 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 such as stationary and periodic signals. All 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 either in one- or many-dimensions (e.g. [68], [69] and [70]).

A necessary and sufficient condition for this 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.

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

Provided w has reasonable decay at infinity and smoothness, as is usually the case in practice, the admissibility condition above is equivalent to the condition

$$\int w(t) dt = 0$$

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

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

There are a wide variety of wavelets available [i.e. functional forms for $w_L(\tau)$] which are useful for processing 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. Note that a random fractal signal in which

$$u(t) = \frac{1}{t^{1-q/2}} \otimes n(t)$$

is, in effect, a wavelet transform for $L = 1$ where

$$w_q = \frac{1}{t^{1-q/2}}$$

Fractal modulation involves the generation of fractal noise for different values of q that can be used to encode information.

3.4 Matched Filtering using Chirps

The Matched Filter (e.g. [76], [77] and [78]]) is one of the most common filters used for pattern recognition. It is based on correlating the data with a matching template of the feature that is assumed to be present in the date [77]]. If the feature does indeed exist, then the output of the filter 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 linear FM wavelets the result has some special and important properties which provide an output that is uniquely robust in the presence of high noise levels, i.e. high degrees of confusion. It is this property that forms the basis for a variety of active information systems such as those used in Real and Synthetic Aperture Radar (e.g. [79], [80] and [81]), active sonar and some forms of seismic prospecting, for example. A chirped signal is used in pulse compression Real and Synthetic Aperture Radar applications because of its ability to give low side-lobes when correlated with itself. This gives an advantage of being detected in the presence of high background noise. In a watermarking application, this background noise becomes the signal to be watermarked. Interestingly, some mammals (dolphins, whales and bats, for example) use frequency modulation for communication and 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 communications and information systems 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. The use of frequency modulation is the basis the method presented here and, in the following section, the theoretical background to the Matched Filter is considered and the result of using chirp functions with matched filtering re-visited.

3.4.1 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 f is the Impulse Response Function (IRF), p is the Instrument Function, n is the noise (which is typically taken to have stationary statistics) and $\otimes$ is the convolution operation, i.e.

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

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

$$r = \frac{| \int Q(\omega) P(\omega) d\omega |^2}{\int | N(\omega) |^2 | Q(\omega) |^2 \, d\omega} \tag{7.2}$$

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 $(-\infty, \infty)$. 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' uniform power spectrum, the filter Q which maximizes the SNR defined by r can be shown to be given by the simple result

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

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 IRF p.

3.4.2 Derivation of the Matched Filter

With the problem specified as given in the previous section, the matched filter is essentially a 'by-product' of the 'Schwarz inequality', i.e. the result

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

If we write

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

then the above inequality becomes

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

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

From this result, using the definition of r given in equation (7.2), we see that

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

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

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

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

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

3.4.3 'White Noise' Condition

The noise term $n(t)$ can, in principle, have arbitrary characteristics although they are usually assumed to be stationary. The noise is characterised by two fundamental characteristics: (i) the Probability Distribution Function (PDF) or the Characteristic Function (i.e. the Fourier transform of the PDF); (ii) the Power Spectral Distribution Function (PSDF). To apply the Matched Filter, the function $\mid N(\omega) \mid^2$ 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 set to zero so that the output of the system is 'noise driven'. In general however, it is often necessary to apply a suitable model for the PSDF. Such models may include random fractal noise, for example. 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

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

3.4.4 Deconvolution of Linear FM Chirps

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

$$p(t) = \exp(i\alpha t^2), \quad \mid t \mid \leq T$$

where α is a constant 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.

Let us first consider the case when the additive noise term is neglected and consider a signal given by

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

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

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

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

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

3.4.5 Approximation for Long Chirps

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 often 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)

$$\hat{F}(\omega) = \begin{cases} F(\omega), & \mid \omega \mid \leq \alpha T; \\ 0, & \mid \omega \mid > T. \end{cases} \tag{2.2}$$

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 or correlate with 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 use of chirps for diffusing an input f allows for a high degree confusion using additive noise with relative low SNRs

An example of a matched filter reconstruction is given in Figure 3.1. Here, two unit 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 has been generated using an SNR of 1 where the SNR is defined by

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

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

3.5 Chirp Code Watermarking

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 a 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) the watermark should be sensitive (and ideally ultra-sensitive) to any degradation in the field $s(t)$. 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 a key for the purpose of authenticating the data. However, this requires the distribution of such keys (public and/or private). Instead, we consider the case

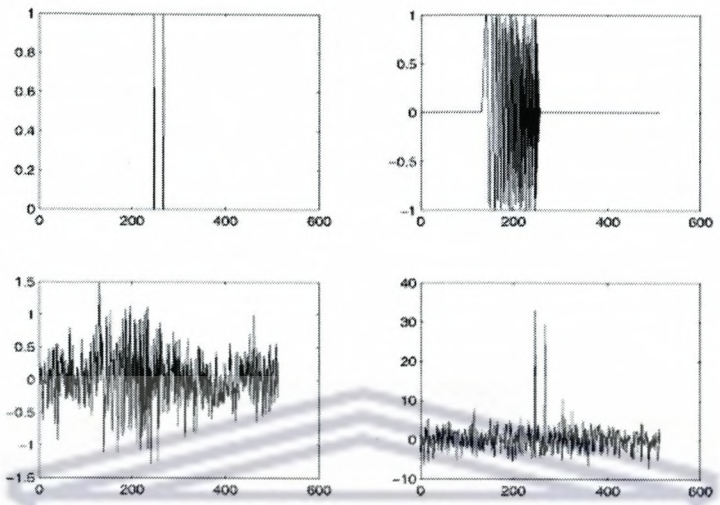


Figure 3.1: Example of a reconstruction using the matched filter (bottom right) from a noisy signal (bottom left) generated by the convolution of an input consisting of two spikes (top left) with a linear FM chirp (top right).

where a binary sequence is generated from the field $n(t)$. There are a number of approaches that can be considered based on the spectral characteristics of $n(t)$, for example, or through application of wavelet decomposition which is discussed later.

3.5.1 Chirp Coding

Given that a binary sequence has been generated from $n(t)$ through some process, 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 $\sin(\alpha t^2)$, $t \in T$ and for 0 we apply the chirp -$\sin(\alpha t^2)$, $t \in T$ 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 or a 'chirp stream'. 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 sweep instead of a linear sweep.

3.5.2 Decoding

Decoding or reconstruction of the binary sequence requires the application of a correlator using the function chirp(t), $t \in [0, T)$. This produces a correlation function that is either -1 or +1 depending upon whether $-$chirp(t) or +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.

3.5.3 Watermarking

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

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

where the coefficients $a > 0$ and $0 < b < 1$ determine the amplitude of the signal s and the chirp stream-to-signal ratio where

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

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

3.6 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 $n(t)$. In this section, the details of this method are presented. There are a wide variety of coding methods that can be applied [82]. 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 in a unique and complete way. 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 [83].

3.6.1 Power Spectrum Decomposition

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

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

Here, we consider a binary sequence that is taken to 'encode' the spectral characteristics of the signal. 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. Hence, we consider the case where the power spectrum is decomposed into N components as follows:

$$P_1(\omega) = P(\omega), \quad \omega \in [0, \Omega_1)$$

$$P_2(\omega) = P(\omega), \quad \omega \in [\Omega_1, \Omega_2)$$

$$\vdots$$

$$P_N(\omega) = P(\omega), \quad \omega \in [\Omega_{N-1}, \Omega_N)$$

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 can consider the energy values in terms of their contribution to the spectrum as a percentage, i.e.

$$E_1 = \frac{100}{E} \int_0^{\Omega_1} P_1(\omega) d\omega$$

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

$$\vdots$$

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

where

$$E = \int_0^{\Omega_N} P(\omega) d\omega.$$

Code generation is then based on the following steps:

(i) Rounding to the nearest integer the (floating point) values of E_i to decimal integer form:

$$e_i = \text{round}(E_i), \quad \forall i$$

(ii) Decimal integer to binary string conversion:

$$b_i = \text{binary}(e_i)$$

(iii) Concatenation of the binary string array b_i to a binary sequence:

$$f_j = \text{cat}(b_i)$$

The watermark f_j is then generated by chirp coding.

3.6.2 Wavelet Decomposition

The wavelet transform was discussed earlier in this chapter and is defined by

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

where

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

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 (which is now a function of L) that gives it the ability to analyse signals in a multi-resolution role. It is this property that is the basis for considering the following approach.

We consider a code generating method that is based on computing the energies of the wavelet transformation over N levels. Thus, the signal $n(t)$ is decomposed into wavelet space to yield the following set of functions:

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

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 spectrum decomposition. Whether applying the power spectrum decomposition method or wavelet decomposition technique, the computations are undertaken in digital form using a DFT and a DWT (Discrete Wavelet Transform) respectively.

3.7 Coding and Decoding Processes

The *coding process* 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 process* 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 process also provides an error measure based on the result

$$e = \frac{\sum_i | x_i - y_i |}{\sum_i | x_i + y_i |}$$

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 which is equivalent to applying and off-set to the data. The segment can be user defined and if required, form the basis for a (private) key system. Further any wavelet can be used for this process and the actual wavelet used yields another feature that can form part of the private key required to extract the watermark.

Coding Process

The coding process is compounded in the following basic steps:

Step 1: Read a .wav file.

Step 2: Extract a section of a single vector of the data (note that a .wav contains stereo data, i.e. two vectors).

Step 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.

Step 4: Compute the (percentage) 'energy values'.

Step 5: Round to the nearest integer and convert to binary form.

Step 6: Concatenate both the decimal and binary integer arrays.

Step 7: Chirp code the binary sequence.

Step 8: Scale the output and add to the original input signal.

Step 9: Re-scale the watermarked signal.

Step 10: Write to a file.

Decoding process

The decoding process is as follows:

Step 1: Steps 1-6 in the coding processes are repeated.

Step 2: Correlate the data with a chirp identical to that used for chirp coding.

Step 3: Extract the binary sequence.

Step 4: Convert from binary to decimal.

Step 5: Display the original and reconstructed decimal sequence.

Step 6: Display the error.

The method of digital watermarking compounded in coding/decoding algorithms given 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 approach considered here allows a code to be generated directly from the input signal and that same code is used to watermark the signal. The code used to watermark the signal is therefore self-generating. Reconstruction of the code only requires a correlation process with the watermarked signal to be undertaken. This means that the signal can be authenticated without access to an external reference code. In other words, the method can be seen as a way of authenticating data by extracting a code (the watermark) within a code (the signal) and is consistent with approaches that attempt to reconstruct information without the host data [84].

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. Figure 3.2 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 subband 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. Figure 3.3 shows the power spectral density of the original, watermarked and an attacked audio signal. The band pass filtering attack is such that there is negligible change in the power spectral density; however, the tampering is easily detected by the proposed technique. Further, chirp based watermarks are difficult to remove from the 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.

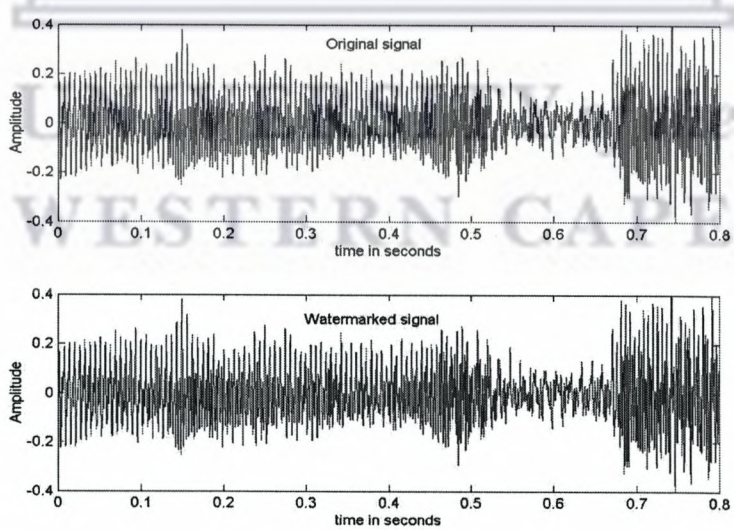


Figure 3.2: Original signal (above) and chirp based watermarked signal for tamper proofing (below)

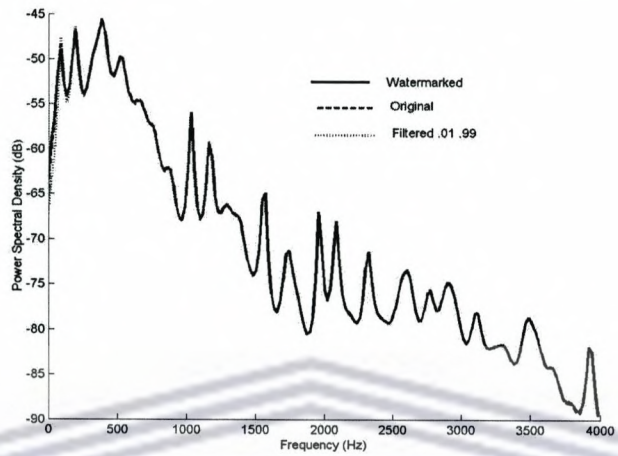


Figure 3.3: Difference in the power spectral density of the original, watermarked and tampered signal. The tampering attack is done by using a band pass filter with normalised lower cut-off frequency of 0.01 and higher cut-off frequency 0.99

3.8 Application to Audio Data Authentication

With the increase in computing power and high bandwidth internet connectivity, copying, editing and illegal distribution for audio has become very easy. To overcome this problem, digital audio watermarking has been proposed for applications such as copyright, annotation, authentication, broadcast monitoring, and tamper-proofing [85], [86], [87]. An important goal to be achieved by a watermark is its imperceptibility such that the end user is unaware of its presence. This is especially important when the audio data is music where degradation in quality cannot be tolerated. Most of the developed algorithms take advantage of the Human Auditory System (HAS) limitations to embed a perceptually transparent watermark into a host signal. A wide range of time domain embedding techniques such as alteration of the Least Significant Bit (LSB) [89], echo addition [90], [91], Quantization Index Modulation (QIM) and spread spectrum methods and transform domain techniques such as Fourier, Cepstral [96], Wavelet [97], [98] , [99] etc. have been tried. Most of these methods and the algorithms developed fall in the category of robust watermarking due their high tolerance towards different attacks. However, there are some applications where there is a need for checking the authenticity/originality of audio. Such applications in digital audio authentication can be found in areas such as broadcasting, sound recording of criminal events etc. Thus, watermarking for such applications must be fragile i.e. the watermark should 'break'

as soon as any tampering is undertaken on the (watermarked) signal. It is also desirable to have a watermark extraction process that is 'blind', implying that the original signal is not required to extract the watermark. To achieve this, a fixed watermark sequence can be embedded. However, this decreases the security of the scheme. Also, a unique watermark sequence should be used to increase security. A solution to this problem is to use a signal dependent watermark sequence. However, to facilitate a blind extraction process, watermark extraction should be transparent to the watermarking scheme (implying that the sequence extracted from the original and watermarked signals are the same). A high robustness to attacks along with a high data rate for the embedded watermark cannot be achieved simultaneously [100]. There have been various attempts to increase the payload capacity for robust watermarking [101] including multi-level watermarking in different domains that have recently been proposed [102].

In this section we consider a new multi-level robust audio watermarking scheme based on embedding 'chirp' (i.e. a stream of frequency modulated signals). The method exploits a unique property of a chirp which is that it can be recovered in very noisy environment. The watermarking sequence is derived from the signal spectrum by dividing it into subbands using wavelet decomposition. Four levels of watermark are embedded without any perceptual distortion. Objective measurements using the Perceptual Evaluation of Audio Quality (PEAQ) test are reported using audio files from Speech Quality Assessment Material (SQAM) with an Objective Difference Grade less than -1.0. The embedded watermark is found to be robust to different simulated attacks. Due to two different processes of watermark sequence extraction, self-authentication and tamper-assessment can also be performed using the proposed technique. Further, the HAS limitation of having poor sensitivity to frequencies below 100Hz can be exploited. Along with robustness, the proposed scheme has a capability for blind self-authentication as well. Thus, the scheme provides a multi-level robust watermarking method with tamper detection capability.

A chirp is a signal whose frequency increases or decreases with time and may be linear, quadratic or logarithmic as shown by the spectrograms in Figure 3.4

For example, the instantaneous frequency of a logarithmic chirp signal is given by

$$\omega_i(t) = \omega_0 + 10^{\beta t} \tag{1}$$

where β is expressed as

$$\beta = \frac{1}{t_1} \log_{10}(\omega_1 - \omega_0) \tag{2}$$

Here, ω_0 is the initial frequency and ω_1 is the final frequency at time t_1.

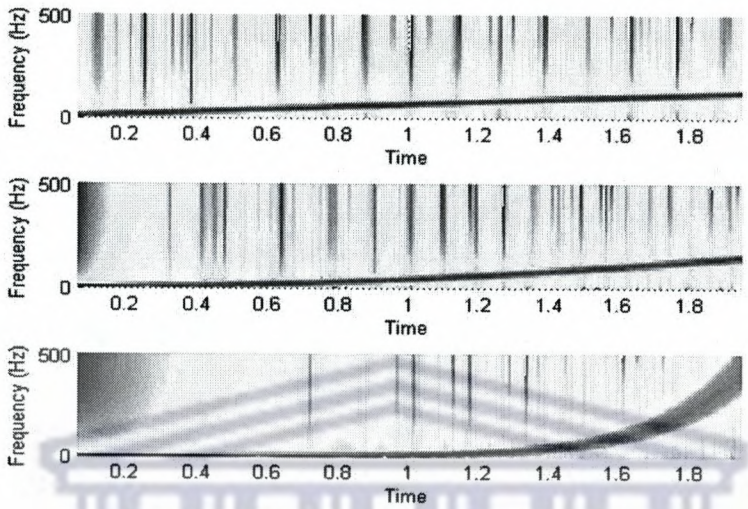


Figure 3.4: Spectrograms of different types of chirp: linear (above), quadratic (centre) and logarithmic (below)

3.8.1 Watermark Generation

To embed the watermark in a signal $n(t)$, we use a chirp $c(t)$ and watermark sequence m based on the following equation

$$y(t) = n(t) + k \cdot c(t) \; if \; m = 1, \, t \in [0, T) \tag{3}$$

$$y(t) = n(t) - k \cdot c(t) \; if \; m = 0, \, t \in [0, T) \tag{4}$$

where $y(t)$is the watermarked signal and k is the scaling factor. The watermark sequence is binary with '+1' or '-1' values which implies that a scaled version of a chirp is added to the original signal in-phase or out-of-phase. This sequence can be generated by using a secret key such that, once the key is known, the entire sequence can be generated. In this paper we propose a technique to generate the watermark sequence using the audio signal itself.

A unique property of an audio signal is its spectral variation. This can be exploited to derive a watermark binary sequence. The entire power spectrum is decomposed into N sub-bands i.e.

$$P_i(\omega) = P(\omega), \qquad \omega \in [\Omega_{i-1}, \Omega_i) \, 1 \leq i \leq N \tag{5}$$

It is important that the signal $n(t)$ is band-limited and has a bandwidth ofΩ_N. The set of functions $P_1, P_2, ..., P_N$ represent the complete spectral characteristics of the signal $n(t)$. Since each of the components represents a unique part of the spectrum, a natural measure is to consider the energy, which is given in terms of the integral of the power spectrum over the frequency range. The energy calculated in each sub-band is represented as a percentage of the total energy of the signal. The reason of calculating percentage energies of the sub-bands is to avoid any influence of signal scaling on the authentication of the signal. The energy in the i^{th} sub-band is given by

$$E_i = \frac{100}{E} \int_{\Omega_{i-1}}^{\Omega_i} P_i(\omega) d\omega \qquad 1 \leq \mathrm{i} \leq \mathrm{N} \tag{6}$$

where

$$E = \int_0^{\Omega_N} P(\omega) d\omega \tag{7}$$

An audio signal is split into sub-bands by applying the wavelet transformation which is defined by [103].

$$W_L(t) = \int f(\tau)\, w_L(t, \tau)\, d\tau \tag{8}$$

where

$$W_L(t, \tau) = \frac{1}{\sqrt{|L|}} w\left(\frac{t-\tau}{L}\right)$$

Concatenating the total energy and sub-band energies gives a watermark vector. All the elements of this vector are converted into binary form in which a '0' is replaced by '-1' thereby giving the watermark binary sequence m. The watermark is then embedded using equation (3). A second level of watermark is applied by choosing a different initial and final frequency with a different watermark sequence. The length of the chirp added can also be different for different levels.

