DIGITAL WATERMARKING IN MEDICAL IMAGES

A Thesis Submitted for the Degree of Doctor of Philosophy

By

Jasni Mohamad Zain

School of Information Systems, Computing and Mathematics
Brunel University

November 2005

To my Parents,

Mohamad Zain Said &
Sabariah Yaacob

To my Husband,

Mohamed Fauzi Abdul Rani

To my Children,

Luqman,
Hilmi,
Hannah,
Syahid &
Tawfeq

Abstract

This thesis addresses authenticity and integrity of medical images using watermarking. Hospital Information Systems (HIS), Radiology Information Systems (RIS) and Picture Archiving and Communication Systems (PACS) now form the information infrastructure for today's healthcare as these provide new ways to store, access and distribute medical data that also involve some security risk. Watermarking can be seen as an additional tool for security measures.

As the medical tradition is very strict with the quality of biomedical images, the watermarking method must be reversible or if not, region of Interest (ROI) needs to be defined and left intact. Watermarking should also serve as an integrity control and should be able to authenticate the medical image.

Three watermarking techniques were proposed. First, Strict Authentication Watermarking (SAW) embeds the digital signature of the image in the ROI and the image can be reverted back to its original value bit by bit if required. Second, Strict Authentication Watermarking with JPEG Compression (SAW-JPEG) uses the same principal as SAW, but is able to survive some degree of JPEG compression. Third, Authentication Watermarking with Tamper Detection and Recovery (AW-TDR) is able to localise tampering, whilst simultaneously reconstructing the original image.

Table of Contents

List of Figures

List of Tables

Abbreviations

CDMA	Code Division Multiple Access
DCT	Discrete Cosine Transform
DE	Digital Envelope
DICOM	Digital Imaging and Communications in Medicine
DWT	Discrete Wavelet Transform
ECC	Error Correction Code
EPR	Electronic Patients Record
HVS	Human Visual System
LSB	Least Significant Bit
JPEG	Joint Picture Expert Group
JPEG-LS	JPEG Lossless Scheme
LUT	Look Up Table
MAC	Message Authentication Code
MD5	Message Digest by Ron Rivest
MSB	Most Significant Bit
NEMA	National Electrical Manufacturers' Association
PN	Pseudorandom Noise
RLE	Run Length Encoding
RSA	Rivest, Shamir and Adleman public key encryption
SHA	Secure Hash Algorithm
TIFF	Tag Image File Format
VPN	Virtual Private Network
VQ	Vector Quantization
VW2D	Variable-Watermark Two-Dimensional Algorithm

Acknowledgements

Alhamdulillahirabbil ‘alamin.

I would like to thank my supervisors, Dr Malcolm Clarke, Prof. Ray Paul and Dr Lynne Baldwin for their invaluable advice and assistance throughout the course of this research.

Declaration

The following papers have been published, or submitted for publication, as a direct result of this research.

J. M. Zain and M. Clarke, "Fragile Image Watermarking with One-to-One Block Mapping ", accepted for presentation in MMU International Symposium on Information and Communication Technologies, Petaling Jaya, Malaysia, 24-25th November 2005.

Jasni M Zain and Malcolm Clarke, "LSB Reversible Watermarking Surviving JPEG Compression", in The 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Shanghai, China, 1-4 September 2005.

J. M. Zain, L.P. Baldwin and M. Clarke, (in submission) "Reversible Watermarking for Authentication of Medical Image", Journal of Advancing Information and Management Studies.

Jasni Zain and Malcolm Clarke, "Issues in watermarking medical images", PREP2005, University of Lancaster, April 2005.

Jasni Zain and Malcolm Clarke, "Security In Telemedicine: Issues in Watermarking Medical Images", 3rd International Conference Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2005), Susa, Tunisia, 27-31 March 2005.

Jasni Zain, "Security in Telemedicine: Watermarking medical images ", Medical Error and Technologies Research Workshop. London, 3 November 2004.

J.M Zain, L. P. Baldwin, M. Clarke, " Reversible watermarking for authentication of DICOM images", in The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, California, 1-4 September 2004.

J. M. Zain, M. Clarke, L. P. Baldwin, “The effect of reversible LSB manipulation to the quality of image”, in PREP2004, University of Hertfordshire, April 2004.

J Zain and R S H Istepanian, “Digital Watermarking in Wireless Telemedical Environment”, in Proceedings of PREP2003, Exeter University, Exeter, April 2003.

J M Zain, “Globalization and Telecommunication Technologies”, in MRG 2nd annual Conference, Manchester, September 2003.

J M Zain, “Threats and Challenges in Securing Telemedicine System”, in MRG 2nd annual Conference, Manchester, September 2003.

Chapter 1

Security of Medical Images

1.1 Introduction to Information Security

One of the major concerns throughout the world today is to make high quality healthcare available to all. Traditionally, part of the difficulty in achieving equitable access to healthcare has been that the provider and the recipient must be physically present in the same place. Recent advances in information and communication technologies have increased the number of ways in which healthcare can be delivered to reduce these difficulties.

Telemedicine, the area where medicine and information and communications technology (ICT) meet, is probably the part of this revolution that could have the greatest impact on healthcare delivery. The prefix 'tele' derives from the Greek 'at a distance', and therefore, more simply telemedicine is medicine at a distance. The information infrastructure of modern healthcare is based on digital information management. While the recent advances in information and communication technologies provide new means to access, handle and move medical information, they also compromise their security due to their ease of manipulation and replication. All patients records, electronic or not, linked to medical secrecy, must be confidential. The

digital handling of the EPR (Electronic Patient Record) on a network requires a systematic content validation that is aimed at quality control: actuality (precise interest of the information at a given instant) and reliability (authentication of the origin and integrity).

Attacks on security are best characterised by viewing the function of the computer system as a provision of information. In general, normal communication is represented as a flow of information from source to destination.

There are four categories of attacks:

- Interruption: An attack on availability. Information is destroyed or becomes unavailable or unusable.
- Interception: An attack on confidentiality. An unauthorised party gains access to information.
- Modification: An attack on integrity. An unauthorised party not only gains access to, but also tampers with information.
- Fabrication: An attack on authenticity. An unauthorised party inserts counterfeit objects into the system.

These attacks can be divided further into two categories, according to the nature of the attacks:

- Active Attacks: These attacks involve modification of the data stream or the creation of a false stream and can be subdivided into four categories:
    1. Masquerade: One entity pretends to be a different entity.
    2. Replay: The passive capture of a data unit and its subsequent retransmission to produce an unauthorised effect.
    3. Modification of messages: Some portion of a legitimate message is altered, or messages are delayed or recorded to produce an unauthorised effect.
    4. Denial of service: One prevents or inhibits the normal use or management of communications facilities.

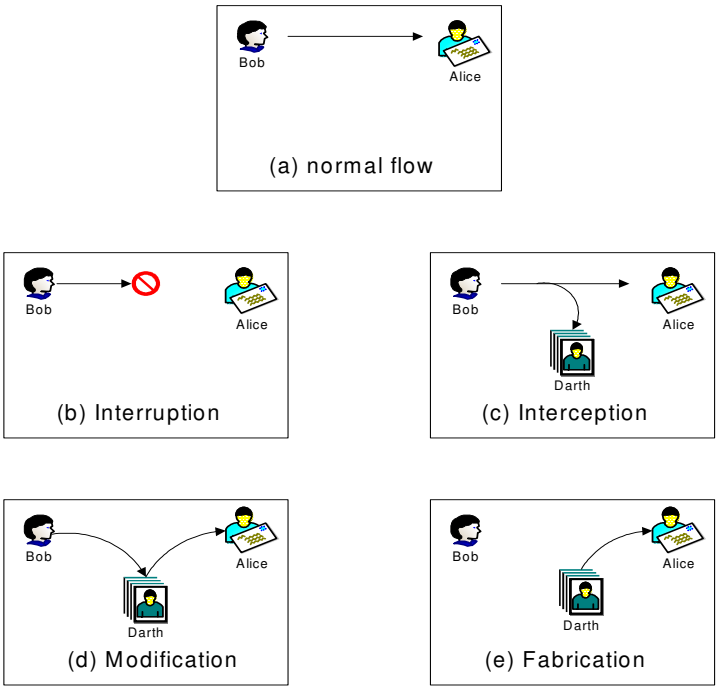


Figure 1.1 Security attacks

- Passive Attacks: These attacks involve eavesdropping on, or monitoring of, transmission and can be subdivided into two categories:
  1. Release of message contents: An unauthorised party obtains information that is being transmitted.
  2. Traffic analysis: An unauthorised party obtains information useful in guessing the nature of communication by observing the pattern of masked message transmissions.

This research will deal with modification and fabrication attacks on medical images. We will look at how to authenticate medical images using watermarking.

1.2 Image Authentication and Motivation

Image authentication can assure receivers that the received image is from the authorized source and that the image content is identical to the one sent. It is becoming easier and easier to tamper with digital image in ways that are difficult to detect. For example, Figure 1.2 shows two nearly identical images using readily available software (e.g.

Adobe Photoshop). The cyst was removed from the image by using the healing brush tool. It is difficult if not impossible to tell which picture is the original and which has been tampered with. If this image were a critical piece of evidence in a legal case or police investigation, this form of tampering might pose a serious problem.

The problem of authenticating messages has been studied in cryptography (Stinson 1995). Specifically, we are interested in methods for answering the following questions:

i. Has the image been altered in any way what so ever?
ii. What parts of the image have been altered?
iii. Can an altered image be restored?

To implement such methods for medical images, the following questions are relevant:

iv. Do medical images have the same properties as other images?
v. What are the requirements needed to make watermarking suitable for medical images?

Many non-watermarking methods exist for answering these questions. Two common cryptographic approaches are the creation of a hash function and digital signature. The classical scenario for image authentication can be described as follows (Stallings 2003): A sender S wants to transmit a digital image I to a receiver R. When the image I^r is eventually delivered to R, by means of a network facility or any other media capable of storing digital data, an effective authentication scheme must ensure with high probability that:

- The Image I^r received by R is exactly the same as I the sender S has sent. (Integrity verification)
- The receiver R can verify the alleged source of the image I^r, where R can determine if S has actually sent I^r, or if a pirate has forged it. (Alleged Source Verification)

(a) An ultrasound image of a cystic breast tissue

(b) An ultrasound image of a normal breast tissue

Figure 1.2 Ease of modifying images

- R can demonstrate that I^r was actually sent by S, and S cannot deny having sent I^r. (non-repudiation Property).

Image authentication techniques are usually based on two kinds of tools: digital signature and watermarking. A digital signature is non-repudiation, encrypted version of the message digest extracted from the data. It is usually stored as a separate file, which can be attached to the data to prove integrity and originality. Watermarking techniques consider the image as a communication channel. The embedded watermark, usually imperceptible, may contain either a specific producer ID or some content-related codes that are used for authentication.

Digital watermarking (Cox et al. 2002, Langelaar et al. 2000) offers a promising alternative to digital signatures in image authentication applications. The use of watermarks instead of digital signatures typically records additional functionality by exploiting inherent properties of image content. The main advantage of digital watermarking is that the authentication information is directly embedded into the image data. As a result, the authentication information survives even when the host image undergoes format conversions. In contrast, a digital signature appended in the header of an image file may be easily stripped off, for instance, when the file is opened and saved in a different format. The digital watermark's capability for isolating manipulated image regions is another advantage. This functionality is known as the tamper localisation property. It is worth mentioning that both digital signatures and authentication watermarks are useful only for establishing the source of the image and detecting manipulations occurring after the signature/watermark has been inserted. Neither technique by itself is capable of certifying that a signal represents an original unaltered scene, unless supported by additional mechanisms (Friedman 1993). In this respect, digital watermarking differs from forensic image analysis (Federation Bureau of Investigation 2000).

Most watermarking techniques modify, and hence distort, the host signal in order to insert authentication information. In many applications, loss of image fidelity is not prohibitive as long as the original and modified images are perceptually equivalent. On

the other hand, in medical, military and legal imaging applications, where the need for authentication is often paramount, there are typically stringent constraints on data fidelity that prohibit any permanent signal distortion in the watermarking process. For instance, artifacts in a patient's diagnostic image may cause errors in diagnosis and treatment with possible life-threatening consequences. Likewise, in military applications, satellite and aerial photographs are often enlarged, enhanced or further processed by computer vision algorithms. Unless the loss of fidelity is either carefully limited or eliminated altogether, the corresponding artifacts may be amplified by the post-processing operations. In these applications, the permanent loss of signal fidelity due to digital watermarking can be remedied by lossless data embedding (also referred as reversible, invertible or distortion-free data embedding) techniques. These techniques, like their lossy counterparts, insert information bits by modifying the host signal, thus induce an embedding distortion. Nevertheless, they also enable the removal of such distortions and the exact/lossless restoration of the original host signal after extraction of the embedded information. Lossless data embedding methods can be employed for digital image authentication. Lossless (reversible) authentication watermarks provide a complete framework; the authentication property of the watermark protects the integrity of the image, whereas the quality is preserved by the reversibility of the watermarking process.

1.3 Current Security Methods for Medical Images

- Data Encryption

Encryption is the most useful approach to assure data security during its transmission through public communication networks. Image data scrambled by a sender cannot be understood by anyone other than an intended party and assures data security during transmission, but not before or after.

  - Virtual Private Network (VPN) is the most common application of data encryption techniques to ensure data security during its transmission through public communication networks.

- DICOM Security or DICOM standard part 15 has been released to provide a standardised method (selection of security standards, encryption algorithms and parameters) for secure medical image communication but has yet to be implemented by the industrial and medical community.

- Data Embedding

Data embedding can be a form of steganography that conceals patient information and the digital signature in the image so that the visual quality of the image is not perceptually affected. It provides a permanent assurance of image data security no matter when and how the image has been manipulated. But there is no standard embedding method and it is also difficult to implement for a variety of medical image modalities. Watermarking researchers in the medical field have also incorporated hashing to produce image digests and use them as watermarks (Cao et al. 2003, Guo and Zhuang 2003, Zhou et al. 2001).

1.4 Watermarking and Steganography

Watermarking, that is the technique of placing and transmitting a small amount of data imperceptibly in the host or cover data has many applications including broadcast monitoring, owner identification, proof of ownership, and content authentication. Paper watermarks are used regularly as an authentication (anti-counterfeiting) measure in valuable documents, such as bank notes, cheques and visa stamps. For instance, the authenticity of a bank note is confirmed by the existence of a visible watermark pattern when the note is held to the light. Paper watermarks are designed to be i) easily detectable, ii) hard to reproduce, and iii) invisible or unobtrusive in normal use of the document. Digital watermarks inherit many of the paper watermarks features and properties: they are digital patterns superimposed on digital signals; the patterns should be easily detectable, yet be very hard to reproduce without specific knowledge (cryptographic keys); the watermark should be invisible or unobtrusive during normal use of the digital signal.