3.8.2 Watermark Recovery

Recovery of the watermark binary sequence requires correlation of the chirp function over the desired interval ([nT, (n + 1)T), n = 0, 1, 2, ...,) with the watermarked signal. This produces a correlation function that is either *+1* or *-1* depending upon whether the embedded chirp is in-phase or out-of-phase. In practice *+1* or *-1* is never obtained and thus, thresholding is applied such that if the value is positive then the bit is assumed to be *1* and *0* otherwise.

Embedding Level	Frequency range (Hz)	SNR (dB)	ODG
1	0-15	30.9050	-0.46556
2	0-30	28.1748	-0.77894
3	0-45	27.3172	-0.88256
4	0-60	26.7349	-0.95156

Table 3.1: SNR and ODG achieved for different level of embedding

The chirp used to recover the watermark must have the same parameters as those used during the embedding watermark sequence. These parameters can be used to define part of a private key.

3.8.3 Results

To verify the proposed scheme, audio files were selected from the Speech Quality Assessment Material (SQAM) which is commonly used for assessing the quality of speech coders. A logarithmic chirp signal with a frequency sweep less than 100Hz was used for embedding the watermark. This frequency range was chosen to keep the watermarks imperceptible. Wavelet decomposition using a 'Daubechies 4' mother wavelet was carried out. The resultant sub-bands obtained for a 48kHz sampled signal are 0-3kHz, 3-6kHz, 6-9kHz, 9-12kHz, 12-15kHz, 15-18kHz and 18-24kHz. A binary watermark sequence was derived by representing percentage sub-band energies by 12 bits with a total energy of 32 bits giving a total of 116 bit information. The coded chirp was scaled and embedded into the signal. The first level of watermark was embedded by using a chirp frequency sweep of 0-15Hz. A total of four embedding levels was investigated with chirp specifications as shown in Table 3.1.

In order to analyze the worst case performance, the same sub-band based watermark sequence was embedded at all four levels. Further, the length of the chirp and the start point for the embedding process was kept the same at all levels giving overlapping chirp frequency ranges over a common time frame and thereby, giving a maximum possibility of error during the detection process. Subjective assessment of the speech quality was carried out by calculating the Signal-to-Noise Ratio (SNR) at each embedding level. The average SNR was evaluated for different levels providing the results shown in Table 1. This SNR can be improved if the length of the chirp is increased. However, it is important to take into consideration the human hearing curve to interpret the SNR achieved. Since the human ear has very poor sensitivity below 100 Hz a lower SNR is still imperceptible upon listening. To further verify the above results, tests based on Perceptual Assessment of Audio

Quality (PEAQ, Basic) [104] were also carried out. The PEAQ algorithm is the ITU-R recommendation (ITU-R BS.1387) for perceptual evaluation of wide-band audio codecs. This algorithm models fundamental properties of the auditory system along with physiological and psychoacoustic effects. It uses both original and test signals, and applies techniques to find differences between them. An Objective Difference Grade (ODG) is evaluated using a total of eleven Model Output Variables (MOV) of the basic version of PEAQ. The original signal and watermarked signal for different embedding levels was used to evaluate ODG with results as given in Table 1. ODG values mimic the listening test ratings and have values -4.0 (very annoying) to 0 (imperceptible difference).

Although all the MOVs were calculated from the PEAQ (basic version) test, only those relevant to watermarking are reported here. The Noise-to-Mask Ratio (NMR) is an estimate in dBs of the ratio between the actual distortion (caused due to the embedding watermark in this case) and the maximum inaudible distortion. The total NMR is the average of the NMRs calculated over all frames. Negative NMR values indicate inaudibility whereas values larger than 0dB indicate audible distortions caused by the watermark. This is an important test for checking the inaudibility of the embedded watermark at different levels. Results for both speech signal and music signals were analysed separately and are plotted in Figure 3 along with the overall average. As stated earlier, the SNR (obtained in Table 1) shows a high level of watermark, but due to its very low frequency, it is not audible.

The noise loudness quantifies the partial loudness of distortions that is introduced during when the watermark is embedded in the host signal. The Root Mean Square (RMS) value of noise loudness has a maximum limit of 14.8197. Figure 4 shows a normailsed plot of the average RMS noise loudness achieved on the SQAM data for all of the 4 levels considered. It is possible for the total NMR to be below 0dB (implying inaudiable), but there may be a large number of frames with small positive values and few frames with large negative value. This distribution can be seen by evaluating the number of disturbed frames. A relatively disturbed frame is one in which the maximum NMR exceeds 1.5 dB and is expresses as a fraction of the total frames. The results in Figure 4 show that with 4 level embedding, less than 3.5 percent of the frames have an NMR above 1.5dB. Thus, the multi-level watermarking proposed is imperceptible. To evaluate the robustness and self-authentication capability, different attacks were simulated simulated as discussed below.

3.8.4 Robustness

To evaluate the robustness of a multi-level watermark, correlation between the watermark sequences (obtained by sub-band energies from the original signal) and the extracted wa-

termark (obtained by correlating the chirp with the watermarked signal) was carried out. Different attacks such as addition of additive white Gaussian noise, up-sampling, down-sampling, re-sampling, low-pass and high-pass filtering were carried out on the multi-level watermarked audio signal. The tests were carried out on all the SQAM files providing 'average results' as described below.

For and attack using additive noise, it was found that watermark embedded using a low frequency sweep was more robust when compared to a high frequency sweep. The average SNR at which the detection error starts for level-1 embedding was found to be -1.9060dB while for level-4, it was 12.14dB. The overall robustness to noise of the scheme was 13.0376dB which clearly shows that a level-4 watermark (sweep from 0-60 Hz) is more sensitive to noise. Thus is, or course, to be expected and illustrates that multi-level watermarking can help in embedding critical and important information using lower frequency sweeps that it more resistant to attack.

The sampling rates were varied from 10% to 200% of the sampling frequency and the watermark recovered at all levels without any error. Thus, the extraction scheme has no effect of changes to the sampling rate. Amplitude scaling also observed to have no effect on watermark recovery provided it was constant over the entire frequency band.

To simulate a filtering attack, the watermarked signal was passed through a finite impulse response high-pass filter of order '50' and cut-off frequency of 'fc'. This cut-off frequency was varied and extraction of watermark was carried out until an error was obtained. It was observed that the error in recovering the watermark occurred at fc=936Hz. Since a filter has smooth transitions from stop-band to pass-band, the embedded chirp streams are not removed but severely attenuated. Since the chirp can be extracted from very noisy environment, it is possible to extract the watermark at high cut-off frequency. However, appreciable degradation of in audio quality occurs using high-pass filters with fc=936Hz. The results obtained are shown in Figure 5. Note that higher cut-off frequencies can be achieved if the order of the filter is reduced. The scheme is resistant to low-pass filter attack since the embedded watermark occupies a very low frequency band. Removing the watermark by a low-pass filter effectively removes the entire signal along with the watermark. To have intelligible audio quality, the bandwidth of the low-pass filter should be at least 4 kHz for which the watermark is fully recoverable without any error.

Chapter 4

Digital Image Watermarking Methods

4.1 Introduction

Like many aspects of digital signal processing, watermarking schemes fall into two categories: spatial domain and transform domain techniques. This depends on whether the watermark is encoded by directly modifying pixels (such as simply flipping low-order bits of selected pixels) or by altering some frequency coefficients obtained by transforming the image in the frequency domain. Spatial domain techniques are simple to implement and often require a lower computational cost, although they can be less robust against tampering than methods that place the watermark in the transform domain. Watermarking schemes that operate in a transform space are increasingly common, as these schema posses a number of desirable features. In this Chapter, these aspects are discussed in depth and implemented. They bring a novel approach to how we have used the embedding techniques based on an overview of the principal techniques associated with digital image watermarking given in the following section.

4.2 Transform Domain Methods

Like many aspects of digital signal processing watermarking schemes fall into two categories: spatial domain and transform domain techniques. This depends on whether the watermark is encoded by directly modifying pixels (such as simply flipping low-order bits of selected pixels) or by altering some frequency coefficients obtained by transforming the image in the frequency domain. Spatial domain techniques are simple to implement and often require a lower computational cost, although they can be less robust against tampering than methods that place the watermark in the transform domain. Watermarking schemes that operate in

a transform space are increasingly common, as these schema posses a number of desirable features. This section[1] provides an overview of the principal techniques associated with digital image watermarking.

Watermarking schemes that operate in a transform space are increasingly common, as these schemes posses a number of desirable features including:

- By transforming spatial data into another domain, statistical independence between pixels as well as high-energy compaction can be obtained.
- The watermark is irregularly distributed over the entire spatial image upon an inverse transformation, which makes it more difficult for attackers to decode and read the mark.
- One can provide markers according to the perceptual significance of different transform domain components, which means that one can adaptively place a watermark where it is least noticeable, such as within a textured area.
- Transform domain methods can hide messages in significant areas of the cover which makes them more robust against several attacks and distortion. However, while they are more robust to various kinds of signal processing, they remain imperceptible to the human sensory system.
- Cropping may be a serious threat to any spatially based watermark but is less likely to affect a frequency-based scheme. Since watermarks applied to the frequency domain will be dispersed over the entirety of the spatial image upon inverse transformation, we can retrieve part of the watermark.
- Lossy compression is an operation that usually eliminates perceptually unimportant components of a signal. Most processing of this sort takes place in the frequency domain. In fact, matching the transform with a compression transform may result in better performance of the data-hiding schema (i.e. DCT for JPEG, Wavelet for JPEG-2000).
- The characteristics of the Human Visual System (HVS) can be fully exploited in the frequency domain.

[1]Based on an edited version of K. W. Mahmoud, Low Resolution Watermarking for Print Security, PhD Thesis, Loughborough University, 2005, (Chapter 2).

4.3 Frequency Domain Processing and HVS

It is usually the case, that the HVS is not sensitive to small changes in edges and texture but they are very sensitive to small changes in the smooth areas of an image. In flat featureless portions of the image, important information concerned with the 'flat parts' concentrate on the lowest frequency components, while, in a highly textured image, energy is concentrated in the high frequency components. Therefore, the human eyes are more sensitive to lower frequency noise, rather than high frequency noise. Taking into account these fundamental points:

- The watermark should be embedded into the higher frequency components to achieve better perceptual invisibility, however, high frequencies might be discarded after attacks such as lossy compression, shrinking or scanning.
- In order to prevent the watermark from being easily attacked, it is often necessary to embed it into the lower frequency coefficients. The attacker can not change these coefficients, otherwise the image can be damaged. However, the human eyes are more sensitive to lower frequency noise.

From the points discussed above, in order to invisibly embed a watermark, which can survive most attacks, it is clear, that a reasonable trade-off is to embed the watermark into the middle frequency range of the image [105].

4.4 Frequency Domain Processing

In watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the watermark. Coefficient selection is based on perceptual significance or energy significance. When the watermarked image is compressed or modified by any image processing operations, noise is added to the already perturbed coefficients. The private retrieval operation subtracts the received coefficients from the original ones to obtain the noise perturbation. The watermark is then estimated from the noisy data as best as possible. The most difficult problem associated with blind watermark detection in the frequency domain is to identify the coefficients used for watermarking. Embedding can be done by adding a pseudo-random noise field, quantization (threshold) or image (logo) fusion. Most algorithms consider HVS to minimize perceptibility. The aim is to place more information bits where they are most robust to attack and are least noticeable. Most schemes operate directly on the components of some transform of the cover such as the Discrete Cosine Transform

(DCT), Discrete Wavelet Transforms (DWT) and Discrete Fourier Transforms (DFT). In this section we introduce each transformation, illustrates its main features and introduce some techniques that use the transformation in watermarking.

4.4.1 Discrete Cosine Transform

The DCT transform has a number of advantages in respect of watermarking:

- The DCT has the primary advantage that it is a sequence of real numbers, provided that the input sequence is real.
- The two-dimensional DCT is the basis for most popular lossy digital image compression system used today, e.g. the JPEG system.
- The sensitivity of HVS to the DCT basis images has been extensively studied resulting in a default JPEG quantization table.

4.5 Embedding Techniques using the DCT

Zhao, et al. [106] approach the problem by segmenting the image into $8x8$ blocks. Block DCT transformation and quantization steps are applied on each block. A bit of information can be encoded in a block using the relation between three quantized DCT coefficients $(c1, c2, and c3)$ from this block. The three coefficients must correspond to the middle frequencies. One block encodes a '1', if $c1 > c3 + d$ and $c2 > c3 + d$. On the other hand, a '0' is encoded, if $c1 + d < c3$ and $c2 + d < c3$. The parameter d accounts for the minimum distance between two coefficients. The higher d is, the more robust the method will be against image processing techniques. If the relations between the coefficients do not correspond to the encoded bit, a change must be made to the coefficients so that they can represent the encoded bit. If the modification required to code one bit of information is too large, then the block is not used and marked as an invalid block. Consequently, the blocks are de-quantized and the inverse DCT is applied. In the decoding step, comparing the three coefficients of every block in the quantized DCT domain can restore the label.

Cox, et al. [107] present an image watermarking method in which the mark (a sequence of real numbers $\{w_i\}$ having a normal distribution with zero mean and a unity variance) is embedded in the n (excluding the DC term) most perceptually significant frequency components $V = \{v_i\}$ of an image's DCT to provide greater robustness to JPEG compression. The watermark is inserted using he procedure $\acute{v}_i = v_i + \alpha * v_i * w_i$. This modulation law

is designed to take into account the frequency masking characteristics of the human visual system. This non-linear insertion procedure adapts the watermark to the energy present in each coefficient. The advantage of this is that when v_i is small, the watermark energy is also small, thereby avoiding artifacts. When v_i is large, the watermark energy is increased for robustness. The parameter α represents a compromise between robustness and image fidelity. The presence of the watermark is verified by extracting the main components of the original image and those with same index from a watermarked image and inverting the embedding formula to give a possibly modified watermark W'. The watermark is said to be present if the correlation between W and W' is greater than a given threshold.

Barni, et al. [108] propose a watermarking algorithm similar to Cox's method. However, instead of using the n largest DCT coefficients as Cox does, the set is produced by arranging the DCT coefficients in a zigzag order and a subset in the mid-frequency range is selected. The lowest coefficients are then skipped to preserve perceptual invisibility. The watermark is then embedded in this set of coefficients in the same way as Cox. In order to enhance the invisibility of the watermark, the spatial masking characteristics of the HVS are also exploited to adapt the watermark to the image being signed: the original image (I) and the watermarked image (I') are added pixel by pixel according to a local weighting factor $b_(i,j)$, thus producing a new watermarked image $\hat{I_{i,j}} = I_{i,j}(1 - b_{i,j}) + b_{i,j} * I'_{i,j}$, in a region characterized by low-noise sensitivity, where the embedding of the watermarking data is easier (e.g. highly textured regions where $b_{i,j} \approx 1$), i.e. in regions more sensitive to change, in which the insertion of the watermark is more disturbing (e.g. uniform regions where $b_{i,j} \approx 0$) the watermark is embedded only to a minor extent. In the extraction phase, one first extracts the subset of modified coefficients from the full frame DCT of the watermarked image. The correlation between the marked (possibly corrupted coefficients) and the mark itself is taken as a measure of the mark presence.

O' Ruanaidh, et al. [109] present a private watermarking technique for images using bi-directional coding in the DCT domain. In bi-directional coding, the image is divided up into blocks. The DCT is computed for each block. The mean of each block is incremented to encode a '1' or decremented to encode a '0'. This may be accomplished by using simple thresholding techniques. A JPEG quantization table (visual masking) is used to weight the DCT coefficients in each block. The most significant components are then selected by comparing the square of the component magnitude to the total energy in the block. In the decoding step, the mean of each block in the original non-watermarked image is compared with the mean of the corresponding blocks in the tested copy to decode the stored bit.

Swanson, et al. [110] embed the watermark by computing the DCT for each block in the cover. A perceptual mask is computed for each block. The resulting perceptual mask is

then scaled and multiplied by the DCT of the pseudo-noise watermark. The schema uses a different sequence for each block. The watermark is then added to the corresponding block. The watermark can be detected by correlating the modified watermark with the original watermark and comparing the result to a threshold.

Chae, et al. [111] use a public technique to embed a signature (watermark) into images. The signature DCT coefficients are quantized according to a signature quantization matrix. The resulting quantized coefficients are encoded using lattice-codes. The choice of the signature quantization matrix affects the quantity and the quality of the embedded data. The codes are inserted into the middle frequency DCT coefficients of the host image. This insertion is adaptive to the local texture content of the host image blocks and is controlled by the block texture factor. The texture factor is computed using a wavelet transform. The selected host coefficients are then replaced by the signature codes and combined with the original unaltered DCT coefficients to form a fused block of DCT coefficients. The fused coefficients are then inverse-transformed to give an embedded image.

Bors, et al. [112] propose watermark algorithm based on imposing constraints in the DCT domain. The block sites for embedding the watermark are selected based on a Gaussian network classifier, then the DCT constraints are embedded in the selected blocks. Two distinct algorithms are considered here. The first algorithm embeds a linear constraint on selected DCT frequency coefficients (i.e. $Y = FQ$, where F is the vector of the modified DCT coefficients, and Q is the weighting vector provided by the watermark). In the second approach, circular regions are defined around certain DCT coefficients. For a selected block site, they evaluate the Euclidean distance between its DCT coefficient vectors and that of the watermark. The chosen DCT coefficients are changed to the value of the closest watermark parameter vector. After modifying the DCT values, the image is reconstructed based on the inverse DCT transform. In the detection stage, a check is made on the DCT constraints after which the respective block is located. A given site is considered as being 'signed' when the probability of detecting the constrained DCT coefficients and the probability of detecting the location constraint is maximized. The original image is not required for watermark detection and simulations have showed this method to be resistant to JPEG compression and filtering.

Kankanhalli, et al. [132] propose a way of analysing the noise sensitivity of every pixel based on the local region content (texture, edges and luminance). If the distortion caused by the watermarking algorithm is at or below a threshold, the degradation in the original image quality is imperceptible. The analysis is based on the DCT coefficients. The energy in the DCT coefficients can be used as a measure of roughness, and the count of large-magnitude fluctuations in a high-energy block can then be used to decide if the block has an edge or is highly textured. The work also analyses the contribution of luminance to noise sensitivity.

This luminance analysis is done at the pixel level in the spatial domain. The authors use the previous mask to embed an invisible watermark in the spatial domain.

B. Tao, et al. [113] have an approach that is similar to that of [132]. Here, the block as a whole is given a sensitivity label that shapes the watermark based on texture and edges analysis. The embedding is then done in the DCT transform domain.

4.6 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is similar to a hierarchical sub-band system, where the sub-bands are logarithmically spaced in frequency and represent octave-band decomposition. The DWT can be implemented using digital filters and down-samplers [114]. The original image is split into four quadrant bands after decomposition. The four quadrants contain approximations to the sub-band (LL), horizontal detail sub-band (LH), vertical detail sub-band (HL) and a diagonal detail sub-band (HH). This process can be repeatedly applied on the approximation sub-band to generate the next coarser scale wavelet coefficients. The process continues until some final scale is reached. The wavelets transform has a number of advantages [115], [116] over other transform that can be exploited for watermarking including the following:

- It is well known that wavelet coding has been exploited in new compression standards such as JPEG2000 and MPEG4 due to the excellent performance in compression.
- The wavelet transform requires a lower computational cost of $O(n)$ compared with the Fourier or the Cosine transform which are of $O(nlogn)$, where n is the length of the signal.
- The wavelet transform processes data at different scales or resolutions, highlighting both large and small features. This allows watermarking to become adaptive as it depends on the local image characteristic at each resolution level.
- The wavelet functions provide good space-frequency localization and thus, they are suited for analyzing images where most of the informative content is represented by components localized in space such as edges and borders.
- With the DWT, the edges and texture are usually exploited very well in the high frequency sub-band (HH, HL and LH). Therefore, adding a watermark via these large coefficients is difficult for the human eye to perceive.

- Wavelet functions have advantages over traditional Fourier methods in analyzing signals containing many discontinuities or sharp changes.
- The wavelet transform is flexible enough to adapt to a given set of images or particular type of application. The decomposition filters (such as Haar, Daubechies-4, 6 or biorthogonal filters) and the decomposition structure (wavelet packet, complex wavelet transform) can be chosen to reflect the characteristics of the image.
- Research into human perception [117] indicates that the retina of the eye splits an image into several frequency channels each spanning a bandwidth of approximately one octave. The signals in these channels are processed independently. Similarly, in a multiresolution decomposition, the image is separated into bands of equal bandwidth on a logarithmic scale. It can therefore expected that use of the discrete wavelet transform will allow the independent processing of the resulting components without significant perceptible interaction between them, and hence makes the process of imperceptible marking more effective. For this reason the wavelet decomposition is commonly used for the fusion of images.

4.7 Embedding Techniques in the DWT Domain

There have and continue to be many attempts to use the wavelet transform in watermarking. These include the following.

Xia, et al. [115] propose a private watermarking system. The method utilizes large DWT coefficients of all sub-bands excluding the approximation image to equally embed a random Gaussian distributed watermark sequence in the whole image. The decoding process is based on a hierarchical correlation of coefficients at different sub-bands. First, they apply a one level DWT to the watermarked image and then on the original image. The difference (corrupted watermark) of the DWT coefficients in the HH band of the watermarked and the original image is then calculated. Then, the cross-correlation between the corrupted watermark and the part of the original watermark that was added in the HH band is determined. If there is a peak in the cross correlation, the watermark is considered to be detected, otherwise they consider the other bands at the same level (i.e. HH and LH, then HH, LH and HL). In case the watermark still cannot be detected, they compute a new level of the DWT and try to detect the watermark again. This process is performed until the watermark is detected or the last level of the DWT has been reached.

Kundur, et al. [117] embed a binary watermark into the detail wavelet coefficients of the host image with the use of a key. This binary randomly generated key is used to select

the exact locations in the wavelet domain (ones location) in which to embed the watermark. First, they compute the L^{th} level discrete wavelet decomposition of the host image to produce a sequence of a L^3 detailed images. Then, for each level, the embedding modulation at any of the selected coefficients is undertaken as follows:

- Order the horizontal, vertical and diagonal detail coefficients at this location (high, middle, and low).
- The range of values between high and low is divided into bins of width $(high - low)/(2Q - 1)$ where Q is a user-defined variable. These bins represent 1 and -1 in a periodic manner.
- To embed a watermark bit of value 1, the middle coefficients are quantized to the nearest 1 bin. Alternatively, to embed a bit of -1, the middle coefficient is quantized to the nearest -1 bin.

Finally, apply the inverse wavelet transform to form the watermarked image.

In Kunder, et al. [118] the host is transformed to the L^{th} level discrete wavelet decomposition. Only the first level discrete wavelet decomposition of the watermark is performed. The watermark is a random binary two-dimensional array. It is required that the size of the watermark in relation to the host image be small. The detail images of the host at each resolution level are segmented into a non-overlapping rectangle. Each rectangle has the same size as any bands of the watermark. A numerical measure of perceptual importance (salience) of each of these localized segments is computed. The watermark is embedded by a simple scaled addition of the watermark to the particular detail component. The scaling of the watermark is a function of the salience of the region. The greater the salience, the stronger the presence of the watermark. Finally the corresponding L^{th} level inverse wavelet transform is performed. Salience is computed based on a well-known model given by Dooly [119], which is based on contrast sensitivity.

The original image is required in the extraction process. The extraction process is done by applying the inverse procedure at each resolution level to obtain an estimate of the watermark. The estimates for each resolution level are averaged to produce an overall estimate of the watermark.

Ohnishi [120] propose an algorithm similar to the Kunder [117] technique. The most significant difference between the two methods lies in the merging stage of the watermark. Here the author marks the host by forcing the modulo 2 difference between the largest and smallest wavelet coefficients for a particular position and resolution level to be one if $w(n) = 1$ and to be zero if $w(n) = -1$.

In M. Barni, et al. [121] the authors present a public watermarking system. A binary pseudo-random sequence is weighted with a function, which takes into account the human visual system (orientation, brightness, and texture) and is then added to the DWT coefficients of the three largest detail sub-bands of the image (i.e. first level). For watermark detection, the correlation between the watermark to be tested for its presence and the marked coefficients is computed. The value of the correlation is compared to a threshold to decide if the watermark is present or not. Experimental results prove the imperceptibility of the watermark and the robustness against the more common attacks. A model for estimating the sensitivity of the eye to noise - previously proposed for compression applications [122]- is used to adapt the watermark strength to the local content of the image.

Inoue, et al. [123] propose two public digital watermarking schemes to embed a binary code. Both methods are built on a data structure called a zerotree, which is defined in the Embedded Zerotree Wavelet (EZW) algorithm of Shapiro [124]. Zerotree coding is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at a finer scale are likely to be insignificant with respect to T. The zerotree is used to classify wavelet coefficients as insignificant or significant as follows: given an amplitude threshold T, if a wavelet coefficient x and all of it descendants (i.e. coefficients corresponding to the same spatial locations but at finer scales of similar orientation) satisfy $|x| < T$ then they are called insignificant with respect to a given threshold T or zerotree for the threshold T (otherwise significant coefficients). In one method, the zerotrees are constructed for any coarsest sub-band (except LL sub-band) for a specific threshold. Each watermark binary digit is embedded by writing the same data in the location of all elements of the current zerotree. Data is redundantly embedded because insignificant coefficients are generally easy to change under the influence of common signal processing. In the second method, the watermark can be embedded by thresholding and modifying significant coefficients at the coarser levels. However, it is well known that the modification of these components can lead to perceptual degradation of the signal. To avoid this, they make the value of T larger than in the previous method. As a result, the regions in which the watermark is embedded are applied to detailed portions, i.e. edges or textures, in the coarsest scale component. Therefore, embedding the watermark into significant coefficients is difficult for human eyes to perceive. The watermark is detected by using the position of the zerotree's root and the threshold value after the wavelet decomposition of the cover image. It is shown that the proposed method is robust against several common signal processes.

4.8 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) of a function provides a quantitative picture of the frequency content in terms of magnitude and phase. This is important in a wide range of physical problems and is fundamental to the processing and analysis of signals and images. It is very important to know the properties of DFT so that it can be exploited efficiently. Some of these properties are listed below:

Positive Symmetry If $f(n,m); n = 1,2,...,N, m = 1,2,...,M$ is real (which is the case of images), its DFT is conjugate symmetric [114]; that is $F(p,q) = F^*(N-p, M-q)$. To ensure the inverse DFT is real, changes in the magnitude must preserve positive symmetry.

Negative Symmetry This is the same as the above with regard to the phase component (ϕ), but in this case, we have a negative symmetry compounded in the result

$$\phi_{p,q} = \phi_{p,q} + \delta$$

$$\phi_{N-p,M-q} = \phi_{N-p,M-q} - \delta$$

Scaling Scaling in the spatial domain causes inverse scaling in the Fourier domain (i.e. as the spatial scale expands, the frequency scale contracts and the amplitude increases vertically in such a way as to keep the area constant), i.e.

$$\text{if } f(x,y) \xrightarrow{\text{DFT}} F(k_x,k_y) \text{ then } f(ax_x, ax_y) \xrightarrow{DFT} \frac{1}{a}F(\frac{k_x}{a}, \frac{k_y}{a})$$

Translation The translation property of Fourier transform is defined as

$$\text{if } f(x,y) \xrightarrow{\text{DFT}} F(k_x,k_y) \text{ then}$$
$$f(x+a, y+b) \xrightarrow{\text{DFT}} F(k_x,k_y)e^{-i(ak_x+bk_y)}$$

This indicates that the phase is altered only by a translation, i.e. the amplitude is insensitive to the spatial shift of an image. Note that both f and F are periodic functions so it is implicitly assumed that the translation causes the image to be 'wrapped around' (circular translation) [125]. By translation, zero padding of an image will occur if it were placed on a scanner and scanned.

Rotation Rotating the image through an angle in the spatial domain causes the Fourier representation to be rotated through the same angle [125], i.e.

$$f(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta) \xrightarrow{\text{DFT}}$$

$$F(k_x\cos\theta - k_y\sin\theta, k_x\sin\theta + k_y\cos\theta)$$

Log-Polar Representation Most watermarking algorithms have problems in extracting the watermark after an affine geometric transformation on the watermarked object. Some methods try to invert the effect of geometric distortion using the original image. An alternative way is to build a system that can detect the watermark even after a geometric distortion is applied, i.e. Rotation, Scaling and Translation invariance (RST invariant). Most of these systems use the properties of log-polar representation of the spectrum. In a log-polar mapping (which is defined as $x = r\cos\theta, y = r\sin\theta$), the rotation and scaling in a Cartesian coordinate system will result in a translation in the logarithmic coordinate system, i.e.

$$\text{if } f(x,y) \xrightarrow{\text{logh polar mapping}} f(\mu,\theta) \text{ then}$$

$$f(ax,ay) \xrightarrow{\text{log polar mapping}} f(\mu + loga, \theta)$$

and

$$\text{if } f(x\cos(\theta+\delta) - y\sin(\theta+\delta), x\sin(\theta+\delta) + y\cos(\theta+\delta)) \xrightarrow{\text{log polar mapping}} f(\mu, \theta+\delta)$$

From the translation property of the Fourier transform as well as the properties of a log-polar mapping, we can create a RST invariant domain by applying the Fourier transform to the log-polar version of the Fourier magnitude of an image, which is equivalent to computing the Fourier-Mellin transform.

Phase and Magnitude Modulation The DFT is generally complex valued. This leads to a magnitude and phase representation for the image [126]. The human visual system is far more sensitive to phase distortion than magnitude distortion and as a consequence, the DFT magnitude can be altered significantly without affecting the perceived quality of the image. The phase modulation can possess superior noise immunity when compared to amplitude modulation. As a consequence of this, if a watermark is introduced into the phase components with high redundancy, the attacker will cause serious damage to the quality of the image.

4.9 Embedding Techniques in the DFT Domain

O' Ruanaidh, et al. [126] investigate the use of DFT phase for the transformation of information. The condition that the image is composed of real data implies that the Fourier spectrum is symmetric, and because the human eye is more sensitive to phase distortion, watermarking that changes the phase must preserve the negative symmetry. The most

significant components are then selected by comparing the component magnitude squared to the total energy in the spectrum. To detect the watermark, the marked image is simply compared with the original image.

Solachidis, et al [127] propose a watermarking method robust to rotation and scaling. The watermark consists of a 2-D circularly symmetric sequence taking values 1, -1. It has zero mean value. The region in which the watermark is embedded should be a ring covering the middle frequencies. The ring is separated in S sectors and in homocentric circles. Each sector is assigned the same value (1, -1). The watermark is added directly to the magnitude of the DFT domain. If the magnitude becomes negative, it is rounded to 0. The 'conjugate symmetry' property for the DFT must be preserved. The original is not required for detection. The detection is done by finding the correlation between the possibly watermarked coefficients and the original watermarks, comparing the correlation against a threshold.

Kim, et al. [128] discuss the embedding of a binary image (seal image) into another image. The entire watermark is modulated by a binary pseudo-noise matrix (P). The pseudo noise serves for spreading the watermark evenly and is the secret key for retrieving the watermark. The watermark is embedded into the Fourier domain of the cover image by altering the magnitude components ($m_{ij} = m_{ij} + \alpha * P_{ij} * w_{ij}$). The amplitude factor a, is a constant determining the signature strength. The retrieval process can be done without knowledge of the original image. This process starts by approximating the magnitude of the Fourier coefficients of the original image. This can be done by finding the average of the magnitude coefficients around each point in the watermarked cover. The difference between the predicted and the actual value in the watermarked version is divided by the pseudo-noise that was used in the embedding process (which can be regenerated using the key). Experimental results show that this schema gives a high robustness to the distortion such as blurring and lossy compression.

Raymound, et al. [129] have proposed a modification to the system in [128]. They embed a reduced version of the watermark several times using the same method. This repetition can be used in the retrieval process to enhance the watermark.

O Ruanaidh, et al. [125] have introduced the use of the Fourier-Mellin transform for watermarking to embed a watermark which is RST invariant from a digital image. A Fourier transform is first applied which is then followed by a Fourier-Mellin transform. The invariant coefficients are pre-selected for their robustness to image processing and are marked using spread spectrum techniques. The inverse mapping is computed (an inverse log-polar mapping followed by an inverse DFT). Note, that the inverse transformation uses the phase computed during the forward transformations. To extract the watermark, the watermarked

image is transformed into the RST invariant domain which then decodes the watermark.

In Herrigel, et al. [130], the embedding process starts by dividing the image into adjacent blocks. They then map each block into perceptually 'flat' domains by replacing the intensity of each pixel with their logarithm. This step ensures that the intensity of the watermark is diminished in the darker regions of the image where it would otherwise be visible (Weber-Fechner law for HVS response to change of luminance). The Fourier transform is then computed for each block. Finally, the watermark is modulated with magnitude components selected from the middle-frequency bands. To detect rotation and scaling, a template T is embedded into selected components in log polar space. To determine the rotation and scaling that the image suffered, they calculate the normalized cross correlation between the log-polar components and the template pattern T to find the point of best correlation. If the image has neither been rotated nor scaled, then this point is at the origin.

Lin, et al. [131] propose a watermarking algorithm that is robust to RST distortions. The watermark is embedded using the following steps: Find the discrete log-polar mapping for the Fourier magnitude components of the input image (M rows, N Columns). Sum the logs of all values in each column (angle dimension) and add the result of summing column j to the result of summing column $j + N/2$ storing the result in a vector (v). Mix the watermark with v using a weighted average of w and v to produce vector s. Modify all the values in column j of the log-polar Fourier transform so their logs sum to s_j instead of v_j. Invert the log-polar re-sampling of the Fourier magnitude, thus obtaining a modified Cartesian Fourier magnitude. The complex terms of the original Fourier transform are scaled to have the new magnitudes found in the modified Fourier transform. The inverse DFT is applied to obtain the watermarked image. The detection process is as follows: Apply the same signal-extraction process to the watermarked image to produce the extracted vector v. Compute the correlation coefficient between v and input watermark vector w. If the correlation is grater than a threshold T, then the watermark is present, otherwise it is absent.

Chapter 5

Document Authentication using DataGlyph's

5.1 Introduction

In this chapter, a new approach to digital watermarking is presented for a range of applications. The process is defined by using analytical techniques and concepts borrowed from Cryptography. It is based on computing a 'scramble image' by diffusing a watermark image with a noise field. The cover image is then introduced using a simple additive process. The watermark is subsequently recovered by removing the covertext and then correlating the output with the original (key dependent) noise source. For covert encryption, this approach provides the user with a method of hiding ciphertexts (the scrambled image) in a host image before transmission of the data. In this sense, it provides a steganographic approach to cryptology in which the ciphertext is not apparent during an intercept. Decryption is based on knowledge of the key and access to the host image. In terms of watermarking a digital image, the method provides a way of embedding information in an image that can be used to authenticate it, in that it has come from an identifiable source, a method that is relatively insensitive to lossy compression, making it well suited to digital image transmission.

The ICA method (discussed in Section 5.11) is used to apply the process of protecting authenticity of black and white documents. This is intended to demonstrate that this technology can be effectively used in the detection of fraudulent documents. We show that this method can recover (although not perfectly) the original image, after being masked by watermarking with a 2D DataGlyph. We consider extending a DataGlyph to one that is based on the computer generation of a grey level and/or colour texture directly - a 'TextureGlyph'. This is based on using diffusion only watermarking for document authentication and

the combined process of diffusion and confusion for e-to-e based watermarking. We consider an embedding process to counterfoil the detection process using Independent Component Analysis (ICA). Through the application of ICA, we present a method that demonstrates how ICA can be used to detect forged documents. We then show that rescaling of the document does not affect the method, except in dimensional extremes. Extensive research has been contributed in which ICA is used in images and videos. Audio processing (as discussed in Section 5.9) uses an on-the-spot mixing model in which audio/sound signals are convolved within their individual broadcasting auditory surroundings. Vector Space Methods (VSM) and Latent Semantic Analysis (LSA) are used handle text (Section 5.10). Finally we show that if the original watermark has been changed (tampered with) there is a low probability of recovering the image. Due to the immensity of the literature in this area we give a conglomeration of applications.

5.2 Document Authetication

The use of image based information exchange has grown rapdily over the years in terms of both e-to-e image storage and transmission and in terms of maintaining paper documents in electronic form. Further, with the dramatic improvements in the quality of COTS printing and scanning devices, the ability to counterfeit documents has become a widespread problem Consequently, there has been an increasing demand to develop digital watermarking techniques which can be applied to both electronic and printed images (and documents) that can be authenticated, prevent unauthorized copying of their content and withstand a substantial amount of abuse and degradation before and during scanning.

In this chapter, a new approach to digital watermarking is presented for applications. The process is defined by using analytical techniques and concepts borrowed from Cryptography. It is based on computing a 'scramble image' by diffusing a watermark image with a noise field. The cover image (covertext) is then introduced into using a simple additive process (Confusion). The watermark is subsequently recovered by removing the covertext and then correlating the output with the original (key dependent) noise source. For covert encryption, this approach provides the user with a method of hiding ciphertexts (the scrambled image) in a host image before transmission of the data. In this sense, it provides a steganographic approach to cryptology in which the ciphertext is not apparent duing an intercept. Decryption is based on knowledge of the key (which can include noise field generated using Crypstic^{TM}, for example) and access to the host image. In terms of watermarking a digital image, the method provides a way of embedding information in an image that can be used to authenticate that it has come from a identifiable source, a method that is relatively

insensitive to lossy compression, making it well suited to digital image transmission.

With regard to document authentication, the use of diffusion and confusion using a cover is not robust. The reason for this is that the registration of pixels associated with a covertext can not be assured when the composite image is printed and scanned. We therefore consider a diffusion only approach to document authentication which is robust to a wide variety of attacks including geometric attacks, drawing, crumpling, and print/scan attacks. This is because the process of diffusion (i.e. the convolution of information) is compatible with the physical principles of an imaging system and the theory of image formation and thus, with image capture devices (digital cameras and scanners, for example) that, by default, conform to the 'physics' of optical image formation. The diffusion of plaintext (in this case, an image) with a noise field (the cipher) has a synergy with the encryption of plaintext using a cipher and an XOR operation (when both the plaintext and cipher are represented by binary streams). However, decryption of a convolved image (deconvolution) is not as simple as XORing the ciphertext with the appropriate cipher. Here, we consider an approach which is based on pre-conditioning the original cipher in such a way that decryption (de-diffusion) can be undertaken by correlating the ciphertext with the cipher. If a high entropy cipher is used that is uniformly distributed, then the PSDF of the output will be determined by the PSDF of the plaintext (image). Now, if the image is based on naturally occurring objects which are roughly of a self-affine type, then the PSDF may tend to scale as $1/\mid \mathbf{k} \mid^{\beta}$. In other words, the diffusion of self-affine images with white noise will generate output images that are, in effect, random fractal images with fractal-type textures. In this sense, the use of white noise diffusion for document authentication is based on using texture maps which are either fully or partially fractal. Either way, the outputs considered for document authentication are based on printing textures of a type that are determined by the spectral characteristics of the plaintext which can be applied using low resolution COTS printers and scanners.