However, steganography or data hiding has a long history and the use of paper watermarks for copy protection can be traced back to the thirteenth century (Murray 1996). The earliest forms of information hiding can actually be considered to be highly crude forms of private-key cryptography; the "key" in this case being the knowledge of the method being employed (security through obscurity). Steganography books are filled with examples of such methods used throughout history. Greek messengers had messages tattooed into their shaved head, concealing the message when their hair finally grew back. Wax tables were scraped down to bare wood where a message was scratched. Once the tables were re-waxed, the hidden message was secure (Petitcolas 2000). Over time these primitive cryptographic techniques improved, increasing speed, capacity and security of the transmitted message.

Today, crypto-graphical techniques have reached a level of sophistication such that properly encrypted communications can be assumed secure well beyond the useful life of the information transmitted. In fact, it is projected that the most powerful algorithms using multi kilobit key lengths could not be comprised through brute force, even if all the computing power worldwide for the next 20 years was focused on the attack. Of course the possibility exists that vulnerabilities could be found, or computing power breakthroughs could occur, but for most users in most applications, current cryptographic techniques are generally sufficient.

Why then pursue the field of information hiding? Several good reasons exist, the first being that "security through obscurity" is not necessarily a bad thing, provided that it is not the only security mechanism employed. Steganography for instance allows us to hide encrypted messages in mediums less likely to attract attention. A garble of random characters being transmitted between two users may tip off a watchful third party that sensitive information is being transmitted; whereas baby pictures with some additional noise present may not. The underlying information in the pictures is still encrypted, but attracts far less attention by being distributed in the picture than it would otherwise.

Nowadays, there exist watermarking methods for virtually every kind of digital media: text documents (Su et al. 1998, Brassil et al. 1999), images (Tsai et al. 2004, Zhang et al. 2003, Paquet et al. 2003), video (Sun and Chang 2003, Okada et al. 2002), audio (Li

and Xue 2003, Yan et al. 2004), even for 3D polygonal models (Kwon et al. 2003, Benedens and Busch 2000), maps (Barni et al. 2001) and computer programs (Monden et al. 2000). Interestingly, watermarking technology is not limited to digital media, but is also applicable to chemical data like protein structures, for example (Eggers et al. 2001).

1.5 Research Objectives

There are three research objectives:

1. To investigate methods for authentication watermarking.
2. To develop techniques for authentication appropriate for a chosen medical image modality.
3. To investigate and evaluate any such technique on the chosen medical image modality.

1.6 Research Strategy and Method

The sources of information for the present work came from three different subject areas. Firstly, information security in general; secondly, hiding information, known as steganography; and thirdly medical imaging.

The research concentrates on authenticity and integrity of medical images, and investigates current techniques of authentication with an emphasis on those most suitable for medical images. A few issues need to be clarified before choosing tools and techniques for this research. The first issue to consider is whether complete authentication or content authentication is required as an entity.

Complete authentication refers to techniques that consider the image and do not allow any manipulations or transformation (Wu and Liu 1998, Yeung and Mintzer 1997). Many existing message authentication techniques can be applied directly. For example, a digital signature might be placed in the LSB of the uncompressed data or the header of the compressed data. Manipulations will be detected if the hash value of the altered message does not match the digital signature. In practice, fragile watermarks or traditional digital signatures may be used for complete authentication.

Content authentication refers to a different objective that is applicable to multimedia data. The meaning of multimedia data is based on its content instead rather than specific bit content. In some applications, manipulations on the bit streams without changing the meaning of content are considered as acceptable. Compression is an example. Digital Imaging and Communication in Medicine (DICOM) standard has included JPEG (lossy and lossless), JPEG-LS and RLE (known as TIFF) compressions in their standard. JPEG2000 has also been considered in the report (National Electrical Manufacturers Association (NEMA) 2002).

The second issue is whether the watermarks are reversible or permanent. Ideally the medical image should be unaltered through the process of watermarking (Macq and Dewey 1999, Giakoumaki et al. 2003, Yang and Bao 2003) and the watermarking should be reversible so that the original image can be restored. However, it may be argued that if the change is imperceptible and has no impact on diagnosis then it is acceptable and may be compared with compression, which is accepted.

The third issue is if we decide to have watermarks for content authentication, whether compression should be distinguished from other manipulations. Previous watermarks are either too fragile for compression or too flexible to detect malicious manipulations. The performance of an authenticator should be simultaneously evaluated by two parameters: the probability of false alarm and the probability of missing manipulations. Fragile watermarks, which have a low probability of missing manipulations, usually fail to survive compression so that their probability of false alarm is very high. Previous researchers have attempted to modify the fragile watermark to make it robust to compression (Wolfgang and Delp 1996, Zhu et al. 1996). However, such modifications then failed to distinguish both compression and tampering. In general, watermarks made robust to most manipulations are usually then too robust to detect malicious manipulations.

Watermarking capacity is determined by invisibility and robustness requirements. The relationship between capacity, invisibility and robustness is shown in Figure 1.3.

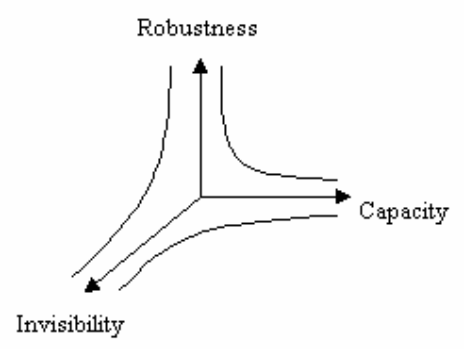


Figure 1.3 Watermarking properties

It is seen that if one parameter is fixed, then the other two parameters are inversely proportional. For instance, a specific application may determine how many message bits are needed, copyright protection may need to embed about 10 bytes and authentication may need from 100-1000 bits for a 265 X 256 image. After the amount to embed is decided, there exists a trade-off between visual quality and robustness. Robustness refers to the extraction of embedded bits with a probability of error equal to or approaching zero. Visual quality represents the quality of watermarked image. In general, if we want to make our message bits more robust against attack, then a longer codeword will be necessary to provide better error resistance. However, degradation in visual quality can be expected.

1.7 Dissertation Outline

This thesis is divided into 7 chapters and organised as follows:

Chapter 1: This chapter introduces the problem area and outlines approaches to be explored. In this chapter watermarking is introduced as the technique used in the research.

Chapter 2: This chapter presents the area of message authentication. The digital signature is discussed as a possible watermark. A review of current image authentication is presented and various embedding techniques are discussed.

Chapter 3: This chapter presents medical image watermarking. Issues and properties of medical image watermarking are discussed. Some approaches in

watermarking medical images and the issues of tamper localisation and restoration are presented.

Chapter 4: This chapter proposes two strict authentication watermarking techniques SAW and SAW-JPEG. SAW embeds the digital signature of the medical image in the region of non-interest. SAW-JPEG is an enhanced SAW and is made to be robust to some degree of JPEG compression.

Chapter 5: This chapter proposes another authentication watermarking with tamper detection and recovery AW-TDR.

Chapter 6: This chapter discusses the results obtained from Chapter 4 and Chapter 5 and evaluates the proposed techniques.

Chapter 7: This chapter presents a summary of the thesis and conclusions to the work.

Chapter 2

Literature Review

2.1 Introduction

This chapter introduces the area of image authentication, the techniques available and introduces the area of image watermarking. The chapter is structured as follows:

- Section 2.2 introduces hash functions and how it is used to secure message transmission and describes the method. The use of private key and public key is also discussed.
- Section 2.3 describes a digital signature algorithm and the purpose of having one person who can produce a signature that can checked by everybody else by using private and public key. This section also describes how a digital signature algorithm is used for image authentication.
- Section 2.4 presents an image authentication scheme using content-based, feature-based, edge-based, mean-based and relation-based methods.
- Section 2.5 gives definition of fragile and semi-fragile watermarking and provides a review of methods available.
- Section 2.6 lists the requirements for watermarking-based authentication system.
- Section 2.7 describes the main components for a watermarking system.

- Section 2.8 will show some of the most frequent attacks that an image authentication system has to overcome
- Section 2.9 presents embedding techniques for watermarking.

2.2 Hashed Message Authentication Code (HMAC)

A hash function such as MD5 (Rivest April 1992) and SHA-1 (National Institute of Standards and Technology 1995) produces a one-way message digest, a fingerprint of a file, message, or other block of data. The hash based message authentication code (HMAC) encrypts the hash value of the message with a secret key shared by the sender and receiver. This technique assumes that two communicating parties, A and B share the same secret key K_{AB}. When A has a message M to send to B, it calculates the message authentication code as a function of the message and the key: $MAC_M = H(K_{AB}, M)$.

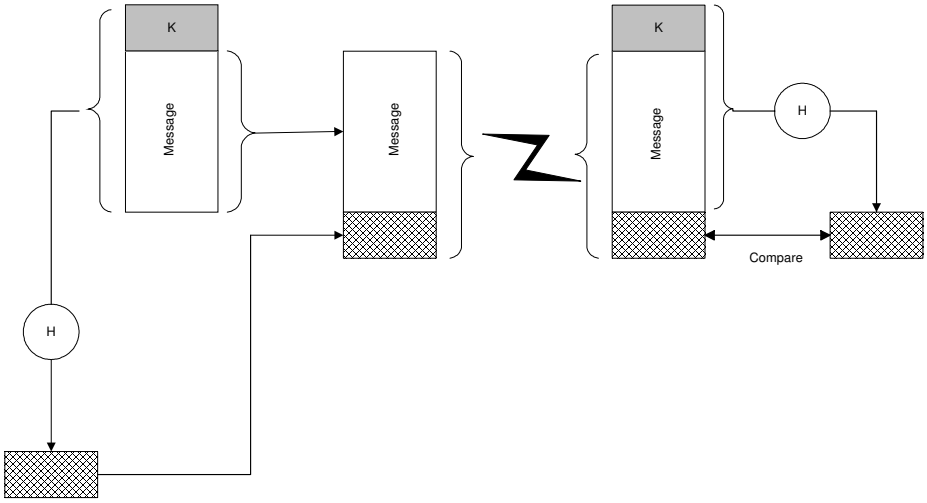


Figure 2.1 HMAC (Adapted from Network Security Essentials page 58)

The message and the MAC code are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new message authentication code. The received code is compared to the calculated code. If we assume that only the receiver and the sender know the identity of the secret key, and if the received code matches the calculated code, then

1. The receiver is assured that the message has not been altered. If an attacker alters the message, the received code will not match the calculated code.
2. The receiver is assured that the message is from the alleged sender as no one else could prepare a message with a proper code.

Modern cryptography can use either private-key or the public-key key method (Garfinkel and Spafford 1996). Private-key cryptography (symmetric cryptography) uses the same key for data encryption and decryption. It requires both the sender and the receiver to agree on a key before they can exchange message securely. Although computation speed for obtaining the private-key is acceptable, the management of the keys is difficult.

Public-key cryptography (asymmetric cryptography) uses two different keys (a key pair) for encryption and decryption. The keys in the key pair are mathematically related, but it is computationally infeasible to deduce one key from the other. In public-key cryptography, the public key can be made public. Anyone can encrypt a message using a public key, but only the corresponding private-key owner can decrypt it. Public-key methods are much more convenient to use because they do not share the key management problem as in private-key methods. However they require a longer time for encryption and decryption. Digital signature is a major application of public-key cryptography (Rivest et al. 1978).

2.3 Digital Signature Algorithms

The basic idea of a digital signature is that a signature on a message can be created by only one person, but checked by anyone. It can thus perform the sort of function in the electronic world that ordinary signatures do in the world of paper. The asymmetric encryption algorithms published in the late 1970s, such as RSA, in conjunction with the secure hash functions, are digital signature algorithms, which allow the sender to associate its unforgeable imprint with the digital image, so that the receiver can check its integrity and its source. Non-repudiation is also guaranteed.

The asymmetric encryption involves the use of two separate keys: a public key made public for others to decrypt a received message, and a private key known only to its owner to encrypt the original. When A has a message M to send to B, it calculates the digital signature sig M as a function of the hashed message H(M) and the private key Kprivate: sigM = F(Kprivate, H(M)). The message plus digital signature are transmitted to the intended receiver. The receiver performs the same hash calculation on the received message to generate a hashed message. The receiver also decrypts the received signature sigM, using public key Kpublic, to get the received hashed message. The received hashed message is compared to the hashed message. If we assume that only the sender knows the identity of the secret key, and if the received hashed message is identical to the new hashed message, then

- The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the code, then the receiver's calculation of the hashed message will differ from the new hashed message. Because the attacker is assumed not to know the private key, the attacker cannot alter the code to correspond to the alterations in the message.
- The receiver is assured that the message is from the alleged sender. Because no one knows the private key, no one else could prepare a message with a proper digital signature.

Image authentication is projected as a procedure of guaranteeing that the image content has not been altered, or at least that the visual (or semantic) characteristics of the image are maintained after incidental manipulations such as JPEG compression. In other words, one of the objectives of image authentication is to verify the integrity of the image. For many applications such as medical archiving, news reporting and political events, the capability of detecting manipulations of digital images is often required.

To address both the integrity and legitimacy issues, a wide variety of techniques have been proposed for image authentication. Depending on the ways chosen to convey the authentication data, these techniques can be divided into two categories: labelling-based techniques (e.g., the method proposed by Friedman 1993) and watermarking-based

techniques (e.g., method proposed by Walton 1995). The main difference between these two categories of techniques is that labelling-based techniques create the authentication data in a separate file while watermarking-based authentication can be accomplished without the overhead of a separate file.