A basic DataGlyphs is a pattern of forward and backward slashes representing ones and zeroes. This pattern forms an evenly textured field. In this chapter we consider extending a DataGlyph to one that is based on the computer generation of a grey level and/or colour texture directly - 'TextureGlyph'. This is based on using diffusion only watermarking for document authentication and the combined process of diffusion and confusion for e-to-e based watermarking which is discussed in the following section.

5.3 Diffusion and Confusion based Watermarking

In this section we consider approch to watermarking plaitext using both diffusion and confusion. The basic approach is as follows: Given a *plaintext* image and a *covertext* image, the

stegotext image is given by

$$stegotext = ciphertext + covertext$$

where

$$ciphertext = cipher \otimes\otimes plaintext$$

where $\otimes\otimes$ denotes the two-dimensional convolution operation. The problem is to find a cipher which provides a ciphertext that, given the equation above can be well hidden in the covertext.

5.3.1 Fresnel Diffusion Watermarking

There is clearly a compatibility between the use of Fresnel diffusion for watermarking digital images and chirp coding for watermarking digital signals as discussed in the previous chapter. We consider a watermarking model given by

$$I_3(x,y) = rp(x,y) \otimes\otimes I_1(x,y) + I_2(x,y)$$

with PSF

$$p(x,y) = \frac{1}{2}(1 + \cos[\alpha(x^2 + y^2)]$$

and where

$$\|p(x,y) \otimes\otimes I_1(x,y)\|_\infty = 1 \text{ and } \|I_2(x,y)\|_\infty = 1.$$

Here, r is controls that extent to which the host image I_2 dominates the diffused watermark image I_1. In effect, r is like a signal-to-noise ratio, or, in this application a 'diffusion-to-confusion' ratio The output of this process I_3 is the watermarked host image. Recovery of the watermark image is then based on the following process:

$$I_1(x,y) = \frac{1}{r}p(x,y) \odot\odot (I_3(x,y) - I_2(x,y)$$

The method is implemented numerically using a Fast Fourier Transform and application of the two-dimensional convolution and correlation theorem

$$p \otimes\otimes f \Longleftrightarrow PF$$

and

$$p \odot\odot f \Longleftrightarrow P^*F$$

respectively.

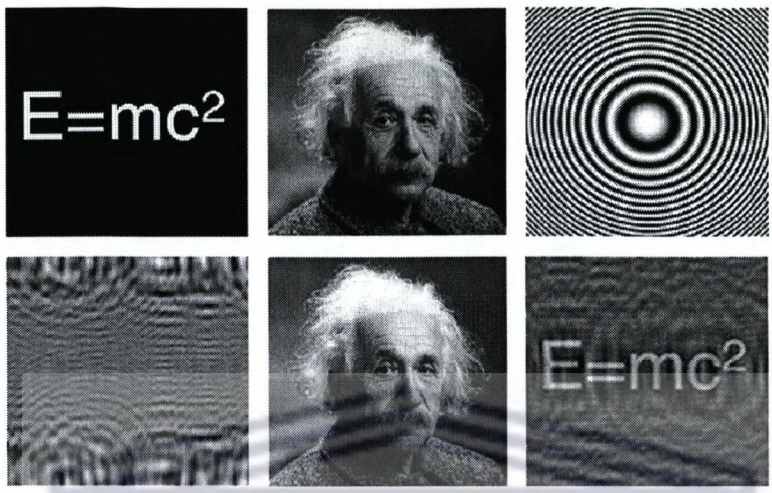


Figure 5.1: From top to bottom and from left to right (all images are 512×512): Watermark I_1, host image I_2, PSF p for $\alpha = 0.001$, diffused image $p \otimes\otimes I_1$, host image after watermarking I_3 for $r = 0.1$, recovered watermark.

Figure 5.1 shows an example result of implementing this watermaking method where a digital image of Albert Einstein (the covertext) is watermarked with a binary image of his most famous of equation (the plaintext). Note that the dynamic range of the diffused field and the reconstruction is relatively low and the images given in Figure 5.1 are displayed by re-quantisation to an 8-bit grey scale of the data $\min[I(x,y)] \leq I(x,y) \leq \max[I(x,y)]$. On the other hand, the low dynamic range of the diffused field allows the diffused field to be added to the host image without any significant distortions becoming discernable other than increasing the brightness of the image[1].

Fresnel diffusion is only of value when the plaintext is of binary form (i.e. a binary image) and when the covertext is well textured throughout in order to 'hide' the diffused plaintext.

In this application, the host image together with (α, r) form the key where the algorithm is taken to be in the public domain. Given these conditions, the method is useful for application to watermarking digital images provided that the distortion accompanying the restoration of the watermark is acceptable. However, the method is not particularly well suited to document (hard-copy) watermarking accept under special circumstances. One such example is given in Figure 5.2 which illustrates a method designed whereby, using standard

[1]In each case the data is re-normalised to floating point values between 0 and 1 before application of grey-scale quantization.

security printing technology, a covert digital thread can be introduced (typically into a print file) that reflects a conventional overt thread. In this example, a one-dimensional Fresnel transform (a symmetric chirp function) is used to encode a single or multiple bar code and the result embedded into an existing print file. Recovery of a digital thread is obtained through correlation of the same symmetric chirp function with a scanned image. This approach is analogous to the application of a matched filter and, in particular, the deconvolution on linear FM (Frequency Modulated) chirps. Applications include currency, bank bonds and other security documents. In this case the model is based on the following:

$$I_3(x,y) = rp(x,y) \otimes \otimes I_1(x,y) + I_2(x,y)$$

where I_1 is a binary image consisting of a bar code (single or multiple bars),

$$p(x,y) = \frac{1}{2}[1 + \cos(\alpha x^2)]$$

and I_2 is the host image. Recovery of the bar code (i.e. estimation $\hat{I}_1$ of I_1 is then given by

$$\hat{I}_1(x,y) \sim p(x,y) \odot \odot I_3(x,y) + \epsilon$$

where

$$\epsilon = p(x,y) \odot \odot I_2(x,y)$$

such that, provided I_2 does not correlate with p (e.g. I_2 is a textured image), then

$$\|\epsilon(x,y)\| << \|p(x,y) \odot \odot I_3(x,y)\|$$

This condition allows an exact reconstruction of I_1 to be obtained through application of a threshold to binarize $\hat{I}_1$.

5.3.2 Noise Diffusion Watermarking

The principal weakness associated with Fresnel diffusion is that the cipher is based on a deterministic function. To overcome this, we consider a noise diffusion approach. Diffusion by noise is compounded in the model

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

Suppose we use this approach to diffuse an image u_0 using a key based noise field n. There are now two approaches to solving the problem: Given u and n, obtain u_0. We can invert or deconvolve by using the convolution theorem giving

$$u_0(\mathbf{r}) = \mathcal{F}^{-1}\left[\frac{U(\mathbf{k})N^*(\mathbf{k})}{|\,N(\mathbf{k})\,|^2}\right]$$

Figure 5.2: 600dpi scan of a 20 Pounds Sterling Bank (of England) note (above) whose graphic file includes the addition of symmetric chirp (centre) and recovery of a digital thread.

where $\mathcal{F}^{-1}$ is the inverse Fourier transform operator, N is the Fourier transform of n and U is the Fourier transform of u. However, this approach requires regularisation in order to eliminate any singularities when $\mid N \mid^2 = 0$ through application of a constrained deconvolution filter such as the Wiener filter. Alternatively, if n is the result of some random number generating algorithm, and since the functional form of n is arbitrary, we can construct the stochastic field

$$m(\mathbf{r}) = \mathcal{F}^{-1}\left[\frac{N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}\right]$$

where $\mid N(\mathbf{k}) \mid^2 > 0$, the diffused field now being given by

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

The inverse problem is then be solved by correlating u with n, since

$$n(\mathbf{r}) \odot_{\mathbf{r}} u(\mathbf{r}) \Longleftrightarrow N^*(\mathbf{k})U(\mathbf{k})$$

and

$$N^*(\mathbf{k})U(\mathbf{k}) = N^*(\mathbf{k})M(\mathbf{k})U_0(\mathbf{k}) = N^*(\mathbf{k})\frac{N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}U_0(\mathbf{k}) = U_0(\mathbf{k})$$

so that

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

The condition that $\mid N(\mathbf{k}) \mid^2 > 0$ is simply achieved by implementing the following process:

$\forall \mathbf{k}$
if $\quad \mid N(\mathbf{k}) \mid^2 = 0$
then $\mid N(\mathbf{k}) \mid^2 = 1$

This result can be used to 'embed' one data field in another. Consider the case when we have two independent images $I_1(x,y) \geq 0 \forall x,y$ and $I_2(x,y) \geq 0 \forall x,y$ and we consider the case of embedding I_1 with I_2. We construct a noise field $m(x,y) \geq 0 \forall x,y$ *a priori* and consider the equation

$$u(x,y) = rm(x,y) \otimes \otimes I_1(x,y) + I_2(x,y)$$

where

$$\|m(x,y) \otimes \otimes I_1(x,y)\|_\infty = 1 \text{ and } \|I_2(x,y)\|_\infty = 1.$$

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 I_1 is a perturbation of the 'host image' I_2. This provides us with a way of watermarking one image with another, R being referred to as the watermarking ratio[2] This approach is of course identical to the Fresnel diffusion method considered in the last section but where the Fresnel PSF is replaced with the pre-conditioned noise field m . However, for applications in image watermarking, the diffusion of an image with noise provides a superior result because: (i) a noise field provides more uniform diffusion; (ii) noise fields can be generated using random number generators that depend on a single initial value or seed (i.e. a private key). The noise field (which should be uniformly distributed) can be created using any number of random number generators. In each case, the algorithm used provides an output that is key dependent and thus, given a known algorithm, reconstruction of the watermark is achieved with knowledge of the private key together with the host image. Generation of the noise field can also be obtained using chaos based ciphers. However, in this case, the floating point output from the cipher(s) must be used to produce a digital integer stream rather than a bit stream through stretching and rounding, e.g. if the uniformly distributed floting point stream is given by c_i where $c_i \in [\text{in}(f_i), \max(f_i)]$ where $f_i \in [0,1]$ is the output from a chaotic cipher then, the noise field is given by

$$x_i = \frac{[c_i - \min(f_i)]}{\max[c_i - \min(f_i)]}$$

so that $\|\mathbf{x}\|_\infty = 1$. The two-dimensional array $n(x,y)$ is generated from x_i on a row-by-row (or column-by-column basis), i.e. filling each row of size N with random floating point

[2]Equivalent, in this application, to the standard term 'Signal-to-Noise' or SNR ratio as used in signal and image analysis.

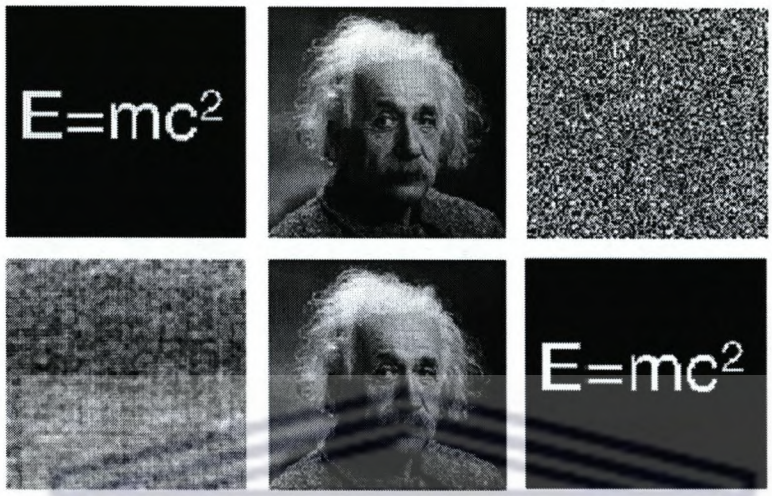


Figure 5.3: From top to bottom and from left to right (all images are 512×512): Watermark I_1, host image I_2, PSF m, diffused image $m \otimes \otimes I_1$, host image after watermarking I_3 for $r = 0.01$, recovered watermark.

numbers from the array x_i, $i = 1, 2, ...N^2$. Here, the seed is any integer such as 1873... (when applying a conventional PRNG with integer arithmetic) or a decimal number (between 0 and 1) such as 0.1873 (when applying a chaos based PRNG with floating point arithmetic) which can be based on the application of a PIN (Personal Identity Number) or a password (e.g. 'Enigma', which in terms of an ASCII string - using binary to decimal conversion - is '216257556149' or "0.216257556149'). Alternatively, noise fields can be created indirectly through application of a variety of cryptographic systems. This is of practical value, because in this application, it is necessary to exchange the host image between sender and receiver and thus, the exchange of a noise field becomes compatible with a necessary protocol, both the noise field and the host image being transmitted as ciphertext using a given encryption standard.

An example of this approach is shown in Figure 5.3 which should be compared with Figure 5.1. Here, the image I_2 (the 'host image' - covertext) is watermarked by another image I_1 (the 'watermark image' - plaintext) with to produce output image I_3 with $r = 0.01$. The relatively small perturbation of the term $m \otimes \otimes I_1$ to the host image I_2 for $r = 0.01$ does not affect the output image in any way that is visually significant.

A further advantage of noise diffusion is that is not limited to watermarking covertexts with binary image plaintexts. However, the effect of adding the diffused greyscale watermark image to the host image yields a different, slightly brighter image because of the perturbation

of I_2 by $Rm \otimes \otimes I_1$. This effect can be minimized by introducing a smaller watermarking ratio such that the perturbation is still recoverable by subtracting the host image from the watermarked image, an example being given in Figure 5.4.

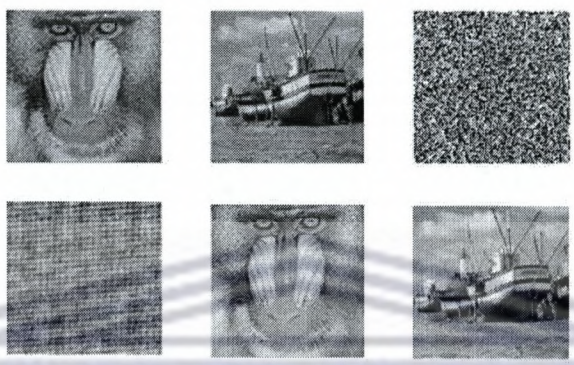


Figure 5.4: Example of watermarking an image with another image using noise based diffusion. The 'host image' I_2 (top-left) is watermarked with the 'watermark image' I_1 (top-centre) using the diffuser (top-right) given by a uniform noise field n whose pixel-by-pixel values depend upon the seed used (the private key). The result of computing $m \otimes \otimes I_1$ (bottom-left) is added to the host image for $r = 0.1$ to generate the watermarked image u (bottom-centre). Recovery of the watermark image I_1 (bottom-right) is accomplished by subtracting the host image from the watermarked image and correlating the result with the noise field n.

For the purpose of further quantifying the basic algorithms involved in this method of watermarking the reader is referred to the pseudo-coded function that follows:

```
void function WM(cipher,plaintext,covertext,wartermark,size,pth)
\\
\\ Fuction: Watermarks an image using noise diffusion
\\
\\ Input arrays: cipher    - noise field image
\\               plaintext - the watermark image
\\               covertext - host image
\\ N.B. All input arrays are taken to be of type float with
\\      values ranging from 0 to 1 inclusively.
\\
\\ Parameters: size - image size (assumed to be size x size)
```

```
\\            pth - diffused plaintext to host image ratio
\\
\\ Ouput: watermrk - watermarked image
\\
\\ Internal functions: FFT  - Foward Fast Fourier Transform
\\                     IFFT - Inverse Fast Fourier Transform
\\                     MAX  - Computes the maximum value
\\                     REAL - Extracts the real component
\\                     ABS  - Computes the absolute value
\\
cipher=FFT(cipher);\\Compute spectrum of cipher
plaintex=FFT(plaintext); \\Compute spectrum of plaintext
powerspectrum=ABS(cipher)*ABS(cipher);\\Compute Power Spectrum

\\ Pre-condition power spectrum of cipher
FOR i=1 to size AND j=1 to size DO:
        temp=pp[i,j];
           IF temp==0
        powerspec[i,j]=1;
           ELSE
        powerspec[i,j]=powerspec[i,j];
           END IF
END DO

\\ Diffuse plaintext image with pre-conditioned cipher
FOR i=1 to size AND j=1 to size DO:
    diffusion[i,j]=cipher[i,j]*plaintext[i,j]/powerspec[i,j];
END DO

diffusion=REAL(IFFT(diffusion));\\ Compute real part of IFFT
diffusion=diffusion/MAX(diffusion);\\ Normalise diffused field

\\ Compute the watermark
FOR i=1 to size AND j=1 to size DO:
watermark[i,j]=r*diffusion[i,j]+covertext[i,j];
END DO
```

```
watermark=watermark/MAX(watermark);\\ Normalise for output

void function RECWM(cipher,watermark,covertext,plaintext,size)

\\ Function: Recovers watermark from watermarked image
\\
\\ Input arrays: cipher    - noise field image
\\               watermark - watermarked image
\\               covertext - host image
\\ N.B. All input arrays are taken to be of type float with
\\      values ranging from 0 to 1 inclusively.
\\
\\ Parameters: n - image size (assumed to be size x size)
\\
\\ Output: plaintext - recovered watermark image
\\
\\ Internal functions: FFT  - Forward Fast Fourier Transform
\\                     IFFT - Inverse Fast Fourier Transform
\\                     MAX  - Computes the maximum value
\\                     REAL - Exatracts the real component
\\                     CONJ - Conjugates a complex array
\\
\\Subtract covertext from watermarked image
FOR i=1 to size AND j=1 to size DO:
        diffusion=watermark-covertext;
END DO

cipher=FFT(cipher);       \\Compute spectrum of cipher
diffusion=FFT(diffusion); \\Compute spectrum of diffused field

\\ Correlate diffused field with cipher
FOR i=1 to n AND j=1 to n DO:
        plaintext[i,j]=CONJ(cipher[i,j])*diffusion[i,j];
END DO
```

```
plaintext=REAL(IFFT(plaintext));   \\Compute real part of IFFT
plaintext=plaintext/MAX(plaintext);\\ Normalise output
```

5.3.3 Steganography and Cryptography

One of the principal components associated with the development of methods and algorithms to 'break' ciphertext is the analysis of the output generated by an attempted decrypt and its evaluation in terms of an expected type. The output type is normally assumed to be plaintext, i.e. the output is assumed to be in the form of characters, words and phrases associated with a natural language such as English or German, for example. If a plaintext document is converted into an image file then the method described in the previous section on 'Noise Diffusion Watermarking' can be used to diffuse the plaintext image I_1 using any other covertext image I_2 to produce stegotext I_3. If both I_3 and I_2 are then encrypted, any attack on these data will not be able to make use of an 'analysis cycle' which is based on the assumption that the decrypted output is plaintext. This approach provides the user with a relatively simple method of 'confusing' the cryptanalyst and invalidates attack strategies that have been designed and developed on the assumption that the encrypted data have been derived from plaintext alone.

In steganography, one message is hidden inside another, without disclosing the existence of the hidden message or making it apparent to an observer that this message contains a hidden message [134]. Moreover, the information hidden by a watermarking system is always associated with the object to be protected or its owner while steganographic systems just hide information. On the other hand, cryptography can be defined as the study of secret writing (i.e. concealing the contents of a secret message by transforming the original message into a form that cannot be easily interpreted by an observer). The method considered here (diffusion and confusion) can be easily used in both applications. The hidden message can be transformed into a diffused form (i.e encrypted) and inserted into the background. The hidden information might have no relation with the text (foreground). At the same time, backgrounds are usually used with documents and so diffused data will not necessarily trigger the attention of an observer. Moreover, the hidden message is also encrypted which increases the security level of such documents.