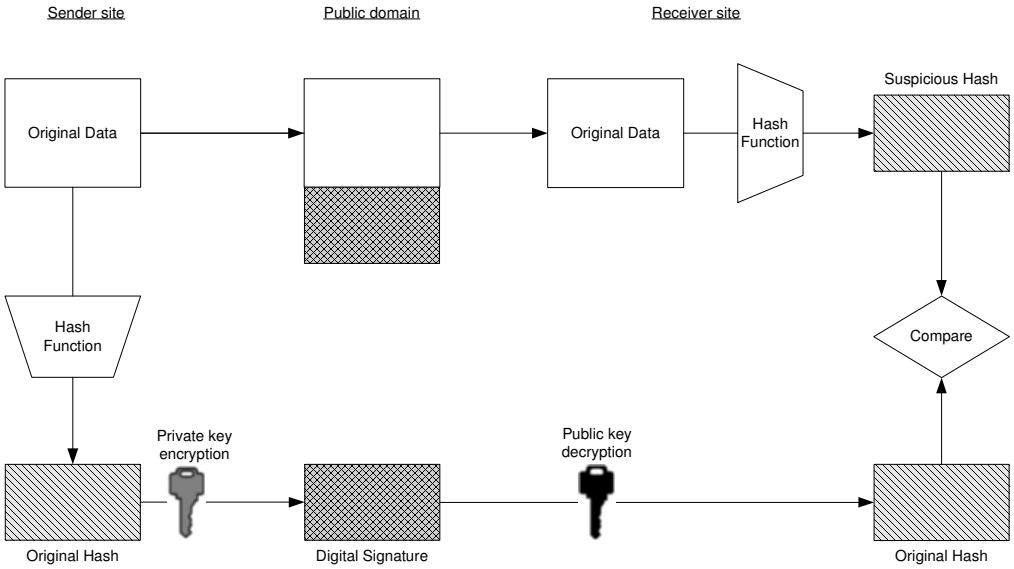


Figure 2.2. Basic model of a digital signature

The digital signature-based image authentication is based on the concept of a digital signature, which is derived from a cryptographic technique called public key cryptosystem (Rivest et al. 1978, Diffie and Hellman 1976). Figure 2.2 shows the basic model of a digital signature. The sender first uses a hash function, such as MD5 (Rivest 1992), to hash the content of the original data to a small file called digest. MD5 was the most widely used secure hash algorithm until the last few years that the security of a 128-bit hash code has become questionable (Dobbertin 1996) and in summer 2004 was broken by Chinese researchers (Wang et al. 2004, Hawkes et al. 2005). Then the digest is encrypted with the sender's private key. The encrypted digest can form a unique 'signature' because only the sender has the knowledge of the private key. The signature is then sent to the receiver along with the original information. The receiver can use the sender's public key to decrypt the signature, and obtain the original digest. The received information can also be hashed by using the same hash function at the sender's side. If

the decrypted digest matches the newly created digest, the legitimacy and the integrity of the message are therefore authenticated.

There are two points worth noting in the process of a digital signature. First, the plaintext is not limited to text file. In fact, any types of digital data, such as digitised audio data and digital image. Therefore the original data in Figure 2.2 can be replaced with a digital image, and the process of a digital signature can then be used to verify the legitimacy and integrity of the image. The concept of the trustworthy digital camera (Friedman 1993) for image authentication is based on this idea. Friedman (1993) associated the idea of a digital signature with the digital camera and proposed a 'trustworthy digital camera'. The proposed digital camera uses a digital sensor instead of film and delivers the image directly in a computer-compatible format. A secure microprocessor is assumed to be built in the digital camera and is programmed with the private key at the factory for the encryption of the digital signature. The public key needed for later authentication appears on the camera as well as the image's border. Once the digital camera captures the image, it produces two output files. One is an all-digital industry-standard file format representing the captured image and the other is an encrypted digital signature generated by applying the camera's unique private key to a hash of the captured image file. The digital image file and the digital signature can later be distributed freely and safely.

Image authentication is accomplished by calculating the hash of the received image, and by using the public key to decode the digital signature to reveal the original hash; the two hash values are compared. If these two hash values match, the image is considered to be authentic. If these two hash values are different, the integrity of this image is questionable. It should be noted that the hash algorithms such as SHA-256 are sensitive to single bit changes. This is strict authentication. However in the process of lossy compression, although the image is essentially retained, individual pixel values may be changed. Strict authentication will determine the image is no longer authentic and does not provide a useful check. A different check is required.

2.4 Other Image Authentication Schemes

- **Content-based authentication**

Image manipulation such as lossy compression, changes individual pixel values and so strict authentication (hash value calculated from all bit values in the image) will fail. In these cases a method must be sought to determine features in the image that will be invariant through the compression-decompression process. Edge information, DCT coefficients, colour, and intensity histograms are regarded as potential invariant features.

In Schneider and Chang's (Schneider and Chang 1996) method, the intensity histogram is employed as the invariant feature in the implementation of the content-based image authentication scheme. To be effective, the image is divided into blocks of variable sizes and the intensity histogram of each block is computed separately and is used as the authentication code. To tolerate incidental modifications, the Euclidean distance between intensity histograms was used as a measure of the content of the image. They pointed out that using a reduced distance function could increase the maximum permissible compression ratio up to 14:1 if the block average intensity is used for detecting image content manipulation.

- **Feature-based method**

Bhattacharjee and Kutter (1998) proposed another algorithm to extract a smaller size feature of an image. Their feature extraction algorithm is based on the so-called scale interaction model. Instead of using Gabor wavelets, they adopted Mexican-Hat wavelets as the filter for detecting the feature points. The algorithm for detecting feature points is depicted as follows.

- Define the feature-detection function, $P_{ij}(.)$ as:

$$\mathrm{P}_{ij}(\vec{x}) = \left| M_i(\vec{x}) - \gamma . M_j(\vec{x}) \right| \tag{2.1}$$

where $M_i(\vec{x})$ and $M_j(\vec{x})$ represent the responses of Mexican-Hat wavelets at the image location $\vec{x}$ for scales I and j respectively. For the image A, the wavelet response $M_i(\vec{x})$ is given by:

$$M_i(\vec{x}) = \langle (2^{-i}\psi(2^{-1}\cdot\vec{x})); A\rangle \tag{2.2}$$

where $\langle .;. \rangle$ denotes the convolution of its operands. The normalising constant γ is given by $\gamma=2^{-(i-j)}$, the operator $|\cdot|$ returns the absolute value of its parameter, and the $\psi(\vec{x})$ represents the response of the Mexican-Hat mother wavelet, and is defined as:

$$\psi(\vec{x}) = (2-|\vec{x}|^2)\exp(-\frac{x^2}{2}) \tag{2.3}$$

- Determine points of local maximum of $P_{ij}(.)$. These points correspond to the set of potential feature points
- Accept a point of local maximum in $P_{ij}(.)$ as a feature-point if the variance of the image pixels in the neighbourhood of the point is higher than a threshold. This criterion eliminates a suspicious local maximum in featureless regions of the image.

The column positions and row positions of the resulting feature points are concatenated to form a string of digits, and then encrypted to generate the image signature. In order to determine whether an image A is authentic with another known image B, the feature set S_A of A is computed. The feature set S_A is then compared with the feature set S_B of B that is decrypted from the signature of B. The following rules are adopted to authenticate the image A.

- Verify that each feature location is present both in S_B and in S_A.
- Verify that no feature location is present in S_A but absent in S_B.
- Two feature points with coordinates $\vec{x}$ and $\vec{y}$ are said to match if:

$|\vec{x}-\vec{y}|<2$

- **Edge-based method**

The edges in an image are the boundaries or contours where significant changes occur in some physical aspects of an image. Edges are a strong content feature for an image. However, coding edge values and positions can carry a large overhead. One way to resolve this problem is to use a binary map to represent only the edges. For example, Li et al (2003) used a binary map to encode the edges of an image in their image authentication scheme. However it is known that edges will be modified when high compression ratios are used. Consequently, the success of using edges as the authentication code is greatly dependent on the capacity of the authentication system to discriminate the differences the edges produced by content-preserving manipulations from those content-changing manipulations.

- **Mean-based method**

The local mean is a simple and practical image feature to represent the content of an image. Lou and Liu (2000) proposed such an algorithm to generate a mean-based feature code. The original image is divided into non-overlapping blocks and the mean of each block calculated and quantized according to a predefined parameter. The calculated results are then encoded to form the authentication code. In the verification process the quantized means of each block of the received image is calculated. The quantized code is compared with the original quantized code on a block-by-block basis. A binary error map is produced as an output with '1' denoting match and '0' denoting mismatch. The verifier can thus tell the possibly tampered blocks by inspecting the error map. There is also some capability to restore the untampered version, which may be attractive in some real time image application such as video.

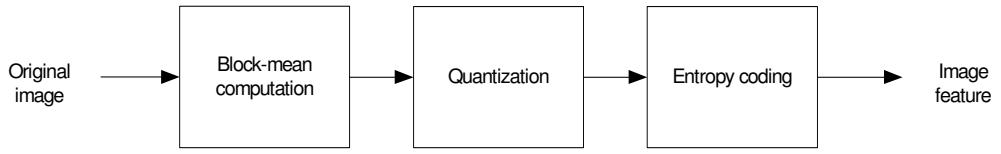


Figure 2.3 Mean based feature code

- **Relation-based method**

Unlike the methods introduced before, relation-based methods divide the original image into non-overlapping blocks, and use the relation between blocks as the feature code. The method proposed by Lin and Chang (Lin and Chang 2001) is called SARI. The feature code in SARI is generated to survive the JPEG compression. To serve this purpose, the process of the feature code generation starts with dividing the original image into 8x8 non-overlapping blocks. Each block is then DCT transformed. The transformed DCT blocks are further grouped into two non-overlapping sets. There are equal numbers of DCT blocks in each set. A secret key-dependant mapping function then maps one-to-one each DCT block in one set into another DCT block in another set, and generates N/2 DCT block pairs. For each block pair, a number of DCT coefficients are then selected and compared. Comparing the corresponding coefficients of the paired blocks then generates the feature code.

The feature code of the received image is extracted using the same secret key and is compared with the original feature code. If neither block in each block pair has been maliciously manipulated, the relation between the selected coefficients is maintained. Otherwise, the relation between the selected coefficients may be changed.

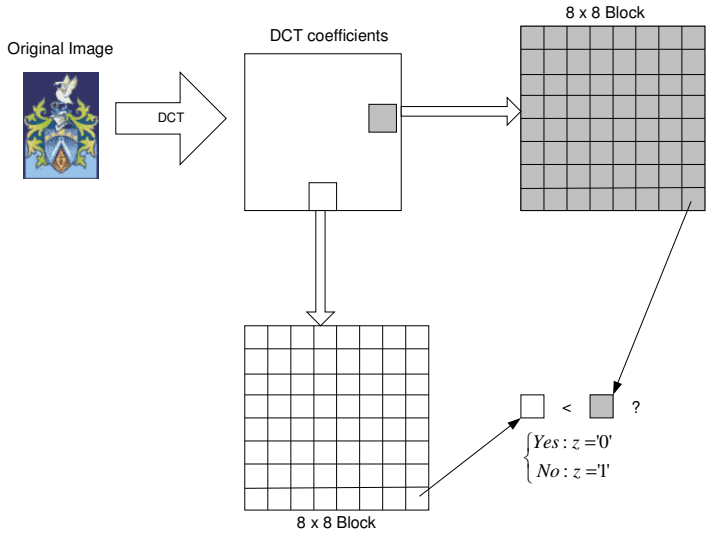


Figure 2.4 Feature code generated with SARI authentication scheme

2.5 Fragile and Semi-fragile Watermarking

A fragile watermarking is one that is likely to be destroyed and become undetectable after the image has been modified in any way. Watermarking researchers have considered fragility as undesirable and therefore seek to design robust watermarks that can survive many forms of distortion. However, fragility can be an advantage for authentication purposes. If a fragile watermark is detected correctly in an image, we can say that the image has not been altered or tampered with since the watermark has been embedded. A fragile watermark is a mark that is readily altered or destroyed when the host image is modified through linear or nonlinear transformation (Yeung and Mintzer 1998). In the case of authenticity, a fragile watermark has to prove that the image has been modified and is no longer authentic. However, for copy protection applications, the watermark has to be robust and be able to withstand different types of alterations such as lossy compression and filtering.

Fragile watermarks are not suited for enforcing copyright ownership of digital images. An attacker would attempt to destroy the embedded watermark and fragile watermarks are by definition easily destroyed. The sensitivity of fragile watermarks to modification leads to their use in image authentication. That is, it may be of interest for parties to verify that an image has not been edited, damaged, or altered since it was marked. Image authentication systems have applicability in law, commerce, defense, and journalism. Since digital images are easy to modify, a secure authentication system is useful in showing that no tampering has occurred during situations where the credibility of an image may be questioned. Common examples are the marking of images in a database to detect tampering (Mintzer et al. 1998), for example in the use of a "trustworthy camera" so news agencies can ensure an image is not fabricated or edited to falsify events (Friedman 1993), and the marking of images in commerce so a buyer can be assured that the images bought are authentic upon receipt (Wong 1998). Other situations include images used in courtroom evidence, journalistic photography, or images involved in intelligence.

As mentioned previously, one of the methods used to verify the authenticity of a digital work is the use of a signature system (Stallings 2003). In a signature system, a digest of the data to be authenticated is obtained by the use of cryptographic hash functions (Stallings 2003, Wolfgang and Delp 1999). The digest is then cryptographically signed to produce the signature that is bound to the original data. Later, a recipient verifies the signature by examining the digest of the (possibly modified) data and using a verification algorithm to determine if the data is authentic.

While the purpose of fragile watermarking and digital signature systems are similar, watermarking systems offer several advantages compared to signature systems (Memon et al. 1999) at the expense of requiring some modification (watermark insertion) of the image data. As a watermark is embedded directly in the image data, no additional information is necessary for authenticity verification. This is unlike digital signatures since the signature itself must be bound to the transmitted data. Therefore the critical information needed in the authenticity testing process is discreetly hidden and more difficult to remove than a digital signature. Also, digital signature systems view an image as an arbitrary bit stream and do not exploit its unique structure. Therefore a signature system may be able to detect that an image has been modified but cannot characterise the alterations. Many watermarking systems can determine which areas of a marked image have been altered and which areas have not, as well as estimate the nature of the alterations.

2.5.1. Examples of Fragile Marking Systems

Early fragile watermarking systems embedded the mark directly in the spatial domain of an image, such as techniques described in Walton (1995) and van Schyndel et al. (1994). These techniques embed the mark in the least significant bit plane for perceptual transparency. Their significant disadvantages include the ease of bypassing the security they provide (Yeung and Mintzer 1998, Fridrich 1998) and the inability to lossy compress the image without damaging the mark.

Any processing of the image, such as compression will result in changes to the LSB. If a watermark is to be embedded in the LSB plane of the image, we imply that the image has not undergone any such process. Fragile watermarking algorithms are concerned with complete integrity verification. The slightest modification of the host image will alter or destroy the fragile watermark. Yeung and Mintzer (1998) embeds a binary logo of the same size as the host image by means of a key dependent look-up table (LUT) that maps every possible pixel luminance value to either 0 or 1. The watermark is inserted by adjusting the least significant bit (LSB) value of each image pixel in the spatial domain to match its corresponding LUT value. At the receiving side, the LUT can be reconstructed due to the knowledge of the secret key. The integrity verification can be performed either by simple visual inspection of the extracted watermark, or by automated comparison with the original one. This watermarking scheme is very sensitive to any distortion in the image and is very vulnerable to a block analysis attack.

Fridrich and Baldoza (2000) improved the algorithm by using 64x64 block cipher instead of LUT, and the watermark is embedded in a 32x32 block. The improved scheme can be used against the block analysis attack.