5.4 Hardcopy Steganography

The use of the model

$$stegotext = ciphertext + covertext$$

can be applied for watermarking digital images associated with electronic-to-electronic type communications in which there is no loss of information. This method can be used to watermark digital image for the purpose of authentication but can also be viewed as a method of covertly transmitting ciphertext when the plaintext is converted to the form of a digital image. Steganography and watermarking techniques are also of value for hardcopy 'data' which has a range of applications for authenticating printed material and copyright validation, for example. However, to be of practical value to the security printing industry the methods must be robust to the significant distortions generated by the printing and/or scanning process. A simple approach is to add information to a printed page that is difficult to see. For example, some modern colour laser printers, including those manufactured by HP and Xerox, print tiny yellow dots which are added to each page. The dots are barely visible and contain encoded printer serial numbers, date and time stamps. This facility provides a useful forensics tool for tracking the origins of a printed document which has only relatively recently been disclosed.

5.4.1 Diffusion Only Watermarking

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

Depending on the printing process applied, a number of distortions will occur which diffuse the information being printed. Thus, in general, we can consider the printing process to introduce an effect that can be represented by the convolution equation

$$u_{\text{print}} = p_{\text{print}} \otimes \otimes u.$$

where u is the original electronic form of a diffused image (i.e. $u = n \otimes \otimes u_0$) and p_{print} is the point spread function of the printer. An incoherent image of the data, obtained using a flat bed scanner, for example (or any other incoherent optical imaging system), will also have a characteristic point spread function p_{scan} say. Thus, we can consider a scanned image to be

given by

$$u_{\text{scan}} = p_{\text{scan}} \otimes \otimes u_{\text{print}}$$

where u_{scan} is taken to be the digital image obtain from the scan. Now, because convolution is commutative, we can write

$$u_{\text{scan}} = p_{\text{scan}} \otimes \otimes p_{\text{print}} \otimes \otimes p \otimes \otimes u_0 = p \otimes \otimes p_{\text{scan/print}} \otimes \otimes u_0$$

where

$$p_{\text{scan/print}} = p_{\text{scan}} \otimes \otimes p_{\text{print}}$$

which is the print/scan point spread function associated with the processing cycle of printing the image and then scanning it back into electronic form. By applying the method discussed earlier, we can obtain a reconstruction of the watermark whose fidelity is determined by the scan/print point spread function. However, in practice, the scanned image needs to be re-sized to that of the original. This is due to the scaling relationship (for a function f with Fourier transform F)

$$f(\alpha x, \beta y) \Longleftrightarrow \frac{1}{\alpha\beta} F\left(\frac{k_x}{\alpha}, \frac{k_y}{\beta}\right).$$

The size of any image captured by a scanner or other device will depend on the resolution used. The size of the image obtained will inevitably be different from the original because of the resolution and window size used to print the diffused image u and the resolution used to scan the image. Since scaling in the spatial domain causes inverse scaling in the Fourier domain, the scaling effect must be 'inverted' before the watermark can be recovered by correlation since correlation is not a scale invariant process. Re-sizing the image (using an appropriate interpolation scheme such as the bi-cubic method, for example) requires a set of two numbers n and m (i.e. the $n \times m$ array used to generate the noise field and execute the diffusion process) that, along with the seed required to regenerate the noise field, provides the 'private keys' needed to recover the data from the diffused image. An example of this approach is given in Figure 5.5 which shows the result of reconstructing four different images (a photograph, finger-print, signature and text) used in the design of an impersonalized debit/credit card. The use of 'diffusion only' watermarking for print security can be undertaken in colour by applying exactly the same diffusion/reconstruction methods to the red, green and blue components independently. This provides two additional advantages: (i) the effect of using colour tends to yield better quality reconstructions because of the colour combination process; (ii) for each colour component, it is possible to apply a noise field with a different seed. In this case, three keys are required to recover the watermark although it should be noted that, due to the errors associated in the extraction of each colour

Figure 5.5: Example of the application of 'diffusion only' watermarking. In this example, four images of a face, finger-print, signature and text have been diffused using the same cipher and printed on the front (top-left) and back (bottom-left) of an impersonalized identity card using a 600 dpi printer. The reconstructions (top-right and bottom-right, respectively) are obtained using a conventional flat-bed scanner based on a 300 dpi grey-level scan.

component from a colour scan, this approach does not yield reconstructions with the same degree of robustness as in the case when the same key/algorithm is used for each colour component.

Because this method is based on convolution alone and since

$$u_{\text{scan}} = p_{\text{scan/print}} \otimes \otimes u_0$$

as discussed earlier, the recovery of the u_0 will not be negated by the distortion of the point spread function associated with the print/scan process, just limited or otherwise by its characteristics. Thus, if an image is obtained of the printed data field $p \otimes \otimes u_0$ which is out of focus due to the characteristics of $p_{\text{scan/print}}$, then the reconstruction of u_0 will be out of focus to the same degree. Decryption of images with this characteristic is only possible using an encryption scheme that is based a diffusion only approach. However, if a covertext image f is introduced so that

$$u_{\text{scan}} = p_{\text{scan/print}} \otimes \otimes u_0 + p_{\text{scan/print}} \otimes \otimes f$$

then because

$$u_{\text{scan}} - f \neq p_{\text{scan/print}} \otimes \otimes u_0$$

recovery of the plaintext is not possible which is why we resort to a diffusion only method approach.

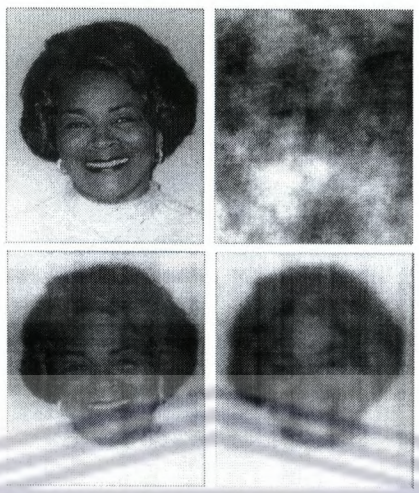


Figure 5.6: Original image (top-left), diffused image (top-right), reconstruction using a flatbed scanner (bottom-left) and reconstruction using a mobile phone (bottom-right). These images have been scanned in grey scale from the original colour versions printed on to a personalised identity card at 600dpi stamp-size (i.e. 2cm×1.5cm).

Figure 5.6 illustrates the recovery of a diffused image printed onto a personal identity card obtained using a flat bed scanner and then captured using mobile phone camera. In the latter case, the reconstruction is not in focus because of the wide-field nature of the lens used. However, the fact that recovery of the watermark is possible with a mobile phone means that the scrambled data can be transmitted securely and the card holders image (as in this example) recovered remotely and transmitted back to the same phone for authentication. This provides the necessary physical security needed to implement such a scheme in practice and means that specialist image capture devices are not required on site.

The diffusion process can be carried out using a variety of different noise fields other than the uniform noise field considered here. Changing the noise field can be of value in two respects: first, it allows a system to be designed that, in addition to specific keys, is based on specific algorithms which must be known *a priori*. These algorithms can be based on different pseudo uniform random number generators and/or different pseudo chaotic number generators that are post-processed to provide a uniform distribution of numbers. Second, the diffusion field depends on both the characteristics of the watermark image and the noise field. By utilizing different noise fields (e.g. Gaussian noise, Poisson noise), the texture of the output field can be changed. The use of different noise fields is of value when different textures are required that are aesthetically pleasing and can be used to create a background that is

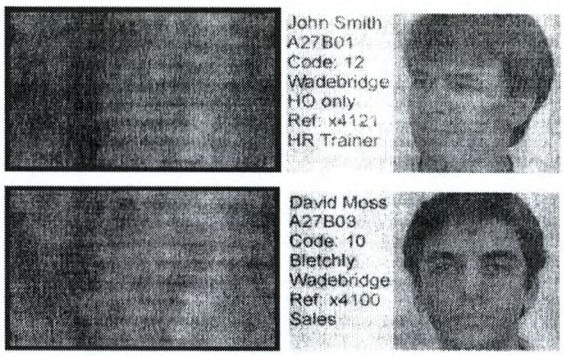


Figure 5.7: Example of the diffusion of composite images with the inclusion of a reference frame for enhancing and automating the processes of cropping and orientation. In each case the data fields have been printed and scanned at 300 dpi.

printed over the entire document - texture maps. In this sense, variable noise based diffusion fields can be used to replace complex print security features with the added advantage that, by de-diffusing them, information can be recovered. Further, these fields are very robust to data degradation created by soiling, for example. In the case of binary watermark images, data redundancy allows reconstructions to be generated from a binary output, i.e. after binarizing the diffusion field (with a threshold of 50% for example). This allows the output to be transmitted in a form that can tolerate low resolution and low contrast copying, e.g. a fax.

The tolerance of this method to printing and scanning is excellent provided the output is cropped accurately (to within a few pixels) and oriented correctly. The processes of cropping and orientation can be enhanced and automated by providing a reference frame in which the diffused image is inserted. This is illustrated in Figure 5.7 which, in addition shows the effect of diffusing a combination of images. This has the effect of producing a diffused field that is very similar but nevertheless conveys entirely different information.

5.4.2 Covertext Addition and Removal

Because diffusion only watermarking is based on convolution/correlation operations it is relatively insensitive to contrast stretching and compression. This provides the opportunity to introduce covertext in the form of the addition of foreground information (e.g. text) to a printed document that has been watermarked *a priori* with a grey scale texture map whose brightness and contrast has been adjusted to be unobtrusive with regard to the covertext

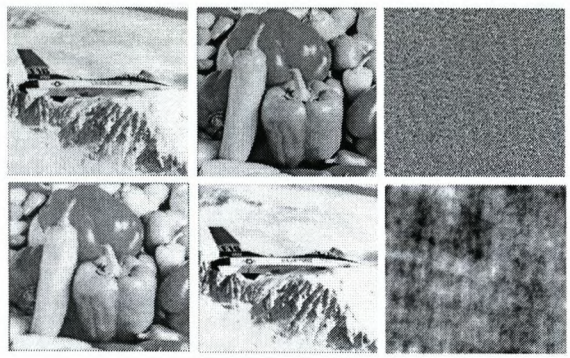


Figure 5.8: Example of a covert watermarking scheme. I_1 (top-left) is 'processed' with I_2 (top-middle) to produce the noise field (top-right). I_2 is printed at 600 dpi, scanned at 300 dpi and then re-sampled back to its original size (bottom-left). Correlating this image with the noise field generates the reconstruction (bottom-centre). The reconstruction depends on just the host image and noise field. If the noise field and/or the host image are different or corrupted, then a reconstruction is not achieved, as in the example given (bottom-right).

(i.e. the watermark is made bright compared to black text). Alternative, once the texture field has been designed, it may be introduced into a text editor that provide the inclusion of watermarks. For example, Microsoft Word has the facility to include a printed watermark (Format→Background→Printed Watermark) that provides the option to select a Picture Watermark (Existing Watermark) with options on scale and 'Washout'.

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

5.5 Covert Watermarking using Diffusion

Watermarking is usually considered to be a method in which the watermark is embedded into a host image in an unobtrusive way. Another approach is to consider the host image to

be a data field that, when processed with another data field, generates new information.

Consider two images I_1 and I_2. Suppose we construct the following function

$$n = \mathcal{F}_2\left(\frac{I_1}{|I_1|^2} I_2\right)$$

where $\widetilde{I_1} = \mathcal{F}_2[I_1]$ and $\widetilde{I_2} = \mathcal{F}_2[I_2]$. If we now correlate n with I_1, then from the correlation theorem

$$I_1 \odot \odot n \Longleftrightarrow \widetilde{I_1^*} \frac{\widetilde{I_1}}{|\widetilde{I_1}|^2} \widetilde{I_2} \Longleftrightarrow I_2.$$

In other words, we can recover I_2 from I_1 with a knowledge of n. Because this process is based on convolution and correlation alone, it is compatible and robust to printing and scanning, i.e. incoherent optical imaging. An example of this is given in Figure 5.8. In this scheme, the noise field n is the private key required to reconstruct the watermark and the host image can be considered to be a public key.

5.6 Applications of Texture Coding

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

In this section, we consider a range of application to which the method can be applied [3]

5.6.1 Authentication

Authentication of a document image should ensure that the document has not been altered from the time it is created and signed by the author to the time it is received at the destination. Authentication of paper documents is an important concern as the ability of counterfeiters has increased substantially in recent years. This is contributed to by the dramatic improvement in the capability of high resolution scanners and printers. Moreover, digital documents can be accessed and modified by intruders relatively easily. This is especially true in the case of documents that are exchanged over the Internet.

[3] Based on K. W. Mahmoud, Low Resolution Watermarking for Print Security, PhD Thesis, Loughborough University, 2005, (Chapter 7).

Using the model and methods discussed here, a *selective authentication* approach can be applied in which only significant changes cause authentication to fail. This can be verified by embedding information in a document that be later be verified as to whether it has been tampered with or otherwise.

5.6.2 Photo Verification

Figure 5.6 shows an example of a photo verification system that can be incorporated into an ID card where a photograph of the card holder is texture coded and printed beside the original image. Substantial editing, such as changing the original photo, will be illegitimate because it will completely change the interpretation of the card. Thus, a photo verification system can be designed to do the following:

1. Capture the diffused watermark using any tool (scanner, camera, etc).
2. Read the key. The key might be:
   (a) Encoded using bar code, or
   (b) Stored in a local Database, or
   (c) Stored in a distance Database that can be accessed via the Internet.
3. Extract the watermark.
4. Verify the authenticity by comparing the original photo with the extracted one. This can be done either by:
   (a) A subjective test, using the judgment of human beings. For more details on the scales that have been suggested for use in evaluation of watermarking quality refer to [135].
   (b) Quality metrics, such as the Mean Square Error or Chi-square test.
   (c) Any other matching algorithm including the application of an Artificial Neural Network as required.

Such a system can be modified to include more information in the diffused watermark as required, such as the name of the ID card holder. Moreover, texture coding can be used to generate a de-personalised ID card either on an individual image base (Figure 5.5) or in terms of a composite image (Figure 5.7).

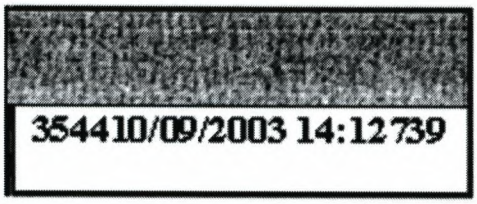


Figure 5.9: Coded Statistical Information

5.6.3 Statistical Verification

When a document is prepared using MS Word, for example, or any other major word processing package, statistical information from the document can be gathered; information about the author, date and time, number of characters and spaces and so on. A verification system can use this information to check the authenticity of the document. Any attempt at modification of the file will be reflected in its statistics. The system can either incorporate these data in plain text or as a diffused code into a patch on the document which is encoded into an indecipherable image as shown in figure 5.9. The image needs to be attractively packaged in an appropriate place on the document - the bottom right hand corner or example.

It is assumed that the recipient of the document (scanned or electronic) will have the appropriate software available. The encoded image is read into the decoding software and *text-recognition* used to reveal the text which is then compared with the plain-text statistics of the document. The data in the image can alternatively be checked manually against the statistics of the file instead of using text recognition. Each author can have a particular key for encoding of image. Upon receipt, the recipient applies that particularly key to decode the image. Alternatively, a separate one-time PIN can be transmitted to the recipient in order to decode the image.

5.6.4 Original Copy Verification

When a document is scanned, subject to the scanner type and settings (including the resolution, for example), the output digital image file will have a specific statistical characteristic compounded in the histogram. If the document is copied and scanned again then this characteristic histogram will change because of the copy process. In general, a copied document will tend to have a smoother histogram since it is, in effect, the original document image convolved with a PSF that is characteristic of the copier (a function of the composite scan/print process). By printing a texture code of the histogram of the original document, typically on the back of the document, the document can be scanned and the histogram compared with

the watermark, at least within an acceptable tolerance. This application has value in the authentication of high value documents such a Bank Bonds, an example of which is given in Figure 5.10. Figure 5.11 shows the texture map and the reconstruction of the plaintext, i.e. a histogram of the luminance of the original colour image together with some basic statistical information. By specifying the type of scanner and operational constraints, statistical information of this type (i.e. the mean, standard deviation and median, for example) can be used to qualify whether or not the Bank Bond or other high value documents has been copied. This statistical verification may include measures relating to the RGB components for the case of high value colour document (which is usually the case but can not be reproduced here). Although each scan (using the same scanner with identical settings) will not output an identical digital image (due slight differences in the crop, for example, as well as the natural 'jitter' of the scanner) the statistical information should not change significantly unless a copy has been made, acceptable tolerances having been established *a priori*.

5.6.5 Component Verification

The system described in the previous section can be extended to include a 'specific parts' from of the text that must be correct, e.g. a sum of money, name of beneficiary etc. An example is shown in figure 5.12.

After decoding, the message would be as given in the figure shown in 5.13 Clearly, the diffused code could be placed into the background of each data field (i.e. instead of placing it in the next empty line).

5.6.6 Transaction Tracking

Also called *fingerprinting*, transaction tracking involves the embedding of a different watermark into each distributed copy. This is especially useful for identifying people who obtain a document legally but illegally redistribute it.

5.6.7 Leaked Document Monitoring

One common method to monitor and discover any 'leak' associated with a very important document is to use visible marks. For example, highly sensitive documents are sometimes printed on backgrounds containing large gray digits using a different number for each copy. Records are then kept about who has which copy. Of course, imperceptible watermarks (or at least diffused watermark) are preferable to visible marks. They are easy to remove/replace from a document when it is copied. Using this model for document watermarking, the

Figure 5.10: Example of a high value Bank Bond (grey scale of colour image).

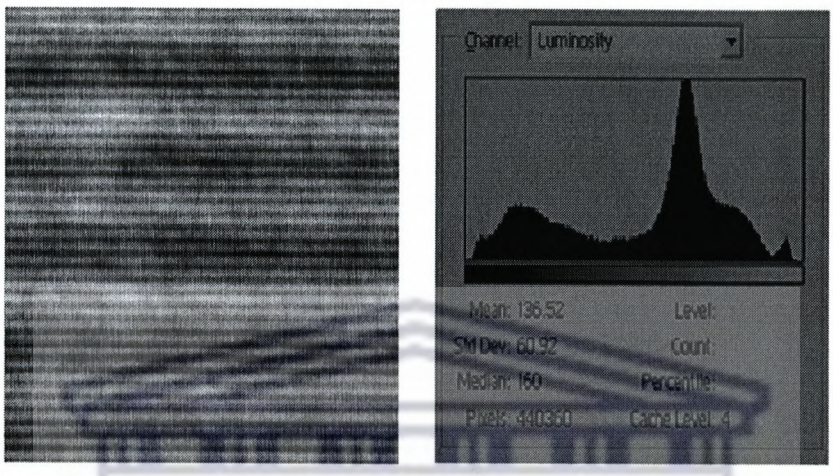


Figure 5.11: Texture map (left) and reconstruction of statistical information relating to scan of original document using Adobe Photoshop V5.

Bank X with sort code 12-23-34 will transfer the sum of $2.500 from account number [redacted]
9876543 to account number 12345678 at Bank Y with sort code 98-76-54.
[redacted]

Figure 5.12: Coded Specific Part from a Document

Bank X with sort code 12-23-34 will transfer the sum of $2.500 from account number
12-23-34 | $2.500 |
9876543 to account number 12345678 at Bank Y with sort code 98-76-54.
| 9876543 | 12345678 | 98-76-54 |

Figure 5.13: Revealed Document after Coding Specific Parts.

tracking number is diffused and inserted into the background. The diffused watermark is inseparable from the document. The adversary (a person who attempts to remove, disable, or forge a watermark for the purpose of circumventing its original purpose) does not know the embedded number and can not recognize the difference between copies (it is difficult for human eyes to find a difference between two copies with different watermarks).

5.6.8 Owner Identification (Copyright)

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

5.6.9 Signature Verification

Handwritten signatures are commonly used to certify the contents of a document or to authenticate legal transactions. A handwritten signature is a well-known biometric attribute. Other biometric attributes, which are commonly used for authentication include iris, hand geometry, face, and fingerprints (See [136] and [137]). While attributes like the iris and fingerprints do not change over time, they require special and relatively expensive hardware to capture the biometric data. An important advantage of the signature over other biometric attributes is that it has been traditionally used in authenticating documents and hence is socially accepted. Signature verification is usually done by visual inspection. In automatic signature verification, a computer takes over the task of comparing two signatures to determine if the similarity between the signatures exceeds some pre-specified threshold. There a many similarity measures that can be used for this purpose. Figure 5.14(left) shows an example for this approach. The signature of the customer is diffused and inserted into the background of the cheque. Each customer has their own key that is known only to them and their bank. They use the key to generate the background and then print the cheque. The bank then uses the key to extract the customer signature from the cheque. If the extracted and the existing signatures on the cheque are matched to each other, then the cheque is accepted.