A further fragile marking technique described by Wong (1999), obtains a digest using a hash function. The image, its dimensions and marking key are hashed during embedding and used to modify the least-significant bit plane of the original image. This is done in such a way that when the correct detection side information and unaltered marked image are provided to the detector, a bi-level image chosen by the owner (such as a company logo or insignia) is observed. This technique has localisation properties and can identify regions of modified pixels within a marked image. However, Holliman and Memon (2000) soon presented a vector quantization (VQ) counterfeiting attack that can construct a counterfeit image from a VQ codebook generated from a set of watermarked images. To solve the problem of VQ counterfeiting attack, several enhanced algorithms were proposed (Holliman and Memon 2000, Fridrich et al. 2000, Wong and Memon 2000). Nonetheless, they either fail to effectively address the problem or sacrifice the tamper localisation accuracy of the original methods (Celik et al. 2002). Celik et al. (2002) then presented an algorithm based on Wong's scheme and

demonstrated that their algorithm can thwart the VQ codebook attack while sustaining the localisation property.

The technique of Yeung and Mintzer (1998), whose security is examined in (Memon et al. 1999), is also one where the correct detection information results in a bi-level image. However, the embedding technique is more extensive than inserting a binary value into the least-significant bit plane. The marking key is used to generate several pseudo-random look-up tables (one for each channel or colour component) that control how subsequent modification of the pixel data will occur. Then, after the insertion process is completed, a modified error diffusion process can be used to spread the effects of altering the pixels, making the mark more difficult to see. As discussed in (Memon et al. 1999), the security of the technique depends on the difficulty of inferring the look-up tables. The search space for the table entries can be drastically reduced if knowledge of the bi-level watermark image is known. A modification (position-dependent lookup tables) is proposed in (Memon et al. 1999) to dramatically increase the search space.

Various transformations, such as the discrete cosine transform (DCT) and wavelet transforms, are widely used for lossy image compression and much is known about how the transform coefficients may be altered (quantized) to minimize perceptual distortion (Wolfgang et al. 1999). There is also a great deal of interest in transform embedding for robust image marking systems to make embedded marks more resilient to attacks.

There are advantages for fragile watermarking systems to use the transform domain. Many fragile watermarking systems are adapted from lossy compression systems (such as JPEG), which have the benefit that the watermark can be embedded within the compressed representation. The properties of a transform can be used to characterise how an image has been damaged or altered. Also, applications may require a watermark to possess robustness to certain types of modification (such as brightness changes) yet be able to detect other modifications (e.g. local pixel replacement). Wu and Liu (1998) describe a technique based on a modified JPEG encoder. The watermark is inserted by changing the quantized DCT coefficients before entropy coding. A special lookup table of binary values (whose design is constrained to ensure mark invisibility) is used to

partition the space of all possible DCT coefficient values into two sets. The two sets are then used to modify the image coefficients to encode a bi-level image (such as a logo.) To reduce the blocking effects of altering coefficients, it is suggested that the DC coefficient and any coefficients with low energy be unmarked.

2.5.2. Examples of Semi-fragile watermarking

A semi-fragile watermark describes a watermark that is unaffected by legitimate distortions, but destroyed by illegitimate distortions. It provides the mechanism for implementing selective authentication. Semi-fragile watermark combines the properties of fragile and robust watermarks. Like a robust watermark, a semi-fragile watermark is capable of tolerating some degree of change to the watermarked image, such as the addition of quantization noise from lossy compression. And like a fragile watermark, the semi-fragile watermark is capable of localising regions of the image that have been tampered with and distinguish them from regions that are still authentic. Thus, a semi-fragile watermark can differentiate between localised tampering and information preserving, lossy transformations. Many fragile watermarking systems perform watermark embedding in the LSB plane and are unable to tolerate a single bit error in this bit. However, the quantization noise introduced by compression is likely to cause many least significant bits to change. Furthermore, recent fragile watermarking systems employ cryptographic hash functions that are not suitable in a semi-fragile framework. A hash function h(x) will produce completely different outputs h(x1) and h(x2) if the binary inputs are distinct but very similar. Even if some characteristic of the image that is expected to remain invariant during lossy compression were hashed, the output of the hash function would have to be embedded in a way that is resilient to errors.

Wolfgang and Delp (1996) extended van Schyndel's work to improve robustness and localisation in their VW2D technique. Adding a bipolar M-sequence in the spatial domain embeds the watermark. Detection is via a modified correlation detector. For localisation, a blocking structure is used during embedding and detection. This mark has been compared to other approaches using hash functions (Wolfgang, Delp 1999).

Fridrich (1998) proposes a similar technique. To prevent unauthorised removal or intentional watermark distortion, the author recommends making the mark dependent on the image in which it is embedded. The binary mark used corresponds to a pseudo-random signal generated from a secret key, the block number and the content of the block represented with an M-tuplet of bits. Each block is then watermarked using O'Ruanaidh (1997) spread spectrum technique. The author claims that the watermark is fairly robust with respect to brightness and contrast adjustment, noise adding, histogram manipulation, cropping and moderate JPEG compression up to 55% quality.

Kundur and Hatzinakos (1998) and Xie and Arce (1998) describe techniques based on the wavelet transform. Kundur embeds a mark by modifying the quantization process of Haar wavelet transform coefficients while Xie selectively inserts watermark bits by processing the image after it is in a compressed form using the SPIHT algorithm (Said 1996). A wavelet decomposition of an image contains both frequency and spatial information about the image. Hence, watermarks embedded in the wavelet domain have the advantage of being able to locate and characterise tampering of a marked image.

2.5.3. Summary of Different Methods

We summarise the different methods presented in this chapter in Table 2.1. The class to which each method belongs is indicated as fragile, semi-fragile, and digital signature, as well as the type of authentication data used and whether the method offers a possible localisation and reconstruction of the areas tampered with. From the table, we notice that, generally, the fragile watermarking methods allow only a strict integrity service, while the semi-fragile watermarking methods and methods based on external signature guarantee a content authentication. It is also interesting to note that few methods are currently able to restore, even partially, the tampered regions of the image.

Method	Class	Mark	Dependent	Integrity	Localization	Recovery
Yeung and Mintzer (1997)	fragile	Predefined logo	no	strict	yes	No
Walton (1995)	fragile	checksums	yes	strict	yes	No
Fridrich and Goljan (1999)	fragile	image comp.	yes	strict	yes	Yes
Wong (1999)	fragile	Hash function	yes	strict	yes	No
Lin and Chang (2000)	semifragile	DCT coeff.	yes	content	yes	Yes
Wolfgang and Delp (1996)	semifragile	m-sequences	no	content	yes	No
Fridrich (1998)	semifragile	Block-based	yes	content	yes	No
Kundur and Hatzinakos (1998)	semifragile	Random noise	no	strict	yes	No
Queluz (2002)	signature	edges	yes	content	yes	No
Bhattacharjee and Kutter (1998)	signature	Interest points	yes	content	yes	No
Lin and Chang (1998)	signature	DCT coeff.	yes	content	yes	No
Wolfgang and Delp (1996)	signature	Hash function	yes	strict	yes	No

Table 2.1 Summary of methods ensuring an authentication service

2.6 Requirements of Watermarking-based Authentication System

A watermarking-based authentication system can be considered as effective if it satisfies the following requirements as outlined by (Tong and Zheng-ding 2002) and (Lin and Chang 2000):

- Invisibility: The embedded watermark is invisible. It is the basic requirement of maintaining the quality of marked images. The marked image must be perceptually identical to the original under normal observation. It is a question of making sure that the visual impact of watermarking is as weak as possible so that the watermarked image remains identical to the original.
- Detect tampering: An authentication watermarking system should detect any tampering in a marked image. This is the most fundamental property to reliably test authenticity of the image. The system must be sensitive to malicious manipulations such as altering the image in specific areas.
- Security: The embedded watermark cannot be forged or manipulated. In such systems, the marking key is private and should be difficult to deduce. Insertion of a mark by unauthorised parties should be difficult.
- Identification of a manipulated area or localisation: The authentication watermark should be able to detect the location of altered areas and verify other areas as authentic. The detector should also be able to estimate what kind of modification has occurred.
- Reconstruction of altered regions: The system should have the ability to restore, even partially, altered or destroyed regions in order to allow the user to know the original content of the manipulated areas.
- Protocols: Protocols are an important aspect of any image authentication. It is obvious that any algorithm alone cannot guarantee the security of the system. It is necessary to define a set of scenarios and specifications describing the operation and rules of the system, such as management of the keys, physical security and the communication protocols between parties and so forth.

We further classify the requirements into mandatory requirements and desirable requirements as seen in Table 2.2.

Classification	Requirements
Mandatory	• Invisibility • Tamper detection • Security
Desirable	• Localize tamper • Reconstruction
Other	• Protocols

Table 2.2 Authentication watermarking requirements

2.7 Main Components of a Watermarking System

A watermarking system can be divided into three main components:

1. The generating function, f_g, of the watermark signal, W, to be added to the host signal. Typically, the watermark signal depends on a key, k, and watermark information, i. Examples of watermark information are company logo and user information.

$$W = f_g(i,k) \tag{2.4}$$

Possibly, it may also depend on the host data, Y, into which it is embedded

$$W = f_g(i,k,Y) \tag{2.5}$$

2. The embedding function, f_m, which incorporates the watermark signal, W, into the host data, Y, yielding the watermarked data Y_w. Typically, the watermark signal depends on a key, K

$$Y_w = f_m(Y,W,K) \tag{2.6}$$

3. The extracting function, f_y, which recovers the watermark information, W, from the received watermarked data, $\hat{Y}_w$, using the key corresponding to embedding and the help of the original host data, Y

$$\hat{W} = f_y(Y, \hat{Y}_w, K) \tag{2.7}$$

Or without the original host data, Y

$$\hat{W} = f_y(\hat{Y}_w, K) \tag{2.8}$$

The first two components, watermarking generating and watermarking embedding, are often regarded as one, especially for methods in which the embedded watermark is independent of the host signal. We separate them out for a better analysis of the watermarking algorithms, since some of the watermark is host signal content dependent, with the watermark generating from the host signal and being embedded back to the host signal.

Figure 2.4 shows the generic watermarking scheme. The inputs to the embedding process are the watermark, the host data, and an optional key. The watermark can take many forms, such as number, text, binary sequence, or image. The key is used to enforce security and to protect the watermark. The output of the watermarking scheme is the watermarked data. The channel for the watermarked image could be lossy and susceptible to malicious attack. The inputs for extraction are the received watermarked data, the key corresponding to the embedding key, and, depending on the method, the original data and/or watermarking information. The output of the watermark recovery process is the recovered watermark. The watermark is inspected to determine if the original image altered and recover information such as copyright status.

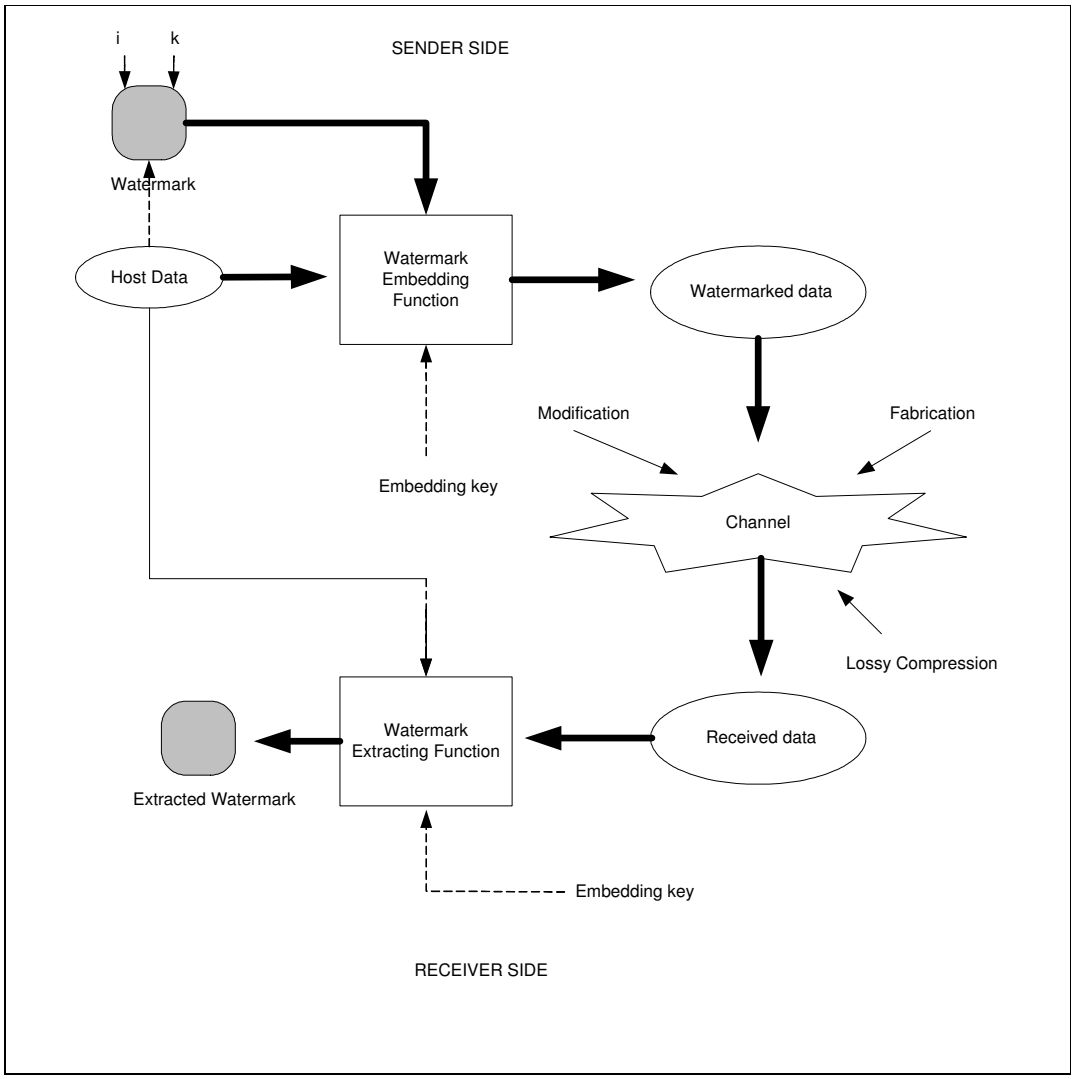


Figure 2.4. Generic Watermarking Scheme

2.8 Malicious Attacks

This section will show some of the most frequent attacks that an image authentication system has to overcome. The common objective of these attacks is to trick the authentication system, in other words, to show that an image remains authentic even though its content has been modified (or sometimes, the opposite).