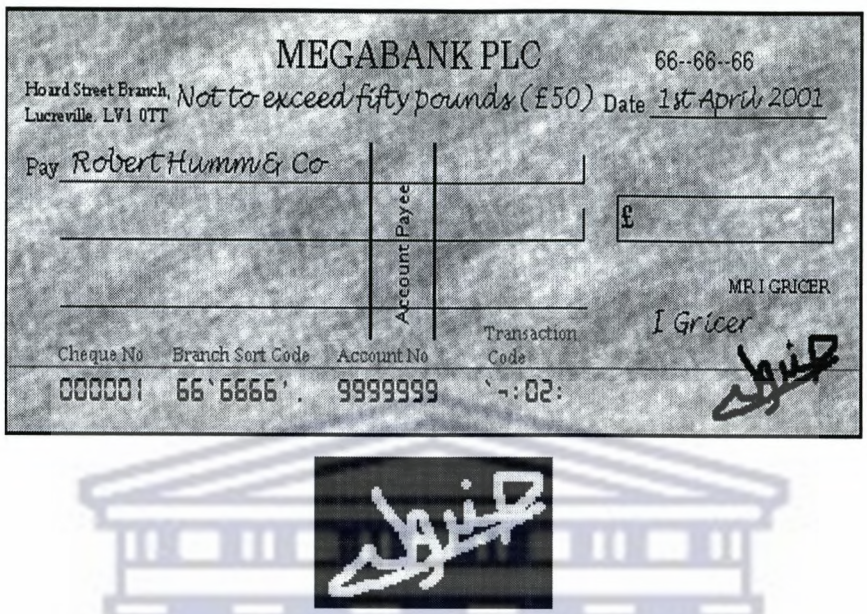


Figure 5.14: Watermarked cheque (above) and recovered signature from watermark (below); from K. W. Mahmoud, Low Resolution Watermarking for Print Security, PhD Thesis, Loughborough University, 2005.

5.7 Case Study: Passport Authentucation

Like any other security document, ID card and so on, a passport consists of a number of security features depending on the sophistication on the design associated with the authority responsible for an issue. These range from the use of printing complex background, microprinting, conventional paper watermarking, UV watermarking, foil holograms, ghost images and so on. Each of these security features my be more or less dificult to counterfeit depending on the sophistication of the feature and the counterfeiter. In this case study, we consider the use of texture coding within the context of authenticating a passport including the protocol associated with a typical 'cycle'. The method is simple and cost effective to implement in terms of the hardware required, i.e. Standard PC, flatbed scanner and printer, all of which are COTS. All that is required is a remote web site hub to which digital scans of the texture code can be emailed and where a decrypt can take place, forwarding the result back to the point of enquiry.

The principal idea is to take a low resolution scan (say 600dpi) of the page (or pages) of a passport that contains the primary information, e.g. Passport number, Name, Date of birth, Signature and Photograph of the passport holder - the plaintext. This plaintext is

then forwarded to a designated Hub where it is diffused with a unique noise field that is maintained at the Hub alone to produce the ciphertext. The result is then emailed back to the user, printed and the result (permanently or as required) inserted into the passport, a process that is similar to issuing a Visa, for example. At any point of contact, if the passport requires authentication, the ciphertext is scanned and the digital image emailed to the appropriate Hub where upon it is decrypted and the result (the watermark) sent back to the point of origin. Automation of this cycle would require a new infrastructure to be established which is both time consuming and expensive. Instead, the cycle described above would be best suited for use with regard to spot checks at an airport terminal, for example, especially if the holder of the passport or the passport itself is suspect. The scanning process (using a standard flat bed scanner, for example) can then be undertaken while the holder of the passport is waiting for it to be authenticated (or otherwise) based on a visual comparison between the decrypt and the plaintext.

An example of the system designed to implement this method working with full colour images in which each RGB component is diffused with the same cipher is given in Figure 5.15. This figure shows the GUI and the result of decrypting the texture code associated with authors passport after being scanned at 300dpi and the output emailed as a JPEG attachment.

5.8 Application of ICA

We consider an embedding process to counterfoil the detection process using Independent Component Analysis (ICA) and present a method that demonstrates how ICA can be used to detect forged documents. The expansion of network systems and the lack of difficulty in reproducing digital media which does not lack quality have necessitated novel techniques to restrain the forbidden reproduction, forgery and dissemination of digital data [191], [192], [193] and [194]. Watermarking and cryptography are devices that assist the enforcing of copyright protection and authentication of digital audio, images and video. For the purposes of this investigation we focused on images only.

Robust digital watermarks used for copyright protection can be explained as undetectable information concealed in a digital media. This information should be detectable as long as the quality of the content is considered as being acceptable. Despite many algorithms, methods, techniques and fully functional systems being used to hide information, the main dilemma is that the bulk of these methods use a symmetric key [195].

Both symmetric watermarks and symmetric cryptography is similar: whereby the exact key is used to encode the information and to authenticate the watermark. The key becomes

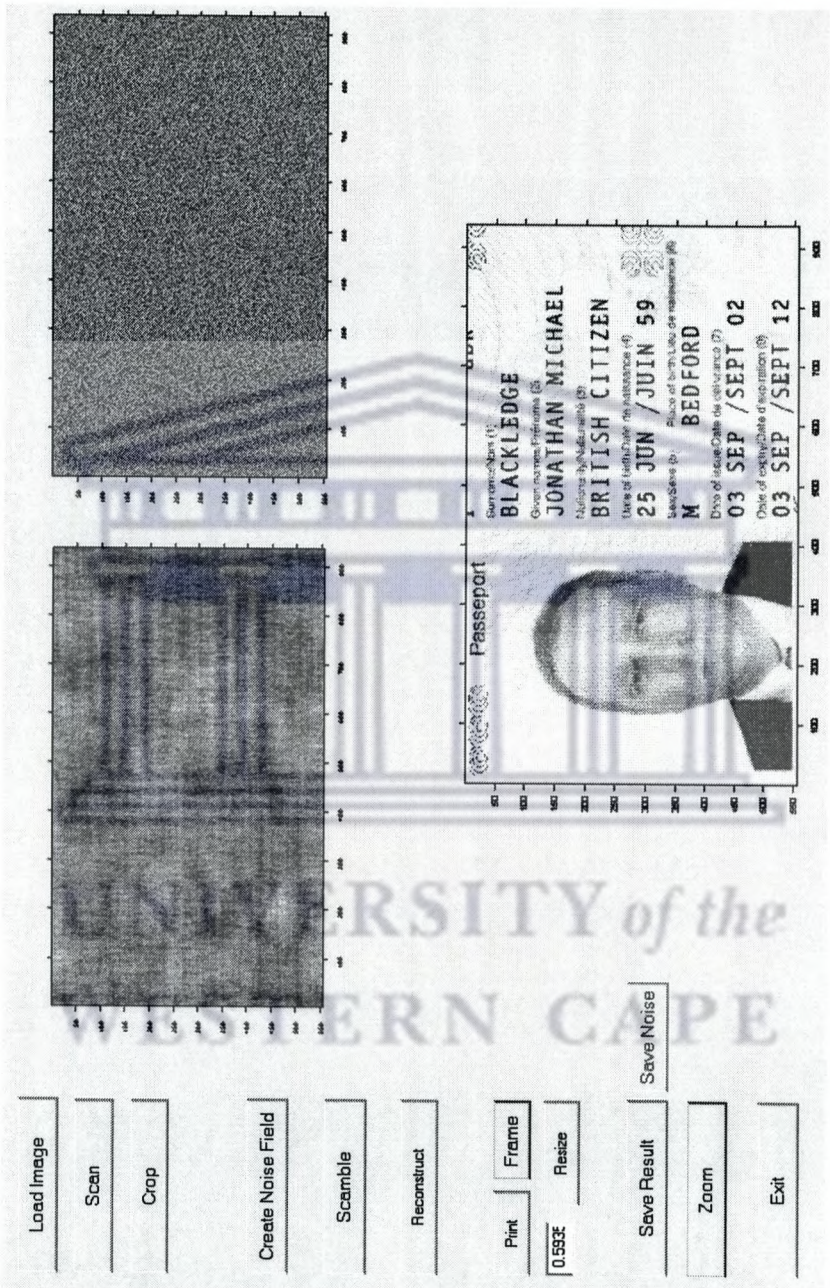


Figure 5.15: Example of passport authentication system (GUI) and a decrypt of the texture code (top-left image) generated by diffusing the cipher (top-right image) with the passport holders details revealed in the decrypt as shown (bottom image). Note that this figure is grey scale images of a colour GUI and colour images.

the 'weakest link'. In order to decipher or decode the hidden data it is necessary to know the secret key [195]. Once the secret key is known, the watermark can be not only decoded, but also easily estimated and removed from the content. In such a state of affairs, a decoder used for copy protection has to be either put into operation as a tamper-proof device, or located in a trusted third party [196]. These solutions are expensive and not necessarily plausible.

We used ICA to separate our 2D DataGlyph from the watermark itself. Of interest is the fact that the 2D DataGlyph plays the role of a watermark and is created to mislead the attacker who would try to crack the watermark typically as a watermark is normally broken. This is done in the view that an attacker, not realizing that a different element had been introduced, would be foiled.

We embedded a 2D Dataglyph as a watermark signature into a watermark. In case of symmetric watermarks, the management of the keys is another challenge. This problem is rectified by ensuring that a watermarker implements the key and only they alone know the key. The 2D DataGlyph will have its own key, whist the image maker will have the original image. Each will be separate and not shared amongst the others. The chances of all three keys being found are then minimal. Within these parameters we used ICA to see if it is possible to break the watermark.

The ICA technique extracts linearly independent distinctive parts from a data-set. Unlike decorrelating data methods, ICA searches for directions in data-space which are independent across all statistical orders [197]. ICA is capable of finding the underlying factors or sources when classic methods fail completely [198]. The approach reported here has the advantage of including a learning probabilistic model (as opposed to projections in data-space), while remaining computationally efficient in high dimensions.

Independent Component Analysis is a computational technique for unraveling a multivariate signal into additive subcomponents under the assumption of mutual statistical independence of non-Gaussian source signals. It is a special case of blind source separation [200], [201] and [202]. ICA is a method of utilizing the process of computer power and statistical techniques for illuminating secret influences which are embedded within sets of random variables, measurements, or signals [197], [203]. ICA has the ability to examine data from various kinds of application fields, as well as digital images, document databases, economic indicators and psychometric measurements [198]. In numerous applications, the dimensions are rendered as a set of parallel signals or time series; the expression blind source separation is employed to differentiate this problem. Common examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process [199]. Here, we used the ICA to see if it is be able to

separate a 2D Dataglyph from a Watermark using the Delome model [204].

The data variables are linear mixtures of some unfamiliar latent variables, and the mixing system is also unfamiliar [204]. The hidden variables are non-Gaussian and mutually independent. They are called the independent components of the observed data [204]. The sources are mixed and then separated into two sources. Delorme [204] uses the time series of two independent sources A and B. Both sources are linearly mixed, i.e. added (or subtracted) to produce a linear combination of the form $aA + bB$ where a and b are assigned coefficients. The ICA algorithm recovers the original sources A and B. Of import is the fact that the algorithm cannot recover the exact amplitude of the sources. Note that ICA can only extract sources that are combined linearly [199].

The focus of ICA is that it bestows unsupported grouping of multimedia data that has shown to be well-aligned with manual grouping [145], [144], [143], [142], [141], [140], [139] and [138] in different media.

5.9 Videos and Images

Extensive research has been contributed in which ICA is used in images and videos. Due to the immensity of the literature in this area we give a conglomeration of applications.

5.9.1 Feature Extraction, Noise Reduction and Natural Scenes

A. Bell and T. Sejnowski [145] and J. H. van Hateren and D. L. Ruderman [142] contemplate basic properties of natural scenes and demonstrate the application of ICA using Gabor-like [174] localized and oriented spatial filters, which resembles receptive fields in the visual cortex. Thus edges are the independent components of natural scenes. P. O. Hoyer and A. Hyvarinen [151] enlarge on these thoughts by extracting features from stereo and color images utilizing Gabor-like spatial filters. A. Hyvarinen [155] uses maximum autocorrelation features which are identical to the Molgedey-Schuster ICA algorithm [175] for remote sensing images and biological shape scrutiny. X. Zhou, B. Moghaddam, and T. S. Huang [154] use ICA to factorize histograms. In order to detect the object within a cluttered space filled scene with other objects, they use joint feature vectors with detection performance being the result of ICA. K. Takaya and K. Y. Choi [156] show dual ICA applications for distinguishing facial components in images contained in a video sequence. There objective is to chart and identify facial features, such as the eyes and nose, onto a 3D wireframe model which are used in face animation. A. Hyvarinen, [155] uses a sparse code shrinkage algorithm and combines ideas of ICA with wavelet shrinkage to show different data techniques for noise modulation

in images. This feature improves on the various approaches such as the Wiener filter, for example.

5.9.2 Detection of Watermarks

Watermark recognition and removal is essential for verification of multimedia data particularly when disseminated over the Internet. S. Noel and H. Szu [157] including F. Sattar D. Yu and K. K. Ma [158] have used ICA for uncovering and recognizing watermarks which appear robust to major image processing attacks.

5.9.3 Content Based Image Retrieval

J. Eakins and M. Graham [176] state that image recovery, which is content driven, is a difficult feature of multimedia investigation. Their reasoning is based on the limited comprehension which exists regarding the narrow understanding of the interactions of image features and abstract content descriptions, i.e. it is difficult to explain content in terms of intensity, edges and texture. The examples they use is Google, FAST Multimedia Search engines, IBM's QBIC system [177], the VIR Image Engine from Virage, Inc. [178] and Visual Retrieval Ware product by Excalibur Technologies [179]. The foundation of repossessing or salvaging images and text on these examples are based on image analysis. These techniques, as well as the research prototypes discussed [180], [176], utilize immature image features for recovery. Of interest is the fact that associated keywords and adjacent text is used to search for an image. To conduct in depth data mining, more context text based tactics needs to be implemented by using latent semantic indexing [160], [161] using statistical tools such as the vector space approach and ICA extensions [138], [162].

5.9.4 Brain Data using Multimodal Interaction

T. P. Jung, S. Makeig, M. J. McKeown, A. Bell, T.-W. Lee, and T. Sejnowski [143] state that a method to eradicate artifacts and separate graphics (EEG) and magnetoencephalographic (MEG) is to use ICA. A comparable tactic used by V. D. Calhoun, T. Adali, L. K. Hansen, J. Larsen, and J. J. Pekar[163] is proving invaluable for functional magnetic resonance brain imaging (FMRI) data.

5.10 Audio

Audio processing uses an on the spot mixing model which is disappointing because audio/sound signals are convolved within their individual broadcasting auditory surroundings. K. Torkkola [164] investigates various areas of blind separation of convolved signals. Torkkola focuses on audio signals and related methods. A broad-spectrum benefit is that extra features are needed. For example, S. Deerwester et al's [181] speech signals can have a better effect, or inserting L. C. Parra and C. V. Alvino's [165] statistical independence with geometric source location (beam forming). A major test is to coordinate the amount of audio source separation tasks with the microphones which are usually less than the audio source. A problem is that sound sources for listening purposes are difficult to extract and identify.

5.10.1 Auditory Evaluation

Natural sound signals are scrutinized by A. J. Bell and T. J. Sejnowski[144] and S. A. Abdallah and M. D. Plumbley [141]. The areas they investigate relate to non-Gaussian signals which emanate from radio stations which broadcast speech and classical music. There output produces wavelets with standard time rate of recurrence which is comparable to a human auditory structure. Multiple-cause neural networks are used by J. Klingseisen et al [187].

5.10.2 Source Separation

ICA can be used to separate musical instruments using Saund's method [183]. Automated music dictation is investigated by M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler [188] with the purpose of identifying instruments and notes and recording the information in writing. The assimilation matrix holds each shape of every musical note and the data of sparse coding together with Saund's multiple cause method is used [183]. S. Dubnov and A. Ben-Shalom [189] also look at the corresponding sound effects needed to edit music notation and use MPEG-7 applications to hunt for similar patterns [146]. The Markov method in spectra arrays in sound media retrieval are investigated together with ICA. M. Casey and A. Westner [166] use an Independent Subspace Analysis (ISA) to take apart sounds from mixed channels. The ISA is an extention of ICA and works by extrapolating a single channel onto a high-dimensional manifold. Dynamic component channels all volatile source signals and they are able to separate speech and audio patterns.

5.11 Text

Vector Space Methods (VSM) and Latent Semantic Analysis (LSA) handle text. The various ICA methods used in this arena search for various subjects [147], [167], [168]. Molegedey-Schuster investigate [175] techniques and divides and sorts dynamic text data [139]. The CNN.com Internet chat room text is one such area using this method. A. Kaban and M. Girolami[184] use the Markov method which is almost the same. The complexity pursuit is considered in [185] which is another method of similar value.

5.11.1 Document Recovery and Hybrid-Language

To search for hybrid-language (Gaelic/English) documents in a semantic space J. Klingseisen and M. D. Plumbley[187] use an extension of the kernel ICA [186] of F. Bach and M. I. Jordan. The recovery rate is highly improved.

5.11.2 Audio-Visual Separation

Dividing a scene is dealt with by Wang, et al[170]. They use MPEG-7[146] to deal with audio-visual separation scene division. Other algorithms for separation and video archiving are explored. T. Darrell, J. Fisher, P. Viola, and B. Freeman [171] and [172] and J. W. Fisher, T. Darrell, W.T. Freeman, and P. Viola use a similar pattern of data in the audio-visual material to find a common denominator. They compare the lip movements of the speaker and other auditory signals [173]. A joint statistical method is used. P. Smaragdis and M. Casey [148] stream audio/visual data and compare the output.

5.11.3 Assimilated Text/Image Removal

The Vector Space Model [159], [160] and [161] regarding the removal and understanding data from images and the text is another manner for gathering information and extruding it. The essential part of the canonical relationship research using ICA is investigated by T. Kolenda et al [138] and A. Vinokourov et al [162].

5.12 ICA Method

The ICA requires at least n versions of the mixture of n signals in order to be able to reasonably and reliably separate n signals. The method is applied to the process of protecting authenticity of black and white documents. This is intended to demonstrate that this

technology can be effectively used in the detection of fraudulent documents. We show that this method can recover (although not perfectly) the original image, after being masked by watermarking with a 2D DataGlyph. We then show that rescaling of the document does not affect the method, except in dimensional extremes. Finally we show that if the original watermark has been changed (tampered with) there is a low probability of recovering the image.

ICA is applied to recover an original image from a mixture of images. A fast ICA algorithm is used to process an image that requires an $n \times m$ matrix of pixels. Black and white images are processed as an image of $n \times m$ pixels. Colored images are $n \times m \times p$ processed, where p describes the components of colour. Colored images must first be converted into black and white images before the processing can be applied.

The technique is investigated using the following steps:

Step 1: We embed the DataGlyph (on the image that has been watermarked) into a BMP image file.

Step 2: The image is then processed. On completion of this process we obtain an image composed of the following features: an original image, the watermark and the 2D DataGlyph.

Step 3. We then apply ICA to recover the original image from the mixed image in step two.

Figure 5.16 shows a typical example of applying these processing steps.

Typical parameters that can be changed are the number of rows or the number of columns of the matrix processed by the ICA, the seed or 'key' used in the generation of the DataGlyph and the key used to in the watermark process. Any change of parameters affects the ICA recovery process.