One of the most common attacks against fragile watermarking systems consists of trying to modify the protected image without altering the embedded watermark, or even

more common, trying to create a new watermark that the authenticator will consider authentic. Take the following simplified example where the integrity of an image is insured by a fragile watermark, independent of the image content and embedded in the LSB of its pixels. We easily see that if we alter the image without modifying the LSB, the watermark will remain as it was, and the authentication process will not detect any falsification. In general, when the integrity of an image is based on a mark that is independent of its content, it is possible to develop an attack that could copy a valid watermark from one image into another image. By doing so, the second image becomes protected even though the second image is false. This attack can also be performed over the same image. First, extract the watermark from the image, then manipulate the image, and finally reinsert the watermark on the altered image. This process will cheat the authentication system. One way to resist this kind of attack is to use a content-based watermarking algorithm or choose a transform domain authentication scheme, which has higher security than a spatial technique.

Another classic attack tries to discover the secret key used to generate the watermark. This kind of attack, also called Brute Force Attack, is well known by the security community. Once the key has been found, it is very easy for an attacker to falsify a watermark of an image that has been protected by this key. The only way to counter this attack is to use long keys to put off the attacker from trying to discover the key, because of the high cost of computing time.

2.9 Embedding Techniques

2.9.1 Least Significant Bit Modification

The most straightforward method of watermark embedding would be to embed the watermark into the least significant bits of the cover object (Johnson and Katzenbeisser 2000). Given the extraordinarily high channel capacity of using the entire cover for transmission in this method, a smaller object may be embedded multiple times. Even if most of these were lost due to attacks, a single surviving watermark would be considered a success.

LSB substitution however, despite its simplicity has many drawbacks. Although it may survive transformations such as cropping, any addition of noise or lossy compression is likely to defeat the watermark. An even better attack would be to simply set the LSB bits of each pixel to one fully defeating the watermark with negligible impact on the cover object. Furthermore, once the algorithm is discovered, an intermediate party could easily modify the embedded watermark.

LSB modification proves to be a simple and fairly powerful tool, however lacks the basic robustness that watermarking applications require.

2.9.2 Correlation-Based Techniques

Another technique for watermark embedding is to exploit the correlation properties of additive pseudo-random noise patterns as applied to an image (Langelaar et al. 2000). A pseudo-random noise (PN) pattern *W(x, y)* is added to the cover image *I(x, y)*, according to the equation shown below in equation 2.9.

$$I_w(x,y) = I(x,y) + k * W(x,y) \tag{2.9}$$

In equation 2.9, k denotes a gain factor, and I_W the resulting watermarked image. Increasing k increases the robustness of the watermark at the expense of the quality of the watermarked image. Rather than determining the values of the watermark from "blocks" in the spatial domain, we can employ CDMA spread-spectrum techniques to scatter each of the bits randomly throughout the cover image, increasing capacity and improving resistance to cropping (Langelaar et al. 2000). To detect the watermark, each seed is used to generate its PN sequence, which is then correlated with the entire image. If the correlation is high, then that bit in the watermark is set to "1", otherwise a "0". The process is then repeated for all the values of the watermark. CDMA improves on the robustness of the watermark significantly, but requires several orders more of calculation.

2.9.3 Frequency Domain Techniques

The classic and still most popular domain for image processing is that of the Discrete-Cosine-Transform, or DCT. The DCT allows an image to be broken up into different frequency bands, making it much easier to embed watermarking information into the middle frequency bands of an image. The middle frequency bands are chosen so that they minimise effects on the most visually important parts of the image (low frequencies) without being removed through compression and noise attacks (high frequencies).

One such technique utilizes the comparison of middle-band DCT coefficients to encode a single bit into a DCT block. To begin, we define the middle-band frequencies (F_M) of an 8x8 DCT block as shown below in figure 2.5.

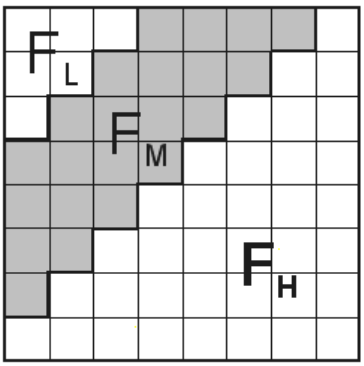


Figure 2.5. Definition of DCT Regions

F_L is used to denote the lowest frequency components of the block, while F_H is used to denote the higher frequency components. F_M is chosen as the embedding region as to provide additional resistance to lossy compression techniques, while avoiding significant modification of the cover image (Hernandez et al. 2000).

2.9.4 Wavelet watermarking

The wavelet transform provides another possible domain for watermark embedding. The DWT (Discrete Wavelet Transform) separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components. The process can then be repeated to compute multiple "scale" wavelet decomposition, as in the 2-scale wavelet transform shown below in figure 2.6.

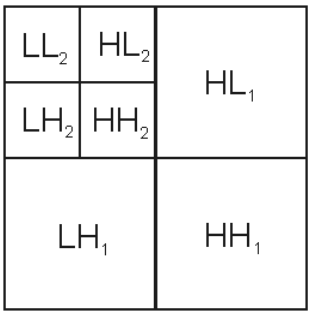


Figure 2.6. 2 Scale 2-Dimensional Discrete Wavelet Transform

One of the many advantages over the wavelet transform is that it is believed to model more accurately aspects of the human visual system (HVS) as compared to the FFT or DCT. This allows us to use higher energy watermarks in regions that the HVS is known to be less sensitive to the eye, such as the high resolution detail bands {LH, HL, HH). Embedding watermarks in these regions allow the robustness of the watermark to be increased (Langelaar et al. 2000).

One of the most straightforward techniques is to use a similar embedding technique to that used in the DCT, the embedding of a CDMA sequence in the detail bands according to the equation shown below in equation 2.10,

$$I_{Wu,v} = \begin{cases} W_i + \alpha |W_i| x_i, & u,v \in HL, LH \\ W_i & u,v \in LL, HH \end{cases} \tag{2.10}$$

where W_i denotes the coefficient of the transformed image, x_i the bit of the watermark to be embedded, and a scaling factor. To detect the watermark we generate the same

pseudo-random sequence used in CDMA generation and determine its correlation with the two transformed detail bands. If the correlation exceeds some threshold T, the watermark is detected.

This can be easily extended to multiple bit messages by embedding multiple watermarks into the image. As in the spatial version, a separate seed is used for each PN sequence, which is then added to the detail coefficients as in equation 2.5. During detection, if the correlation exceeds T for a particular sequence a "1" is recovered; otherwise a zero. The recovery process then iterates through the entire PN sequence until all the bits of the watermark have been recovered.

Chapter 3

Medical Image Watermarking

3.1 Introduction

This chapter discusses the properties of medical image watermarking and discusses techniques available for tamper localisation and image reconstruction. This chapter is structured as follows:

- Section 3.2 highlights the properties of medical image watermarking and outlines the objectives for watermarking in medical domain.
- Section 3.3 introduces reversible watermarking and describes such a scheme.
- Section 3.4 introduces the concept of the region of interest (ROI) in medical images.
- Section 3.5 discusses authentication watermarking with localisation capabilities and the security risk of such techniques.
- Section 3.6 presents a few techniques available for using watermark as an aid in the reconstruction of image that have been corrupted.
- Section 3.7 gives a review of previous work done on medical images.

- Section 3.8 introduces Digital Imaging and Communications in Medicine (DICOM) standard and Picture Archiving and Communication System (PACS).
- Section 3.9 discusses methods for evaluating perceptual impacts of watermarks.

3.2 Properties of Medical Image Watermarking

Security of medical information, derived from strict ethics and legislative rules, gives rights to the patient and duties to the health professionals. This imposes three mandatory characteristics: confidentiality, reliability and availability:

- Confidentiality means that only the entitled persons have access to the information and that information is not made available or disclosed to unauthorised individuals, entities or processes
- Reliability which has two aspects; Integrity: the information has not been modified or destroyed by non-authorized person, and authentication: proof that the information belongs indeed to the correct patient and is issued from the correct source
- Availability is the ability of an information system to be used by the entitled persons in the normal conditions of access and exercise.

Security risks of medical images can vary from random errors occurring during transmission to lost or overwritten segments in the network during exchanges in the intra- and inter-hospital networks. One must also guarantee that the header of the image file always matches that of the image data. In addition to these unintentional modifications one can envision various malicious manipulations to replace or modify parts of the image, called tampering. The usual constraints of watermarking are invisibility of the mark, capacity, secrecy to unauthorised persons, and robustness to attempts to suppress the mark. These demands also exist in the medical domain but additional constraints are added. Three main objectives are foreseen in the medical domain (Coatrieux et al. 2000, Mintzer et al. 1997):

1. Imperceptible / Reversible Watermarking

Medical tradition is very strict with the quality of biomedical images. Thus the watermarking method must be reversible, in that the original pixel values must be exactly recovered (Macq and Dewey 1999). This limits significantly the capacity and the number of possible methods.

An alternative way is to define regions of interest, to be left intact, and leave us with regions of insertion where a watermark could be inserted and does not interfere or disturb the radiologist.

2. Integrity Control

The "secure camera" concept applies also to biomedical images, especially in the context of legal aspects and insurance claims. There is thus a need to prove that the images on which the diagnoses and any insurance claims are based have preserved their integrity.

3. Authentication

A critical requirement in patient records is to authenticate the different parts of the electronic patient record, in particular the images. More often an attached file or a header, which carries all the needed information, identifies an image. However, keeping the meta-data of the image in a separate header file is prone to forgeries or clumsy practices. An alternative would be to embed all such information into the image data itself.

3.3 Reversible Watermarking

Reversible watermarking means that the original data will be available after a watermark is embedded. In summary any reversible watermarking system comprises the following steps:

i. Embedding a digital watermark, w in an original image x resulting in $y = f(x, w)$

ii. Transmitting the watermarked image y from the encoder to the decoder through an error-free transmission channel

iii. Extracting the watermark image w and restoring the original image $x = f^{-1}(y, w)$

The concept for reversible data embedding first appeared in an authentication method for images in a patent from the Eastman Kodak Company (Honsinger et al. 2001). There are several techniques for reversible data embedding and the scheme proposed by Goljan et al (2001) will be described.

Let us assume that the original image is a greyscale image with M x N pixels and with pixel value from the set P. For example, for an 8-bit greyscale image, P= {0, …, 255}. They start with dividing the image into disjoint groups of n adjacent pixels (x_1, …, x_n). For example we can choose groups of n = 4 consecutive pixels in a row. They also define so called discrimination function f that assigns a real number $f(x_1,...,x_n) = f(G) \in \Re$ to each pixel group G = (x_1, …, x_n). The purpose of the discrimination function is to capture the smoothness or regularity of the group of pixels G. Discrimination function used was:

$$f(x_1, x_2,..., x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Then an invertible operation F on P called 'flipping' is defined. Flipping is a permutation of grey levels that consists of 2-cycles. Thus, F will have the property that F^2 = Identity or $F(F(x)) = x$ for all $x \in P$. For example, the permutation F_{LSB} defined as 0 ↔ 1, 2 ↔ 3, …, 254↔255 correspond to flipping the LSB of each grey level. The permutation 0↔2, 1↔3, 4↔6, 5↔7, … corresponds to an invertible noise with larger amplitude. Discrimination function f and the flipping operation F were used to define three types of pixel groups: R, S and U.

Regular groups: $G \in$ R if $f(F(G)) > f(G)$

Singular groups: $G \in$ S if $f(F(G)) < f(G)$

Unusable groups: $G \in$ U if $f(F(G)) = f(G)$

3.4 Region of Interest (ROI)

Typically, a medical image is diagnosed before being archived in long-term storage, so the significant part of the image is already determined. The significant part is called ROI (Region Of Interest), which must be preserved without any lack of information. In Chapter 4, we propose a strict authentication watermarking considering ROI. In general, the ROI is stored as it is or compressed by a lossless algorithm and the other part is compressed by a lossy algorithm, which can achieve a higher compression rate than lossless compression algorithm (Wakatani 2002).

Distant learning is one of applications using a database of medical images, which may refer to the image of a newly discovered medical case, and there may be images with the ROI part for long-term storage. Therefore, it is desirable that the copyright and integrity of the medical image with ROI part are protected. However it is impossible to embed signature information into the ROI part since the ROI must be kept without any distortion.

3.5 Localisation and Security Risk

Many authentication methods based on watermarking have the ability to identify regions of the image that have been tampered with, while verifying that the remainder of the image has not been changed. This capability is referred to as localisation. Localisation is useful because knowledge of where an image has been tampered with can be used to infer: 1) the motive for tampering; 2) a possible attacker; and 3) whether the alteration is legitimate. For example, consider an ultrasound image of a kidney. If our authenticator simply states that the image has been modified, the tampered image is useless. However, if the authenticator also indicated that the modification only occurred within the region of non -interest, the image is still very useful for learning purposes.

Most localised authentication methods rely on some form of block-wise authentication, in which the image is divided into a number of spatial regions, each of which is authenticated separately. If part of the image is modified, only the affected regions fail to authenticate.

There are a number of security risks associated with localised authentication systems. Although the risks discussed here can be countered with simple modifications, it is important to be aware of them. We are concerned with forgery attacks in which an attacker wishes to embed a valid watermark into either a modified or false image. Two basic attacks will be examined. In search attacks, the attacker is assumed to have a detector that can determine whether the image is authentic or not. In collage attacks, the attacker is assumed to have two or more images embedded with the same watermark.

3.5.1 Search Attacks

Let us consider the situation in which everyone, including potential attackers, has access to a watermark detector. This situation might arise, for example, if images are being distributed to the public over the Internet, and an authentication system is to be used to guarantee that each image is delivered without corruption or tampering. In theory, the attacker can use the detector to defeat any authentication system, regardless of whether it is localised or not. To do so requires a brute-force search. To embed a forged watermark into an image, the attacker can enter slightly modified versions of the image into the detector until one is found that the detector reports as authentic.

In practice, this search would usually be prohibitive. However, with a block-wise authentication system, the search space can be considerably smaller. The attacker can perform a separate, independent search on each block. If the block size is small enough, the search becomes feasible. Such attacks can be countered by choosing a sufficiently large block size.

3.5.2 Collage Attacks

The second category of attacks relies on having access to one or more authentic watermarked images. By examining these images, an attacker can come up with sets of blocks that are authentic and construct a forged image from them like a 'collage'. Holliman and Memon (2000) describe an attack applicable to block-wise watermarks, in which a cryptographic signature is embedded in each block and the signature depends only on the content of the block itself. Consider what happens in such a system when two blocks of a watermarked image are interchanged, thereby changing the image as a whole. Because each block contains a self-authenticating watermark, and because each block remains unaltered, the image is deemed authentic. Thus, even if all blocks are scrambled into a random order, the system will regard the entire image as authentic.

By exploiting this weakness of block-wise independent system, it is possible to create a completely new image that is assembled from the set of independent, authentic blocks. Suppose an attacker has a number of images available, all watermarked using the same key, this can be viewed as a large database of authentic blocks from which a new image can be built. To forge a watermark in an unwatermarked image, the attacker divides the image into blocks and replaces each block with the most similar block from the database. With a large enough database of watermarked images, the results may be quite effective.

The solution to counter these types of attacks is to use a different key for watermarking every image. However, such an approach is not always feasible, in that a given image can only be authenticated if the correct key is available. The keys would need to be either known to the users or stored as associated data.

A more practical approach suggested in Holliman and Memon (2000) is to make the blocks overlap so that the signature of each block depends on surrounding data, as well as the data within the block itself. This introduces ambiguity to the localisation, because a change in one block will change the signature that must be embedded in its neighbours. This complicates the attacker's attempt to build an image out of

watermarked blocks, as each block must match up properly with the neighbouring blocks.

3.6 Restoration

From the previous section we have seen that it is possible to verify if an image has been altered and determine where it has been altered. This section considers the manner in which an altered or tampered image might be restored.

There are two restoration strategies: exact restoration and approximate restoration. In exact restoration the image is restored to its original state, where the goal is to create a perfect copy. This is a well-studied problem in communication and will be discussed in the next section. Approximate restoration is a more recent concept that seeks to restore an image to approximately the original state while accepting that there will be differences between the restored and original image. However, the restored image may still be valuable if these differences are not significant.

3.6.1 Embedded Redundancy

It is well known that error detection and error correction codes allow changes in data to be detected, and in the latter to be corrected. Error correction codes (ECC) are widely used in communication and data storage to maintain the integrity of digital data. ECC codes are like digital signatures, are appended to the data. However, there are important differences between ECC codes and signatures:

1. Digital signatures are used to verify that the data has not been altered.
2. Digital signatures need fewer bits than ECC to detect a change has occurred than are needed to perform correction (Shannon 1948).
3. ECC codes usually assume a maximum number of bit changes. If this number is exceeded, it is possible for errors to go undetected

The size of an ECC code is usually very much larger than a digital signature. In fact, an ECC code can represent a significant fraction of transmitted bits. The size of the ECC

code determines both the maximum number of bit changes that can be detected and the maximum number of bits that can be corrected.

An image can be considered as a collection of bits, and a variety of different error correction codes can be applied (e.g., Hamming codes, turbo codes, and trellis codes). This metadata can be represented as watermark. For example, a Reed Solomon ECC code can be used to generate parity bytes for each row and column of an image (Lee and Won 1999, Lee and Chen 2002). These parity bytes can be embedded as a watermark in the two significant bit planes of the image. It is reported that for a 229 X 229 image, up to 13 bytes in a single row or column can be corrected. Even if the errors cannot be corrected, they can be localised, because parity bytes are calculated for each row and column.

This method is modified when it is expected that errors will come as bursts (Lee and Won 2000). This is the case when a localised region of an image has been modified or cropped. To increase the resistance to burst errors, the locations of the pixels in the image are randomised prior to calculation of the ECC codes. This randomisation is a function of a watermark key.

If we want to restore an image to its original state, a very significant cost must be incurred to store the ECC codes. If this cost is too high, or the resources are simply unavailable, then approximate restoration techniques may be a good compromise.

3.6.2 Self-embedding

A further approach is self-embedding (Lin and Chang 2000, Fridrich and Goljan 1999), which is a highly compressed version of the image in the image itself. Thus, if portions of the watermarked image are removed or destroyed, these modified regions can be replaced with their corresponding low-resolution versions.

In the algorithm of Fridrich and Goljan (1999), a highly compressed JPEG (50% quality factor) version of the image is produced. This low-resolution image requires only one bit per pixel and can thus be inserted in the LSB plane of the image. However, each

compressed DCT block is not simply inserted into the LSB of its corresponding spatial block; rather the binary sequence is first encrypted and then inserted in the LSB plane of a block that is some distance away and in a randomly chosen direction. The authors suggest a minimum distance of 3/10 the image size. The random mapping is generated using a key that must be known to both the embedder and the detector. Storing the low-resolution version of a block some distance away from the corresponding block allows this block to be restored even if it has been completely deleted. A higher quality reconstruction is possible if more bits are allocated to the storage of the low-resolution image. The method is severely affected by any modifications to the encoding bit plane.

3.6.3 Blind Restoration

An alternative approach to approximate correction of errors is based on blind restoration. Blind restoration attempts to first determine what distortions an image has undergone, and then to invert these distortions to restore the image to its original state (Kundur and Hatzinakos 1996). Such a process is only appropriate if the distortion is invertible. Thus, blind restoration is not useful against, for example, clipping.

The method assumes that the image and the watermark undergo the same distortion. If the watermark is made capable of determining the distortion that has occurred, then an inverse process (assuming such a process exists), can be applied to restore the watermark and the image. A combination of blind restoration and self-embedding may also be appropriate. In principle, blind restoration might allow a lower resolution image to be embedded. This is because at least some of the distortion may be invertible. In addition, where clipping or other non-invertible distortions have been applied, the self-embedded information allows for a low-resolution restoration.

3.7 Previous Work on Medical Image Watermarking

Digital watermarking can imperceptibly embed messages without changing image size or format. When applied to medical images, the watermarked image can still conform to the DICOM format (Guo and Zhuang 2003). Some researchers already apply watermarking technique to medical data. Zhou et al (2001) present a watermarking

method for verifying the authenticity and integrity of a digital mammography image. They used a digital envelope as a watermark and the least significant bits (LSB) of one random pixel of the mammogram are replaced by one bit of the digital envelope (DE) bit stream. Instead of the whole image data, only partial image data (i.e., the most significant bits (MSB) of each pixel is used for verifying integrity). Cao et al (2003) extend their work on digital envelopes and embed their DE by making a random walk sequence and replacing the LSB of each selected pixel.

Other researchers adapt digital watermarking for interleaving patient information with medical images to reduce storage and transmission overheads (Acharya et al 2001). Again, the LSB of image pixels are replaced for embedding. Chao et al (2002) propose a discrete cosine transform (DCT) based data-hiding technique that is capable of hiding those EPR related data into a marked image. The information is embedded in the quantized DCT coefficients. The drawback of the above watermarking approaches is that the original medical image is distorted in a non-invertible manner. Therefore it is impossible for a watermark decoder to recover the original image.

A reversible watermarking scheme involves inserting a watermark into the original image in an invertible manner, so that when the watermark was later extracted, the original image can be recovered completely. Research has also been done in the area of reversible watermarking in medical images. Trichili et al (2002) proposes an image virtual border as the watermarking area. Patient data is then embedded in the LSBs of the border. Guo and Zhuang (2003) present a scheme where the digital signature of the whole image and patient information is embedded. They define three types of pixel groups as suggested by Goljan et al (2001), R, S, and U. The problem with this technique is that the capacity for embedding is highly dependent on the number of R and S group of pixels. The maximum number of bits available for embedding in Guo and Zhuang's (2003) scheme for ultrasound images of 640x480x8 bits is 1668 bits. This will give an embedding rate of 0.0054 bits/pixel.

Cho et al (2001) studied watermark methods appropriate for medical images and conclude that the spatial watermark method such as LSB had the advantage that it did

not damage the important information if the watermark was embedded outside the region of interest.

3.8 DICOM and PACS

The initial goal in developing a standard for the transmission of digital images was to enable users to retrieve images and associated information from digital imaging equipment in a standard format that would be the same across multiple manufacturers. The first result was the American College of Radiology (ACR)-National Electrical Manufacturers' Association (NEMA) standard, which specified a point-to-point connection. The rapid evolution of computer networking and of picture archiving and communication systems meant that this point-to-point standard would be of limited use. Consequently, a major effort was undertaken to redesign the ACR-NEMA standard by taking into account existing standards for networks and current concepts in the handling of information on such networks. The Digital Imaging and Communications in Medicine (DICOM) standard was the result of this effort. Its popularity has made discussion, if not implementation, of the standard common whenever digital imaging systems are specified or purchased.

The use of DICOM has now extended beyond only an image and has been adapted to manage data from many medical specialties (e.g., pathology, ECG). It is also a global standard being adopted by the European standards organization, the Comitee Europeân de Normalisation (CEN), as MEDICOM standard. In Japan, the Japanese Industry Association of Radiation Apparatus and the Medical Information Systems Development Centre have adopted portions of DICOM that pertain to the exchange of images on removable media and are considering DICOM for future versions of the Medical Image Processing Standard. The DICOM standard is now being maintained and extended by an international, multi-specialty committee (Horii 1997).

The DICOM standard has become the predominant standard for the communication of medical images. The DICOM standard consists of multiple documents (National Electrical Manufacturers Association 2003), which at the time of writing consist of 16

published parts. Each DICOM document is identified by a title and standard number, which takes the form "PS 3.X-YYYY," where "X" is commonly called the part number and "YYYY" is the year of publication. For example, DICOM Part 2 has a title of "Conformance" and document number PS 3.2-2003. In informal usage, the year is often dropped. Watermarking is not currently considered in any part of this standard.

Picture archiving and communication system (PACS) is a work flow-integrated system for managing medical image and related data. It is designed to streamline operations throughout the whole patient care delivery process (Huang 2003). PACS was originally developed for radiology services over 20 years ago to capture digital medical images rather than in film-based media. Figure 3.3 describes the enterprise level web-based image/data EPR server with archive.

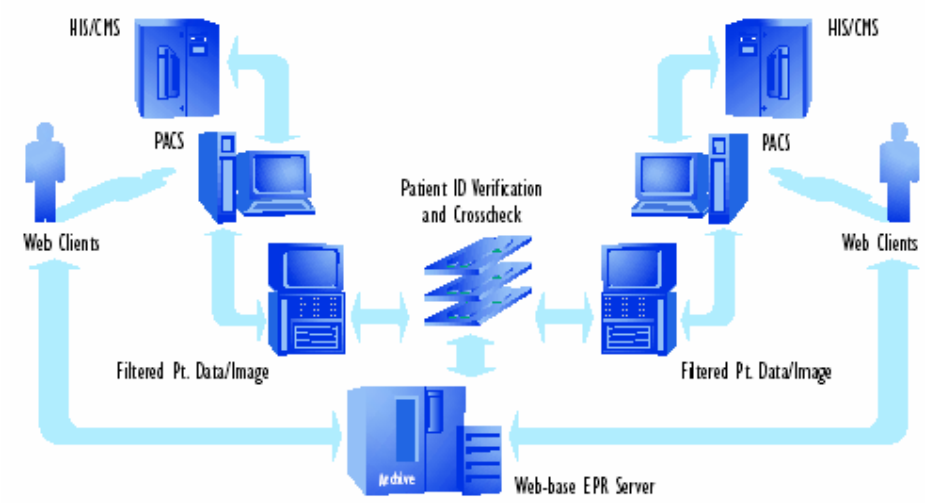


Figure 3.3 Enterprise Level Web-based Image/Data EPR server with archive

3.9 Evaluating Perceptual Impact of Watermarks

There is few, if any, watermarking systems producing watermarks that are perfectly imperceptible. However, the perceptibility of a given system's watermark may be high or low compared against other watermarks or other types of processing, such as compression. In this section we address the question of how to measure that

perceptibility, so that such comparison can be made. This section begins with a discussion of two types of perceptibility that exist as causes for concern.

3.9.1 Fidelity and Quality

In the evaluation of a watermarking system, there are two different types of perceptibility that can be judged: fidelity and quality. Fidelity is a measure of the similarity between images before or after watermarking (Cox et al. 2002). A high fidelity reproduction is a reproduction that is very similar to the original. A low fidelity reproduction is dissimilar or distinguishable from the original. Quality on the other hand is an absolute measure of appeal. A high quality image simply looks good. It has no obvious processing artefacts. Both types of perceptibility are significant in evaluating watermarking systems.

To explore the difference between fidelity and quality, consider an example of video from a surveillance camera. The video is typically greyscale, low resolution, compressed and generally considered to be of low quality. Consider a watermarked version of this video that looks identical to the original. This watermarked video must also have low quality, but as it is indistinguishable from the original, it has high fidelity.

For some watermarking applications, fidelity is the primary perceptual measure of concern. In these cases, the watermarked image must be indistinguishable from the original. For example, a patient may require this of a watermark applied to his medical images.

3.9.2 Human Evaluation Measurement Techniques

Although the claim of imperceptibility is often made in the watermarking literature, rigorous perceptual quality and fidelity studies involving human observers are rare. Some claims of imperceptibility are based on automated evaluations, discussed in section 3.9.3. However, many claims are based on a single observer's judgements on a

small number of trials. These empirical data points are not sufficient for proper perceptual evaluation or comparison of watermarking algorithms.

An experimental paradigm for measuring perceptual phenomena is the two alternatives forced choice (2AFC) (Green and Swets 1974). In this procedure, observers are asked to give one of two alternative responses to each of several trial stimuli. For example, to test the quality impact of a watermarking algorithm, each trial of the experiment might present the observer with two versions of one image. One version of the image would be the original, the other would be watermarked. The observer, unaware of the differences between the images, must decide which one is higher in quality. In the case where no difference in quality can be perceived, the responses are expected to be random. Random choices suggest that observers are unable to identify one selection as being consistently better quality than the other. Thus, 50% correct answers correspond to zero JND, while 75% correct corresponds to one JND (Cox et al. 2002).

The 2AFC technique can also be used to measure fidelity. Consider an experiment in which the observer is presented with three images (Figure 3.4). One is labelled as the original. Of the other two, one is an exact copy of the original and the other is the watermarked version. The subject must choose which of the two latter images is identical to the original. The results are tabulated and examined statistically. Any bias in the data represents the fact that the observers could distinguish between the original and watermarked images, and serves as a measure of the fidelity of the watermarking process.

A second, more general experimental paradigm for measuring quality allows the observers more latitude in their choice of responses. Rather than selecting one of two images as ‘better’, observers are asked to rate the quality of an image, sometimes with a reference to a second image. For example, the ITU-R Rec. 500 quality rating scale specifies a quality scale and an impairment scale that can be used for judging the quality of television pictures (ITU 2000). These scales, summarised in Table 3.1, have been suggested for use in the evaluation of image watermarking quality (Kutter and Hartung 2000).

Original

A

B

Figure 3.4 A two alternative, forced choice experiment studying image fidelity.