When trying to measure the recovery rate of using ICA, it is possible to provide a measure. There are two methods to measure the rate of recovery. The first is asymptotic and the second is based on a probablity measure. If we consider that the outcome of using ICA recovers a clear image once a 2D DataGlyph and a Watermark are added then from probability theory we can consider that the probability of reconstructing the original image using ICA is give by the nuber of 'clean images' generated using ICA divided by all possible transformations that are not representative of the original image. The term 'clean images' specifies an image that is distinguishable and is visually recognisable as being the original image after applying ICA. The topic of fuzzy logic can be applied in deciding whether the ICA object is recognizable as the original.

Figure 5.16: Original image (top-left), DataGlyph (bottom-left), texture map composed of the original image (after resizing) and the DataGlyph (top-right), reconstruction (bottom-right)

5.13 Discussion

The ICA model has been used extensively to extract information from various multimedia digital fields of investigation. Using asymmetrical assimilation of both cryptography and watermarking applications including embedding a 2D DataGlyph within the algorithm makes the extraction of data more difficult. We have offered a different method to both robust and fragile watermarking, using ICA and the ability to discover hidden material within the hidden material. The novel line of attack, founded on how data is extruded using the latest data of extraction, namely ICA, shows a high rate of data removal. Our combination of watermark and 2D DataGlyph was not able to be successfully broken and the rate of robustness is positive. We have examined its performance on a set of representative images, random messages and various attacks, and experiments show promising performance on all the attacks examined. The main advantage of our approach is that, being based on embedding information using statistically independent sources, the watermarking held up against the latest method of data removal used. Based on local information and a linear transform, our method is computationally efficient, and offers additional security in the use of mixing and demixing matrices that are not easy to obtain. Further research may improve the performance by refining existing statistical models or identifying a new approach for selecting the IC's to be watermarked or the distortion measures used (for instance a measure based on the human visual system).

Chapter 6

Conclusions and Further Development

6.1 Introduction

In this thesis, we have developed a robust watermarking method for paper security. Unlike traditional watermarking techniques, in this approach we can extract the hidden watermark after a print/scan attack which is achieved by using the convolution and correlation processes for coding and decoding respectively. This approach is chosen because of its compatibility with the principles of physical optics involved in scanning of a document. The system is secure in that it can not be attacked easily. First, the feature is not 'suspicious' as many documents have a background texture. Second, the attacker does not know the algorithm used to generate the diffused watermark. Even if the attacker does know the algorithm, he/she must still know a significant amount of information before the system can be broken, such as: the correct key, the diffusion operator type, the original image size.

6.2 Conclusions

Valuable paper documents are subject to misuse by criminals. This is largely due to the dramatic improvement in personal computer hardware and peripheral equipment. Embedding watermarks into a printed document is one way to secure them. The ability to extract the watermark from a printed copy is generally useful to help establish ownership, authenticity, and to establish the origin of an unauthorized disclosure. However, finding a robust watermarking technique is a continuing challenge. This is due to extensive amount of noise that is added when a document is printed and scanned. Moreover, printed documents do not maintain their quality over time.

The basic approach developed for this thesis is compounded in the following. The wa-

termark w is diffused (convolved) with a noise field n and placed into the background of a covertext, typically a text document. The watermark is extracted by removing the covertext f using a modified median filter. We then correlate the diffused watermark with the original noise field. This process (i.e. coding and decoding) is compounded in the following formula:

$$\begin{aligned} w' &= \mathcal{F}\left\{ \overbrace{\left(\overbrace{\left(\overbrace{A_w e^{i\theta_w} A_n e^{i\theta_n}}^{\text{Diffusion}} + f \right)}^{\text{Confusion}} - f \right) A_n e^{-i\theta_n}}^{\text{Covertext removal}} \right\}^{\text{Correlation}} \\ &= \mathcal{F}^{-1}\left\{ A_w e^{i\theta_w} |A_n|^2 \right\} \end{aligned}$$

where w' is the extracted watermark, A_w and A_n are the amplitude spectra of the watermark and cipher respectively, θ_w and θ_n are the respective phase spectra. The extracted watermark is a noisy version of the embedded watermark. This noise is due to the power spectrum $| A_n |^2$. In order to enhance the extracted watermark, we have to eliminate the power spectrum term or at least minimize its effect. One way to do this is to divide the output over the power spectrum during the diffusion step or correlation step. To avoid singularities, we replace each and all zero in $| A_n |^2$ by 1. Alternatively, we can choose n such that it has a homogeneously distributed power spectrum across all frequencies, (such as white noise) or pre-process n by replacing it's amplitude spectrum with a constant value. However, these conditions are restrictive and the regularisation method discussed above to avoid singularities is both simple and effective.

The method is robust to a wide variety of attacks including geometric attacks, drawing, crumpling and print/scan attacks. The method is relatively insensitive to lossy compression, filtering, amplitude adjustments, additive noise and thresholding. The principal weakness of the system is its sensitivity to rotation and cropping. This can be minimized by orienting the document correctly and accurately before scanning and using automatic cropping software which is available with selected scanners (e.g. Cannon scanners). Alternatively, introduction of a frame provides a reference feature from which an accurate crop can be obtained.

The visibility of the diffused watermark and the compatibility of this system with the physical principles of an imaging system, increase the robustness of the system and provided a successful approach to the extraction of the watermark after scanning at low resolution. Moreover, using correlation in the extraction phase increases the robustness of the system to some important attacks such as translation and cropping (most likely to occur during a scan).

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

6.3 Covert Encryption using Digital Image Steganography

In terms of further developments, the principles discussed in this thesis can be used to design an entirely covert encryption system. By inputting any encrypted file as binary data, we can generate a binary image (consisting entirely of pixels with values of 0 or 1). For example, consider the plaintext *Cryptology* which is encrypted to provide the ciphertext string *ydr39bkLP9* and is equivalent to the 7-bit ASCII bit stream

111100111001001110010011001101110011100010110101110011001101000000111001

This bit stream is converted into the 9×9 square image[1] with zero padding being used to complete the array as given below:

$$\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

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

[1]The image does not necessarily need to be square and is used here for illustrative purposes only

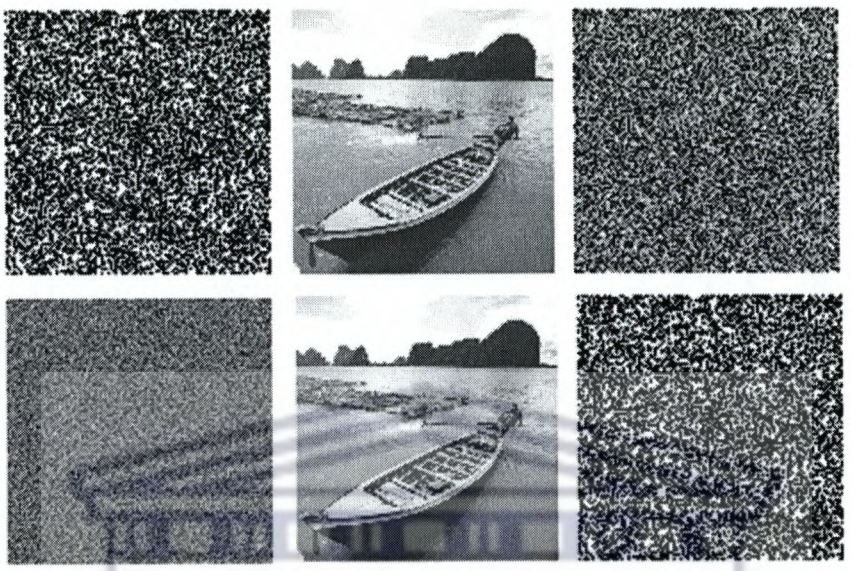


Figure 6.1: From top to bottom and from left to right (all images are 512×512): Binary image of ciphertext, covertext (digital image), cipher, diffused image, stegotext after addition of the diffused data (for a confusion-to-diffusion ratio of $r = 0.01$), reconstruction.

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

An example of using this approach is illustrated in Figure 6.1. The 'watermark' (top-left) is a 512×512 binary image obtained from the ciphertext of a 6000 word document after encryption with 7-bit ASCII binary conversion. The reconstruction is a bit-for-bit replica of the input and can thus, be decrypted without error. The binary image is first converted back into a bit stream (upto the point beyond which padding is applied) and each consecutive 7-bit block, converted back into the ciphertext which is then decrypted.

In order to enhance the cryptographic strength associated with approach, the cipher shown in Figure 6.1 can be obtained from a genuine random number generator such as HotBits (http://www.fourmilab.ch/hotbits/) and then encrypted (to secure the data file) using a specified cryptosystem. Cleary, in addition to the receiver of the stegotext requiring the facility to decrypt the reconstruction, in order to obtain this reconstruction, the receiver

must have the cipher and the covertext. The covertext should be one of a database of images maintained by both parties together with the cipher that is ideally stored in encrypted form. Because the stegotext and covertext images look identical, the receiver can search through the image data base to select the appropriate covertext. The whole point of this process is that it provides a way of camouflaging the encrypted data during transmission, the difference between sending the ciphertext and the stegotext being illustrated in Figure 6.1 as digital images. However, in this process, a macro-key is required to be exchanged which is composed of the following:

- the cipher
- the covertext database
- the decryption system

A covertext database is required for two reasons: (i) each time a transmission is undertaken, it is safer to transmit a different stegotext in order not to alert a potential attacker to multiple transmissions of the same data; (ii) a database of images should be stored rather than a single image in order that no apparent significance is given to a single image should the platform (i.e. PC or USB stick, for example) be compromised.

References

[1] Singh S, *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots to Quantum Cryptography*, Doubleday, 1999.

[2] Schneier B, *Beyond Fear: Digital Security in a Networked World*, Wiley, 2000.

[3] Schneier B, *Thinking Sensibly about Security in an Uncertain World*, Copernicus Books, 2003.

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

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

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

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

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

[9] Hershey J, *Cryptography Demystified*, McGraw-Hill, 2003.

[10] Gaines H F, *Cryptanalysis*, Dover, 1939.

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

[12] http://vl.fmnet.info/safety/

[13] http://www.amazon.com/Network-Security-process-not-product

[14] http://en.wikipedia.org/wiki/Enigma_Machine

[15] Author(s): von Neumann J and Morgenstern O, *Theory of Games and Economic Behaviour*, Princeton University Press, 1944.

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

[17] Briggs J, *Fractals: The Patterns of Chaos (Discovering a New Aesthetic of Art, Science, and Nature)*, Touchstone, 1992.

[18] Hacker's Black Book, http://www.hackersbook.com

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

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

[21] Kipper G, *Investigators Guide to Steganography*, CRC Press, 2004.

[22] Shulsky A N and Schmitt G J, *Silent Warefare: Understanding the World of Intelligence*, Brassey, 2002.

[23] Hough R, *The Great War at Sea*, Oxford University Press, 1983

[24] Halpern P G, *A Naval History of World War One*, Routledge, 1994.

[25] Ratcliff R A, *Delusions of Intelligence*, Cambridge University Press, 2006.

[26] R A Woytak, *On the Boarder of War and Peace: Polish Intelligence and Diplomacy and the Origins of the Ultra-Secret*, Columbia University Press, 1979.

[27] Kozaczuk W *Enigma: How the German Machine Cipher was Broken, and how it was Read by the Allies in World War Two*, University Publications of America, 1984.

[28] Booss-Bavnbek B and Hoyrup J, *Mathematics at War*, Birkhäuser, 2003.

[29] Copelend B J, *Colossus: The Secrets of Bletchley Parks Code Breaking Computers*, Oxford University Press, 2006.

[30] Stripp A and Hinsley F H, *Codebreakers: The Inside Story of Bletchley Park*, Oxford University Press, 2001

[31] http://www.gchq.gov.uk/

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

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

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

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

[36] http://eprint.iacr.org/1996/002.

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

[38] Delfs H and Knebl H, *Introduction to Cryptography: Principles and Applications*, Springer, 2002.

[39] Ashchenko V V, Jascenko V V, and Lando S K, *Cryptography: An Introduction*, American Mathematical Society, 2002.

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

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

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

[43] Garrett P, *Making, Braking Codes*, Prentice Hall, 2001.

[44] Reynolds P, *Breaking Codes: An Impossible Task?*, 2004; http://news.bbc.co.uk/1/hi/technology/3804895.stm.

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

[46] Blackledge J M, Foxon B and Mikhailov S, *Fractal Dimension Segmentation*, Proceedings of the First IMA Conference on Image Processing: Mathematical Methods and Applications (Ed. J M Blackledge), Oxford University Press, 249-292, 1997.

[47] Blackledge J M, Foxon B and Mikhailov S, *Fractal Coding Techniques*, European Military Communications Conference, Nice, November 26-33, 1996.

[48] Blackledge J M, Foxon, B and Mikhailov S, *Fractal Modulation Techniques for Digital Communications Systems*, Proceedings of IEEE Conference on Military Communications, Boston, USA, 1998.

[49] Blackledge J M, London M, Mikhailov S and Smith R, *On the Statistics of Dimension: Fractal Modulation and Quantum Fractional Dynamics*, Proceedings of the Second IMA Conference on Image Processing: Mathematical Methods, Algorithms and Applications (Eds. J M Blackledge and M Turner), 184-227, 2000.

[50] Blackledge J M and Turner M J, *Analysis of the Limitations of Fractal Dimension Texture Segmentation for Image Characterization*, 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, 114-137, 2002.

[51] Mahmoud K W, *Low Resolution Watermarking for Print Security*, PhD Thesis, Loughborough University, 2004.

[52] Mahmoud K W, Blackledge J M, Datta S and Flint J, *Print Protection using High Frequency Fractal Noise*, Security, Steganography and Watermarking of Media Contents VI (Eds. E J Delp and P W Wong), Proc. SPIE-IS& T Electronic Imaging, SPIE **5306**, 446-454, 2004.

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

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

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

[56] F. Petitcolas, A. R, and M. Kuhn, "Information hiding: A survey," *IEEE* **87**(7), pp. 1062–1077, 1999.

[57] B. Pfitzmann, "Information hiding terminology," in *First International Workshop on Information Hiding*, pp. 347–350, 1996.

[58] K. Mahmoud, *Low Resolution Watermarking for Print Security*, PhD Thesis, loughborough University, 2004.

[59] I. J. Cox, M. Miller, and J. Bloom, "A review of watermarking and the importance of perceptual modeling," *SPIE, Human Vision and Electronic Imaging* **3016**(2), pp. 92–99, 1997.

[60] S. Craver, B. Yeo, and M. Yeung, "Technical trails and legal tribulations," *Communications of the ACM* **41**, pp. 44–54, July 1998.

[61] E. Ferril and M. Moyer, "A survey of digital watermarking," Feb 1999. http://elizabeth.fer

[62] F. Mintzer, J. Lotspiech, and N. Morimoto, "Safeguarding digital library contents and users," *D-Lib Magazine* , December 1997.

[63] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in *Information Hiding: Second International Workshop*, **1525**, pp. 218–238.

[64] Hannigan, T.Brett, Reed, Alastair, Bradley, and A. Brett, "Digital watermarking using improved human visual system model," in *Security and Watermarking of Multimedia Contents III*, P. W. Wong and E. J. Delp, eds., **4314**, pp. 468–474, SPIE, Aug 2001.

[65] M. Kankanhalli and R. Ramakrishnan, "Content based watermarking of images," in *6th ACM International Multimedia Conference*, pp. 61–70, (Bristol, England), Sep 1998.

[66] J. Johnston, N. Jayant, and R. Safranek, "Signal compression based on models of human perception," in *Proceedings of the IEEE*, **81**, pp. 1385–1422, Oct 1993.

[67] A. M. Alattar, "Smart images using digimarc's watermarking technology," in *Proc. Of SPIE Electronic Imaging'00, Security and Watermarking of Multimedia Contents*, **3971**(25), 2000.

[68] Lumini D M A, 2000,*A Wavelet-based Image Watermarking Scheme*, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '00), IEEE, 122-127.

[69] Kundur D and Hatzinakos D, 1997, A Robust Digital Image Watermarking Method using Wavelet based Fusion, Proceedings of the International Conference on Image Processing (ICAP '97), IEEE, 544-547.

[70] Tassignon H, *Wavelets in Image Processing*, Image Processing II:Mathematical Methods, Algorithms and Applications (Eds. J M Blackledge and M J Turner), Horwood Publishing, 2000.

[71] Cox J I, Miller M L and Bllom J A, *Digital Watermarking*, Morgan Kaufmann Publishers, Academic Press, 2002.

[72] Katzenbaiser S and Petitcolas F A P, *Informatiom Hiding Techniques for Steganography and Digital Watermarking*, Artech, 2000

[73] *Maximum Entropy in Action*, (Eds. B Buck and V A Macaulay), Oxford Science Publications, 1991.

[74] Anderson R J and Petitcolas F A P, 1998, *On the Limits of Steganography*, IEEE Journal of Selected Areas in Communication (Special issue on Copyright and Privicy Protection), 16(4), 474-481.

[75] Petitcolas F A P, Anderson R J and Kuhn M G, 1999 *Information Hiding - A Survey*, Proc. IEEE, 87(7), 1062-1078.

[76] Jazinski A, *Stochastic Processes and Filtering Theory*, Academic Press, 1970.

[77] Papoulis A, *Signal Analysis*, McGraw-Hill, 1977.

[78] Bateman A and Yates W, *Digital Signal Processing Design*, Pitman, 1988.

[79] Rihaczek A W, *Principles of High Resolution Radar*, McGraw-Hill, 1969.

[80] Mitchell R L, *Radar Signal Simulation*, Mark Resources Incorporated, 1985.

[81] Kovaly J J, *Synthetic Aperture Radar*, Artech, 1976.

[82] *Cryptography and Coding*, (Ed. M Darnell), Lecture Notes in Computer Science (1355), Springer, 1997.

[83] Kundur D and Hatzinakos D, 1997, Digital Watermarking using Multi-resolution Wavelet Decomposition, Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '98), IEEE, 2969-2972.

[84] Chae J J and Manjunath B, 1999, *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.

[85] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," Tech. Rep., MIT Media Lab, 1994.

[86] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in Information Hiding, pp. 293–315, 1996.

[87] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in IEEE Int. Conf. on Multimedia Computing and Systems, pp. 473–480, Hiroshima, Japan, June 1996.

[88] M. D. Swanson, B. Zhu, A. H. Tewk, and L. Boney, "Robust audio watermarking using perceptual masking", Signal Processing 66, pp. 337–355, 1998.

[89] Yin Xiong and Zhang Xiao Ming, "Covert Communication Audio Watermarking Algorithm Based on LSB", International Conference on Communication Technology, 2006. ICCT '06. Nov. 2006 pp.1- 4.

[90] Byeong Seob Ko, R. Nishimura and Y. Suzuki, "Time-spread echo method for digital audio watermarking",IEEE Trans. on Multimedia Vol. 7, Issue 2, 2005 pp. 212-221.

[91] H. O. Oh, J. W. Seok, J. W. Hong and D. H. Youn, "New echo embedding technique for robust and imperceptible audio watermarking," Proc. ICASSP 2001, pp. 1341-1344.

[92] Yi-Wen Liu and J.O. Smith, "Watermarking sinusoidal audio representations by quantization index modulation in multiple frequencies", Proc. of ICASSP 2004 Vol. 5, pp. 373-376.

[93] Darko Kirovski and Henrique S. Malvar, "Spread-spectrum watermarking of audio signals", IEEE Trans. on Signal Processing, Vol. 51, Issue 4, 2003 pp. 1020-1033.

[94] Li, Lili; Hu, Jianling and Fang, Xiangzhong, "Spread-Spectrum Audio Watermark Robust Against Pitch-Scale Modification",IEEE International Conference on Multimedia and Expo, 2007, pp. 1770-1773.

[95] Ling Xie; Jiashu Zhang and Hongjie He, "Robust Audio Watermarking Scheme Based on Nonuniform Discrete Fourier Transform",IEEE International Conference on Engineering of Intelligent Systems, 2006, pp. 1 – 5.

[96] Sang-Kwang Lee and Yo-Sung Ho, "Digital audio watermarking in the cepstrum domain", IEEE Transactions on Consumer Electronics, Vol. 46, Issue 3, pp. 744-750.

[97] R. Vieru, R. Tahboub, C. Constantinescu and V Lazarescu, "New results using the audio watermarking based on wavelet transform", International Symposium on Signals, Circuits and Systems, 2005. Vol. 2, pp. 441-444.

[98] Xiaomei Quan and Hongbin Zhang, "Audio watermarking based on psychoacoustic model and adaptive wavelet packets",Proc. of 7th Int. Conf. on Signal Processing, 2004 Vol. 3, pp. 2518-2521.

[99] Xiang-Yang Wang and Hong Zhao, "A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT",IEEE Transactions on Signal Processing, Vol. 54, Issue 12, 2006 pp. 4835-4840.

[100] I. Cox, M. Miller and J. Bloom, "Digital Watermarking", Morgan Kaufmann Publishers, placeCitySan Francisco, StateCA, 2003

[101] J. Chou, K. Ramchandran and A.Ortega, "High capacity audio data hiding for noisy channels", Proc. Int. Conf. on Information Technology: Coding and Computing, 2001, pp. 108–111.

[102] N. Cvejic and T. Seppnen, "Fusing digital audio watermarking and authentication in diverse signal domains",Proc. European Signal Processing Conference, placeCityAntalya, country-regionTurkey, 2005 pp. 84-87.

[103] S. Mallat, "A Wavelet Tour of Signal Processing", ISBN: 0-12-466606-X; Academic Press, 1999.

[104] P. Kabal, "An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality", Technical Report, McGill University, version 2, 2003.

[105] C. T. Hsu and J. Ling, *Hidden digital watermarks in images*, IEEE Transactions on Image Processing, Vol. 8, 58-68, 1999.

[106] J. Zaho and E. Koch, *Embedding robust labels into images for copyright protection*, Proceedings of the International Conference on Intellectual Property Rights for Information, Knowledge and New Techniques, 242-251, (München, Wien: Oldenbourg Verlag), 1995.

[107] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, *A secure, robust watermark for multimedia*, First International Workshop on Information Hiding, Ed. R. Anderson, 1174 of Lecture Notes in Computer Science, 183-206, Springer-Verlag, 1996.

[108] M. Barni, F. Bartolini, V. Cappellini, and A.Piva, *A DCT-domain system for robust image watermarking*, Signal Processing (EURASIP), Vol. 66, 357-372, 1998.

[109] J. J. K. O. Ruanaidh, W. J. Dowling, and F. M. Boland, *Watermarking digital images for copyright protection*, IEE Proceedings on Vision, Signal and Image Processing, Vol. 143, 250-256, 1996.

[110] M. D. Swanson, B.Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *International Conference on Image Processing, IEEE*, **3**, pp. 211–214, 1996.

[111] J. J. Chae and B. Manjunath, "A technique for image data hiding and reconstruction without host image," in *Proc. Of SPIE Electronic Imaging'99, Security and Watermarking of Multimedia Contents*, Wong and Delp, eds., **3657**, (San Jose, California), January 1999.

[112] A. Bors and I. Pitas, "Image watermarking using block site selection and d.c.t. domain constraints," *Optics Express* **3**, pp. 512–523, Dec 1998.

[113] B. Tao and B. Dickenson, "Adaptive watermarking in the DCT domain," in *International Conference on Acoustics and Signal Processing*, 1997.

[114] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New Jersey, 2nd ed., 2002.

[115] X. Xia, C. Boncelet, and G. Arce, "A multi-resolution watermark for digital images," in *Proc. IEEE Int. Conf. On Image Processing*, **1**, pp. 548–551, Oct 1997.

[116] A. Lumini and D. Maio, "A wavelet-based image watermarking scheme," in *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, pp. 122–127, March 2000.

[117] D. Kundur and D. Hatzinakos, "Digital watermarking using multi-resolution wavelet decomposition," in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing*, **6**, pp. 2969–2972, 1998.

[118] D. Kundur and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," in *In International Conference on Image Processing, IEEE*, pp. 544–547, (Santa Barbara, California, U.S.A.), Oct 1997.

[119] M. D. Levine, *Vision in Man and Machine*, McGraw-Hill, Toronto, 1985.

[120] J. Ohnishi and K. Matsui, "Embedding a seal into a picture under orthogonal wavelet transform," in *Proc. Int. Conference on Multimedia Computing and Systems*, pp. 514–521, June 1996.

[121] M. Barni, F.Bartolini, V.Cappellini, A. Lippi, and A. Piva, "A DWT-based technique for spatio-frequency masking of digital signatures," in *Proc. Of SPIE Electronic Imaging'99, Security and Watermarking of Multimedia Contents*, **3657**, pp. 31–39, (San Jose, California), January 1999.

[122] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE trans. Image Processing* **1**, pp. 240–250, April 1992.

[123] H. Inoue, T. Katsura, A. Miyazaki, and A. Yamamoto, "A digital watermark technique based on the wavelet transform and its robustness on image compression and transformation," *IEICE Transactions on Fundamentals of Electronics* **E82-A**, pp. 2–10, Jan 1999.

[124] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing* **41**(12), pp. 3445–3462, 1993.

[125] J. J. K. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing(EURASIP)* **66**, pp. 303–317, May 1998.

[126] J. J. K. O. Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proceedings of the IEEE International Conference on Image Processing*, **3**, pp. 239–242, sep 1996.

[127] V. Solachidis and I. Pitas, "Circularly symmetric watermark embedding in 2-d DFT domain," in *International Conference on Acoustics, Speech and Signal Processing, IEEE Signal Processing Society*, pp. 1563–1565, (Phoenix, Arizona, USA), March 1999.

[128] W. Kim, J. Lee, and W. Lee, "An image watermarking scheme with hidden signature," in *IEEE Proceeding of the International Conference on Image Processing*, pp. 206–210, (Japan), Oct 1999.

[129] H. Raymound, F. Chan, and K. Yeung, "A frequency domain watermarking scheme," Jan 2001.

[130] A. Herrigel, J. O'Ruanaidh, H. Petersen, and S. Pererira, "Secure copyright protection techniques for digital images," in *Information Hiding of Lecture Notes in Computer Science,*, D. Aucsmith, ed., **1525**, pp. 169–190, Springer-Verlag, 1998.

[131] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient public watermarking for images," in *Security and Watermarking of Multimedia Contents II, Proceedings of SPIE*, P. W. Wong and J. E. Delp, eds., **3971**, pp. 90–98, 2000.

[132] M. Kankanhalli and R. Ramakrishnan, "Content based watermarking of images," in *6th ACM International Multimedia Conference*, pp. 61–70, (Bristol, England), Sep 1998.

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

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

[135] M. Kutter and F. Hartung, "Introduction to watermarking techniques," in *Information Hiding: Techniques for Stegangraphy and Difgital Watermarking*, S. Katezenbeisser and F. A. P. Petitcolas, eds., ch. 5, pp. 97–120, Artech House, Boston, 2000.

[136] A. K. Jain, S. Pankanti, and R. Bolle, eds., *BIOMETRICS: Personal Identification in Networked Society*, (Kluwer), 1999.

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

[138] T. Kolenda, L. K. Hansen, J. Larsen, and O. Winther, Independent component analysis for understanding multimedia content," in Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII, H. Bourlard, T. Adali, S. Bengio, J. Larsen, and S. Douglas, Eds., Matigny, Valais, Switzerland, Sep. 4-6 2002, pp. 757-766, IEEE Press.

[139] T. Kolenda, L.K. Hansen, and J. Larsen, "Signal detection using ICA: Application to chat room topic spotting," in Proceedings of ICA'2001, T.-W. Lee, T.-P. Jung, S. Makeig, and T. Sejnowski, Eds., San Diego, USA, December 2001, pp. 540 -545.

[140] T. Kolenda, L.K. Hansen, and S. Sigurdsson, "Independent components in text," in Advances in Independent Component Analysis, M. Girolami, Ed., pp. 229-250. Springer-Verlag, 2000.

[141] S. A. Abdallah and M. D. Plumbley, "If edges are the independent components of natural images, what are the independent components of natural sounds?," in

Proceedings of ICA'2001, T.-W. Lee, T.-P. Jung, S. Makeig, and T. Sejnowski, Eds., San Diego, USA, December 2001, pp. 534-539.

[142] J. H. van Hateren and D. L. Ruderman, "Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex," in Proceedings of the Royal Society of London - B - Biological Sciences, 1998, vol. 265,

[143] T.-P. Jung, S. Makeig, M. J. McKeown, A. Bell, T.-W. Lee, and T. Sejnowski, "Imaging brain dynamics using independent component analysis," IEEE Proceedings, vol. 89, no. 7, pp. 1107-1122, 2001.

[144] A. J. Bell and T. J. Sejnowski, "Learning the higher order structure of a natural sound," Network: Computation in Neural Systems, vol. 7, no. 2, pp. 261-267, 1996.

[145] A. Bell and T. Sejnowski, "Edges are the independent components of natural scenes," in Advances in Neural Information Processing Systems, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. 1997, vol. 9, pp. 831-837, The MIT Press.

[146] J.M. Martinez, "Overview of the MPEG-7 standard (version 5.0)," Tech. Rep., ISO, Coding of moving pictures and audio, 2001.

[147] L. K. Hansen, J. Larsen, and T. Kolenda, "On independent component analysis for multimedia signals," in Multimedia Image and Video Processing, L. Guan, S.-Y. Kung, and J. Larsen, Eds., pp. 175- 199. CRC Press, Sep. 2000.

[148] P. Smaragdis and M. Casey, "Audio/visual independent components," in Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, Nara, Japan, April 2003

[149] K. V. Mardia, J. T. Kent, and J. M. Bibby, Multivariate Analysis, Academic Press Ltd., 1979.

[150] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini, "Inferring a semantic representation of text via cross-language correlation analysis," in Advances in Neural Information Processing Systems, S. Becker, S. Thrun, and K. Obermayer, Eds. 2003, vol. 15, The MIT Press.

[151] P. O. Hoyer and A. Hyvarinen, "Independent component analysis applied to feature extraction from colour and stereo images," Computation in Neural Systems, vol. 11, no. 3, pp. 191-210, 2000.

[152] J. Hurri and A. Hyvarinen, "Simple-cell-like receptive fields maximize temporal coherence in natural video," Neural Computation, 2003.

[153] R. Larsen, "Decomposition using maximum autocorrelation factors," Journal of Chemometrics, vol. 16, no. 8-10, pp. 427-435, 2002.

[154] X. Zhou, B. Moghaddam, and T. S. Huang, "ICA-based probabilistic local appearance models," in Proceedings of International Conference on Image Processing (ICIP'01), October 2001.

[155] A. Hyvarinen, "Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation," Neural Computation, vol. 11, no. 7, pp. 1739-1768, 1999.

[156] K. Takaya and K.-Y. Choi, "Detection of facial components in a video sequence by independent component analysis, "in Proceedings of ICA 2001, T.P. Jung, S. Makeig. and T. Sejnowsji. Eds., San Diego, USA, December 2001, pp. 260-265.

[157] S. Noel and H. Szu, "Multimedia authenticity with independent-component watermarks," in 14th Annual International Symposium on Aerospace / Defense Sensing Simulation, and Controls, Orlando, Florida, April 2000.

[158] F. Sattar D. Yu and K. K. Ma, "Watermark detection and extraction using independent component analysis method," EURASIP Journal on Applied Signal Processing, vol. 2002, no. 1, pp. 92-104, 2002.

[159] M. La Cascia, S. Sethi, and S. Sclaroff, "Combining textual and visual cues for content based image retrieval on the world wide web," in IEEE Workshop on Content Based Access of Image and Video Libraries. 1998, pp. 24-28, IEEE Computer Society.

[160] Z. Pecenovi, "Image retrieval using latent semantic indexing," M.S. thesis, AudioVisual Communications Lab, Ecole Polytechnique Federale de Lausanne, Switzerland, 1997.

[161] T. Westerveld, "Image retrieval: Content versus context," in Proceedings Content Based Multimedia Information Access, RIAO 2000, Paris, France, 2000, pp. 276-284.

[162] A. Vinokourov, D. Harddon, and J. Shawe-Taylor, "Learning the semantics of multimedia content with application to web image retrieval and classification," in Proceedings of the International Work- shop on Independent Component Analysis and Blind Signal Separation, Nara, Japan, April 2003.

[163] V. D. Calhoun, T. Adali, L. K. Hansen, J. Larsen, and J. J. Pekar, ICA of functional MRI data: An overview," in Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, Nara, Japan, April 2003.

[164] K. Torkkola, "Blind separation for audio signals" in Proceedings of ICA'99, Aussois, France, January 1999, pp. 239- 244.

[165] L. C. Parra and C. V. Alvino, "Geometric source separation: Merging convolutive source separation with geometric beam forming," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 6, pp. 352-362, 2002.

[166] M. Casey and A. Westner, "Separation of mixed audio sources by independent subspace analysis," in Proceedings of the International Computer Music Conference, ICMA, Berlin, August 2000.

[167] C. L. Isbell, Jr. and P. Viola, "Restructuring sparse high dimensional data for effective retrieval," in Advances in Neural Information Processing Systems, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds., 199, vol. 11, pp. 480-486.

[168] F. A. Nielsen and L. K. Hansen, "Author cocitation analysis of articles from "NeuroImage"," NeuroImage, vol. 13, no. 6, part 2, pp. S212, June 2001.

[169] D. Cohn and T. Hofmann, "The missing link - a probabilistic model of document content and hypertext connectivity," in Advances in Neural Information Processing Systems, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. 2001, vol. 13, pp. 430-436, MIT Press.

[170] Y. Wang, Z. Liu, and J.C. Huang, "Multimedia content analysis-using both audio and visual clues," IEEE Signal Processing Magazine, vol. 17, no. 6, pp. 12-36, 2000.

[171] T. Darrell, J. Fisher, P. Viola, and B. Freeman, Mutation and the cocktail party effect," in Proceedings International Conference Advances in Multimodal Interfaces (ICMI), October 2000, vol. 1948, pp. 32-40.

[172] J. W. Fisher III, T. Darrell, W.T. Freeman, and P. Viola, "Learn- ing joint statistical models for audio-visual fusion and segregation," in Advances in Neural Information Processing Systems, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. 2001, vol. 13, MIT Press.

[173] D. Sodoyer, J.-L. Schwartz, L. Girin, J. Klinkisch, and C. Jutten, "Separation of audio-visual speech sources: A new approach exploit- ing the audio-visual coherence of speech stimuli," EURASIP JASP, vol. 1, pp. 1165-1173, 2002.

[174] B. MacLennan, "Gabor representations of spatiotemporal visual images," Tech. Rep. CS-91-144, Computer Science, Univ. of Tennessee, 1994.

[175] L. Molgedey and H. Schuster, "Separation of independent signals using time-delayed correlations," Physical Review Letters, vol. 72, no. 23, pp. 3634-3637, 1994.

[176] J. Eakins and M. Graham, "Content based image retrieval," Tech. Rep., University of Northumbria at Newcastle, 1999.

[177] M. Flickner et al., "Query by image and video content IEEE Computer, vol. 28, no. 9, pp. 23-32, 1995.

[178] A. Gupta et al., "The virage image search engine: an open framework for image management," Storage and Retrieval for Image and Video Databases IV, vol. Proceedings SPIE 2670, pp. 76-87, 1996.

[179] J. Feder, "Towards image content based retrieval for the world-wide web," Advanced Imaging, vol. 11, no. 1, pp. 26-29, 1996.

[180] J. Eakins, "Towards intelligent image retrieval," Pattern Recognition, vol. 35, pp. 3-14, 2002.

[181] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," J. Amer. Soc. for Inf. Science, vol. 41, pp. 391-407, 1990.

[182] Gil-Jin Jang and Te-Won Lee,"A probabilistic approach to single channel blind signal separation," in Advances in Neural Information Processing Systems, S. Becker, S. Thrun, and K. Obermayer, Eds. 2003, vol. 15, The MIT Press.

[183] E. Saund, "A multiple cause mixture model for unsupervised learning," Neural Computation, vol. 7, no. 1, pp. 51-71, January 1995.

[184] A. Kab an and M. Girolami, "A dynamic probabilistic model to visualise topic evolution in text streams," Journal of Intelligent Information Systems, vol. 18, no. 2-3, pp. 107-125, March-May 2002.

[185] E. Bingham, A. Kaban, and M. Girolami, "Finding topics in dynamical text: application to chat line discussions," in 10th International World Wide Web Conference (WWW10), Hong Kong, May 1-5 2001, pp. 198-199.

[186] F. Bach and M. I. Jordan, "Kernel independent component analysis," Journal of Machine Learning Research, vol. 3, pp. 1-48, 2002.

[187] J. Klingseisen and M. D. Plumbley, "Towards musical instrument separation using multiple-cause neural networks," in Proceedings of ICA'2000, Helsinki, Finland, June 2000, pp. 447-452.

[188] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," Cybernetics and Systems, vol. 33, no. 6, pp. 603-627, 2002

[189] S. Dubnov and A. Ben-Shalom, "Review of ICA methods for query by similarity of natural sounds and sound effects," in Proceedings of ICA'2003, Nara, Japan, April 2003.

[190] T. Westerveld, "Probabilistic multimedia retrieval," in Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), 2002.

[191] A. Adelsbach and A.-R. Sadeghi, Zero-knowledge watermark detection and proof of ownership. In: I.S. Moskowitz, Editor, Information Hiding-Fourth International Workshop, IHW 2001, Lecture Notes in Computer Science, vol. 2137, Springer, Berlin, Germany/Pittsburgh, PA, USA (2001), pp. 273-288.

[192] D. Bogumi?, Reversing global and local geometrical distortions in image watermarking, in: Proceedings of the Sixth Information Hiding Workshop, Toronto, Canada, June 2004.

[193] D. Bogumi?, Removing digital watermarks based on image autocorrelation features (in Polish). TPO 2002, Serock, Poland, November 2002.

[194] H. Choi, K. Lee, T. Kim, Transformed-key asymmetric watermarking system, in: Proceedings of SPIE, vol. 4314, Security and Watermarking of Multimedia Contents III, 2001, pp.280-289.

[195] D. Bogumi, An asymmetric image watermarking scheme resistant against geometrical distortions, Signal Processing: Image Communication, Volume 21, Issue 1, 59-66, 2006.

[196] R.G. van Schyndel, A.Z. Tirkel, I.D. Svalbe, Key independent watermark detection, IEEE International Conference in Multimedia Computing and Systems Systems, vol. 1, 1999.

[197] P. Comon, Independent Component Analysis: a new concept, Signal Processing, Elsevier, 36(3), 287-314, 1994.

[198] J. V. Stone, A Brief Introduction to Independent Component Analysis in Encyclopedia of Statistics in Behavioral Science, Volume 2, pp. 907-912, Editors Brian S. Everitt & David C. Howell, John Wiley & Sons, Ltd, Chichester, 2005, ISBN 978-0-470-86080-9

[199] T. W. Lee, Independent component analysis: Theory and applications, Boston, Mass: Kluwer Academic Publishers, 1998, ISBN 0 7923 8261 7

[200] A. Hyvrinen and E. Oja. A fast fixed-point algorithm for independent component analysis. Neural Computation, 9(7):1483-1492, 1997.

[201] J. Karhunen, E. Oja, L. Wang, R. Vigrio, and J. Joutsensalo. A class of neural networks for independent component analysis. IEEE Trans. on Neural Networks, 8(3):486-504, 1997.

[202] T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. Neural Computation, 11(2):417-441, 1999.

[203] Nomura, T. Eguchi, M. Niwamoto, H. Kokubo, H. Miyamoto, M. An extension of the Herault-Jutten network to signals includingdelays for blind separation. ATR Human Inf. Process. Res. Labs., Kyoto; Neural Networks for Signal Processing 1996 VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop, Publication Date: 4-6 Sep 1996, Pages: 443-452

[204] Delorme, Anauld. ICA (Independent Component Analysis) for dummies. Home Page. http://www.sccn.ucsd.edu/ arno/indexica.html