Five-Grade Scale	
Quality	Impairment
5 Excellent	5 Imperceptible
4 Good	4 Perceptible, but not annoying
3 Fair	3 Slight annoying
2 Poor	2 Annoying
1 Bad	1 Very annoying

Table 3.1 Quality and impairment scale as defined in ITU-R Rec. 500

3.9.3 Automated Evaluation

The experimental techniques outlined previously can provide very accurate information about the fidelity of watermarked content. However, they can be very expensive and are not easily repeated. An alternative approach is the use of an algorithmic quality measure based on a perceptual model. The goal of a perceptual model is to predict the response of an observer. The immediate advantages of such a system are that it is cheaper and faster to implement and the evaluation can be repeated so that different methods can be compared directly.

Ideally, a perceptual model (function) intended for automated fidelity tests should predict the results of tests performed with human observers. However, for the purposes of comparing the fidelity of different watermarking algorithms, or watermarking strength, it is sufficient for the model to provide a value that is related to the results of human tests, that is to produce a measure of the perceptual distances between watermarked and unwatermarked images (Cox et al. 2002). One of the simplest distance functions is the mean squared error (MSE). This is defined as:

- The mean square error (MSE),

$$MSE = \frac{1}{n}\sum_{i}^{n}(I_i' - I_i)^2 ,$$

which is the averaged term by term difference between the original image, *I*, and the watermarked image, *I'*. Although MSE is often used as a rough test of a watermarking

system's fidelity impact, it is known to provide a poor estimate of the true fidelity (Girod 1993).

Some perceptual distance functions are asymmetric. In these functions, the two arguments have slightly different interpretations. By convention, the first argument is interpreted as an original image, and the second as a watermarked version of it. For example, one commonly used asymmetric distance is based on the reciprocal of the signal-to-noise ratio (SNR). This is defined as:

- The signal-to-noise ratio (SNR),

$$SNR(dB) = 10\log_{10}\frac{\sum_{i}^{n} I_i^{\,2}}{\sum_{i}^{n}(I_i^{'} - I_i)^2},$$

- The peak signal to noise ratio (PSNR),

$$PSNR(dB) = 10\log_{10}\frac{\max I^2}{MSE},$$

where max I is the peak value of the original image (usually 255 for 8 bit grey-scale image). The PSNR of an image is a typical measure used for assessing image fidelity by considering that the just noticeable distortions are uniform in all coefficients in a specific domain, such as spatial domain, frequency domain, or some other transform domain. It is well known that these distance functions are not well correlated with the human visual system (Kutter and Hartung 2000). In this thesis, PSNR is used as a measure of image fidelity.

There are a few assumptions that were made:

- All images will be stored in their original sizes.
- All tampering is done locally, using image editing software and includes:
    - Cut and paste (including cutting from another watermarked image)
    - Cloning
    - Healing brush

Chapter 4

Strict Authentication Watermarking (SAW)

4.1 Introduction

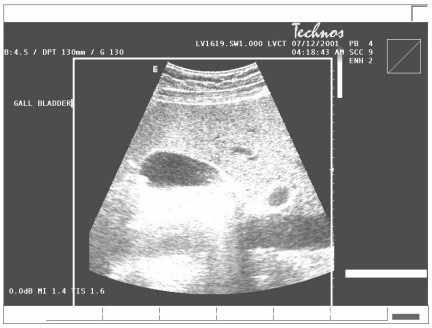
This chapter proposes two types of strict authentication watermarking for medical images. This chapter is structured as follows:

- Section 4.2 proposes strict authentication watermarking for ultrasound images. In this scheme, we define region of interest (ROI) by taking the smallest rectangle around an image. The watermark is generated from hashing the area of interest. The embedding region is considered to be outside the region of interest as to preserve the area from distortion as a result from watermarking.

- Section 4.3 proposes another strict authentication watermarking that is robust to some degree of JPEG compression (SAW-JPEG). JPEG compression will be reviewed. To embed a watermark in the spatial domain, we have to make sure that the embedded watermark will survive JPEG quantization process.

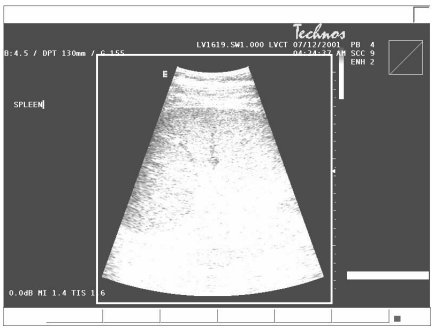
4.2 Strict Authentication Watermarking (SAW)

We propose a strict authentication watermarking for ultrasound images, where the watermark embedded will remain lossless in storage and transmission. In order to reduce the effects on the image, it is to be embedded into a constrained area that is defined to be outside the ROI. In this scheme, we define region of interest (ROI) by taking the smallest rectangle around an image (figure 4.1). This border will be used for our watermark embedding later. The watermark is generated from hashing the area of interest. The embedding region is considered to be outside the region of interest as to preserve the area from distortion as a result from watermarking. Our scheme is an enhancement of the scheme proposed by Cao et al (2003) with reversible capability.

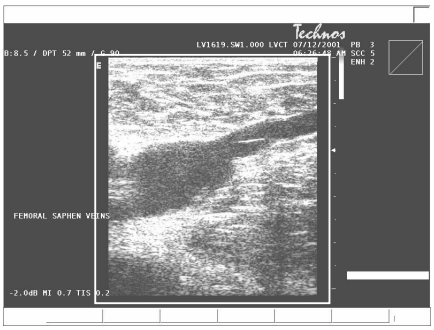


(a) gallbladder

(b) kidney



(c) spleen



(d) vein

Figure 4.1. Ultrasound images with a border drawn around them

4.2.1 Watermark

The watermark is generated by creating a hash value from the region of interest (inside the rectangle), *X* of size *m x n*. .The pixels will be arranged in a string, *S*.

$$S = B(X_{(1,1)}X_{(1,2)}...X_{(1,m)}X_{(2,1)}...X_{(m,n)}), \tag{4.1}$$

where X_{mn} is the 8 bit binary value of each pixel.
The hash value is obtained by applying a hash function to the string

$$Hash = \mathrm{H}(S) \tag{4.2}$$

where H is any hash function such as MD5 and SHA256.

4.2.2 Embedding Region and Domain

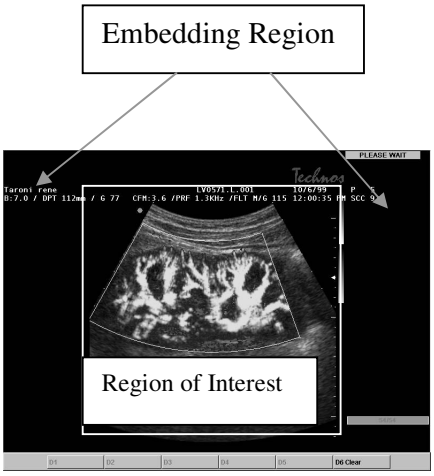


Figure 4.2. Embedding region

The embedding region is considered to be outside the region of interest in order to prevent distortion to the area as a result of adding the watermark. In an ultrasound image, the embedding region is normally a dark region with pixel values 0. This feature will be exploited to create a reversible or invertible watermarking.

In strict authentication watermarking, it is vital that the system will detect any change to the image. Fragile watermarking is the most appropriate as any change in the image will also affect the watermark. Least Significant Bit (LSB) watermarking has an advantage as the method of choice, as it is well known that LSB is vulnerable and easy to manipulate.

4.2.3 Security

A watermark is secure if it is able to resist intentional tampering by an attacker. This would include remaining secure even when the attacker knows the algorithm for embedding and extracting the watermark.

The strength of the security of the watermark will depend on the key chosen. A typical attack would involve removing the watermark, changing the image, then recalculating and embedding the new hash value into the embedding area. If the key for calculating the hash value remains secret, then the system may be considered secure. The secret key can be used to create the hash value and to create a random embedding. These will be examined in turn.

- Key for hashing

A key can be used to create the hash for the selected region. In this method, the sender and recipient will use the same key to carry out the hash function. The hash value obtained will be used as the watermark. At the recipient end, the key will be used to carry out the hash function on the received image and the hash value will be compared with the hash extracted.

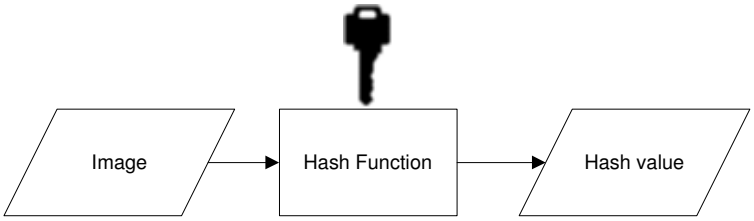


Figure 4.3. Key for hash

- Key for embedding

A key used for embedding will determine the random mapping of watermark values into the embedding region as in figure 4.4.

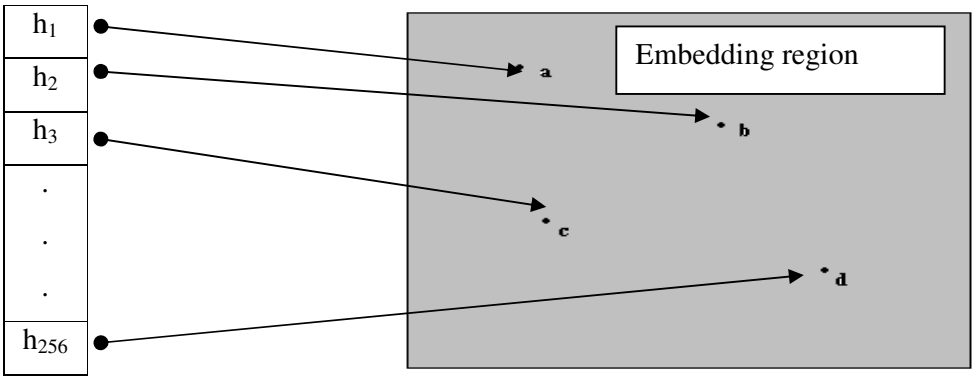


Figure 4.4. Hash value mapping in the embedding region

This supposes that the number of points or pixels in the embedding region is greater than or equal to the number of bits in the hash value holds. As an example, suppose the pixels are arranged as a simple raster scan as in figure 4.5.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Figure 4. 5. Embedding region of 5 x 4 pixels

which may be described by the mapping function of equation 4.3:

$$f(x) = x \bmod n \tag{4.3}$$

where x is the bit position and $x \in \{1, h\}$ and n is the total number of pixels available for embedding. In this example, we use h=20 to make full use of the embedding region. Applying equation 4.3, bit position one will be located in pixel number one, bit position

two will be located in pixel number 2 and so on. By using a key, *k*, the position will be randomised. If a simple function, e.g. equation 4.4 is applied,

$$f(x) = kx \bmod n \tag{4.4}$$

where k is a prime key, then the mapping will be a randomised one-to-one mapping. This is illustrated for k=37 and n=20 in table 4.1.

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F(x)	18	15	12	9	6	3	20	17	14	11	8	5	2	19	16	13	10	7	4	1

Table 4.1. Mapping for k=37, n=20

The method may be extended so that a number, h, of hash values are distributed within a region having many more pixel points, n so that the results appears as a sparse random distribution. The method is illustrated for k=37, h= 20 and n=100 in table 4.2.

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
f(x)	38	75	12	49	86	23	60	97	34	71	8	45	82	19	56	93	30	67	4	41

Table 4.2. Mapping for k=37, h=20,n=100

If the embedding region is 10 x 10 pixels, then the distribution of embedding will be pictured as in figure 4.6.

			19				11		
	3							14	
		6							17
			9				1		
20				12				4	
					15				7
						18			
10				2					
	13				5				
		16				8			

Figure 4.6. Distribution of embedding for k=37, h=20, n=100

This simple method relies on the use of symmetric keys, which has an associated problem of key management. This is beyond the scope of this research. In practice asymmetric key systems are favoured; these are discussed in the next section.

4.2.4 Hashing – SHA256

The Secure hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS PUB 180) in 1990. The algorithm is an iterative, one-way hash function that can process a message to produce a condensed representation called a *message digest*. The algorithm enables the integrity of a message to be determined and any change to the message will, with a very high probability, result in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes. It is based on a public/private key, and thus overcomes the problem of key management.

4.2.5 Method

SHA-256 may be incorporated into a watermarking algorithm as shown in Figure 4.7. The general methodology and principles as listed below:

At sender site

1) **Define Area**: The Region of Interest (ROI) is determined as the smallest rectangle that bounds the known image area. Figure 4.1 shows an example of a rectangle defining the ROI in an ultrasound image.
2) **SHA-256**: The hash value for the whole image using SHA-256 is calculated. This produces a 256-bit one-way hash value that can be the basis of the watermark.
3) **Embed**: The hash value is embedded into the Region of Non-Interest (RONI) in the LSB. The specific location is not important, as it is known that it will not affect the image under any circumstances.

At receiver site:

1) **Extract watermark**: The watermark is extracted by recovering the LSB from the watermarking area.
2) **Flipping**: In flipping, the LSB in the watermarking area are reset to their original values. This acts as the reversible function and is possible for any image that has an area of known constant. In the case of ultrasound images, this may be easily achieved by resetting all the bits to zero.
3) **SHA-256**: the SHA-256 algorithm is applied to the received image and the hash value computed.
4) **Authentication**: the hash value calculated in step 3 is compared to that extraction in step 2. If found to be the same, the image is authenticated.

4.2.6 Experimental Results

An 800 x 600 pixels ultrasound image was watermarked using the method described in section 4.2.5. The watermarked image was modified using the cloning tools of Adobe Photoshop CS2. The cloning area was around 50x50 pixels and the change may be seen as the image in figure 4.9. Figure 4.8 shows the results of hashing using SHA-256.

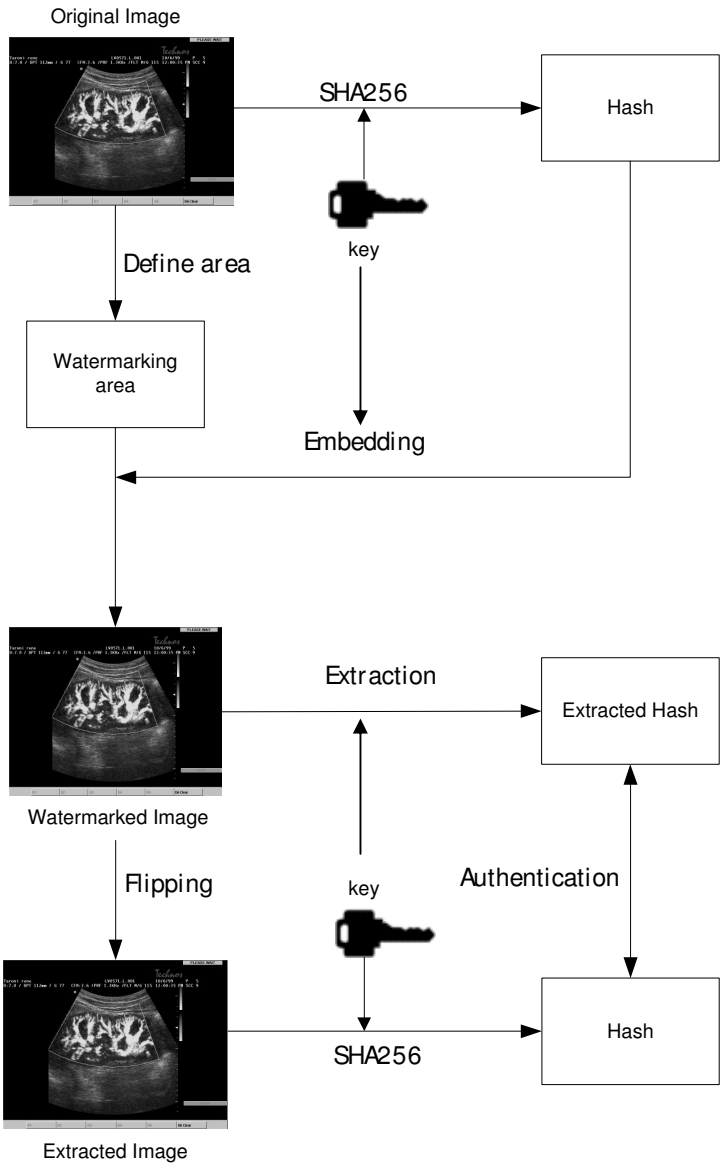
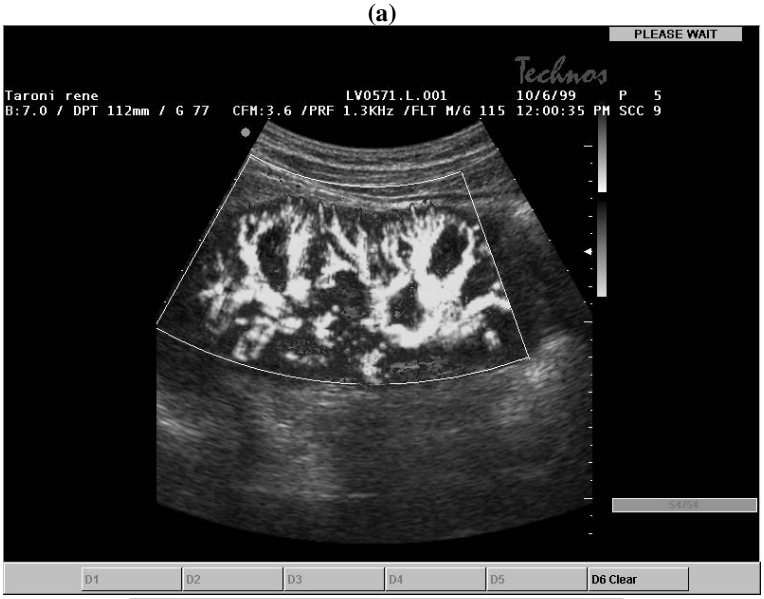


Figure 4.7 Strict Authentication Watermarking (SAW) System

68fb5d32d1fcabd3b22d53da68a30667
0c8ff9a348623072080f8feb28764bfa

(a)



838145455036aa5521d96d5d4657b8
b1d43be9107661ce492015bc999008

(b)

Figure 4.8 (a) Original image and its hash (b) Tampered image and its hash

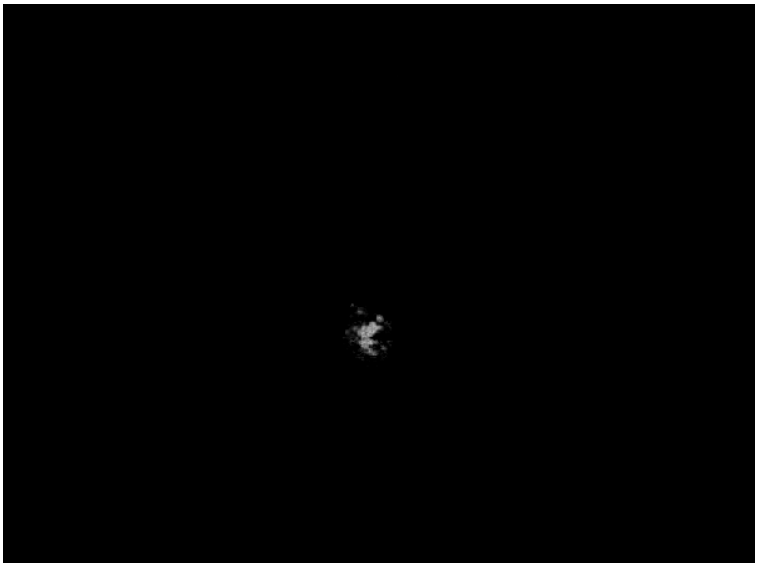


Figure 4.9. Image difference

Two blocks were then watermarked, using one LSB and two LSBs, increasing in the number of bits embedded to determine the capacity of LSB embedding before the recommended PSNR of 32dB was reached. Table 4.3 shows the result of embedding 270kb up to 550kb in the region of non-interest.

Figure 4.10. Watermarked image with 550kb payload

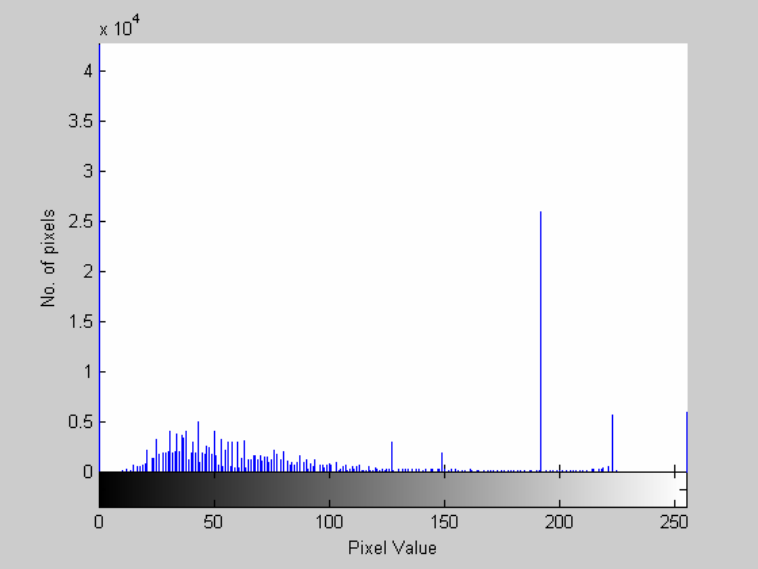


Figure 4.11(a) Histogram of Original Image

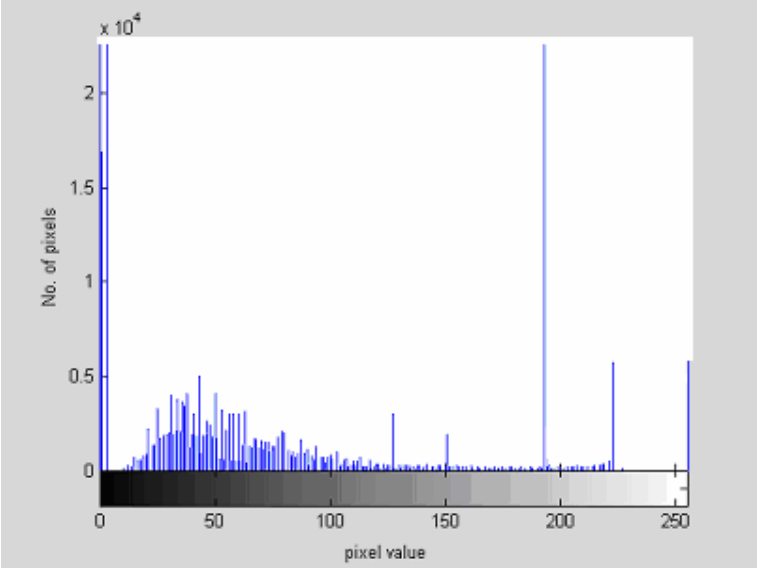


Figure 4.11(b) Histogram of watermarked image (550kb)

Figure 4.11 (a-b) shows the image histogram of the original image and a watermarked image with 550 kb payload. The histogram clearly shows the dramatic increase in the pixels with values 1 and 3, but keeping the remaining pixels exactly the same.

Capacity (kb)	PSNR (dB)
270	249.6
430	51.5
475	42.9
510	31.7
550	27.4

Table 4.3 Capacity and PSNR for 800x600 US Image

4.2.7 Conclusion

Watermarking in medical images holds great potential. From the large capacity available for embedding, a lot more information can be added to the image to make it more secure. Combining cryptography and compression will add security and more information to the limited capacity. The most important aspect regarding watermarking for medical image communications is that the image still conforms to the DICOM image format after watermarking takes place. In keeping distortion level very low, we could ensure that the watermarked image can still be valuable for other purposes, such as a case study for schools that does not disclose the patient's confidential information. A lossless watermarking scheme capable of verifying the authenticity and integrity of DICOM images is proposed. In addition, the original image can be recovered at the receiver site with the whole image's integrity being strictly verified. The watermarking scheme, including data embedding, extracting and verifying procedure were presented. Experimental results showed that such a scheme could embed a large payload while keeping distortion level very low.

4.3 Strict Authentication Watermarking with JPEG Compression (SAW-JPEG)

4.3.1 Image Compression

Image compression seeks to reduce the number of bits required to represent the image information. Two fundamental properties used in image compression are removal of redundancy and reduction of irrelevant content. Irrelevant content may include information not perceived by the viewer, namely the human visual system (HVS). Three types of redundancy may be exploited:

- Spatial redundancy or correlation between neighbouring pixels
- Spectral redundancy or correlation between different frequency bands
- Temporal redundancy or correlation between adjacent frames in a sequence of images (in video applications).

Compression algorithms can be divided into two main groups, lossless and lossy methods. In lossless compression schemes, only the redundancy is exploited, and the image is recorded in a more efficient manner. All the information is retained and so the reconstructed image is numerically identical to the original image. In lossy compression, information deemed irrelevant to the visual perception of the human viewer is discarded and so the compressed image cannot be perfectly reconstructed and distortion is introduced into the reconstructed image.

While lossless compression does not harm a watermarking system in any way (the original data can be perfectly reconstructed), lossy compression methods introduce distortion that has to be taken into account in watermarking applications. Lossy compression techniques are nowadays being commonly used as a means to effect a reduction on the requirement for bandwidth and storage space. It is therefore necessary to study the effects of lossy image compression on watermarking systems.

It should be observed that the design goal of lossy compression systems is opposed to that of watermark embedding systems. The HVS model of the compression system attempts to identify and discard perceptually insignificant information of the image, whereas the goal of the watermarking system is to embed the watermark information without altering the visual perception of the image. An optimal compression or de-noising system would immediately discard any such watermark information. Fortunately, all current compression methods are not optimal and allow watermarking schemes to be devised that will embed watermark information that is robust.

It remains unresolved how lossy compression should best be employed for the storage and transmission of medical images. There is little guidance from the scientific

literature, professional practice standards, regulatory authorities, or the common law. Although lossy compression schemes are included in medical standards such as DICOM, their clinical use is not defined; it is only that the technology is available for use at the discretion of the user or implementer.

There is no good metric by which to judge lossy compression schemes or determine appropriate threshold levels for diagnostic use. Quantitative metrics based on an analysis of the image pixels such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) do not correlate well with observers' opinions of image quality, or the measurement of observers' performance when undertaking diagnosis. Metrics based on models of human visual perception are still in their infancy. They have not been thoroughly compared to observer performance for medical applications (Clunie 2000).

Hybrid lossless/lossy compression schemes have been developed for medical applications. These identify regions of images that are determined by some criterion to be of little or no clinical interest. These regions are then either discarded or compressed with greater loss. The remaining regions, which contain the regions of clinical interest, are compressed using a lossless compression scheme. This approach can result in a high compression overall and retain the effective quality of a lossless compression scheme. The difficulty is to determine the areas of clinical interest. There has been work to find automate algorithms, but the only reliable method has been to determine regions defined by physical characteristics Some early CT compression schemes did not encode information outside the circular reconstructed area at all (perimeter coding) and were very effective. However, if such areas are filled with a constant pixel value then most general-purpose lossless image compression schemes perform equally well.

4.3.2 JPEG Compression

JPEG (Wallace 1991) is currently the most frequently used compression algorithm for medical imaging. For example it is included within the DICOM standard. Improved compression algorithms such as JPEG2000, will replace JPEG in time. For the purposes of this work, the watermarking method will focus specifically on JPEG, although the

method should be extensible to other compression schemes based on a block compression scheme.

In this section, we briefly review the JPEG lossy compression standard (Wallace 1991). At the input to the JPEG encoder, the source image, X, is grouped into ρ nonoverlapping 8x8 blocks, X_p. Each block is sent sequentially to the Discrete Cosine Transform (DCT). Instead of representing each 8x8 matrix, we can rewrite it as a 64x1 vector following the "zigzag" order (Wallace 1991). Therefore the DCT coefficients, F_p, of the vector, X_p, can be considered as a linear transformation of X_p with a 64x64 transformation matrix D, such that,

$$F_p = DX_p \tag{4.5}$$

The two-dimensional DCT of an M x N image X is defined as follows:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N \end{cases} \tag{4.6}$$

The values B_{pq} are called the DCT coefficients of X. The DCT is an invertible transform. Each of the 64 DCT coefficients is uniformly quantized with a 64-element quantization table, Q.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 4.12. JPEG quantization table

In JPEG, the same table is used on all blocks of an image. Quantization is defined as the division of each DCT coefficient by its corresponding quantizer step size, and rounding to the nearest integer:

$$\tilde{f}_p(v) \equiv IntegerRound(\frac{F_p(v)}{Q(v)}), \tag{4.7}$$

where v = 1…64. In equation (4.7), $\tilde{f}_p$ is the output of the quantizer. We define $\tilde{F}_p$, a quantized approximation of $F_{\mathrm{p},}$ as

$$\tilde{F}_p \equiv \tilde{f}_p(v) \cdot Q(v) \tag{4.8}$$

In addition to quantization, JPEG also includes scan order conversion DC differential encoding, and entropy coding.

Inverse DCT (IDCT) is used to convert $\tilde{F}_p$ to the spatial domain image block $\tilde{X}_p$

$$\tilde{X}_p = D^{-1}\tilde{F}_p \tag{4.9}$$

All blocks are then tiled to form a decoded image frame. Theoretically, the results of IDCT are real numbers. However the brightness of an image is usually represented by

an 8-bit integer from 0 to 255 and thus a rounding process mapping those real numbers to integers is necessary.

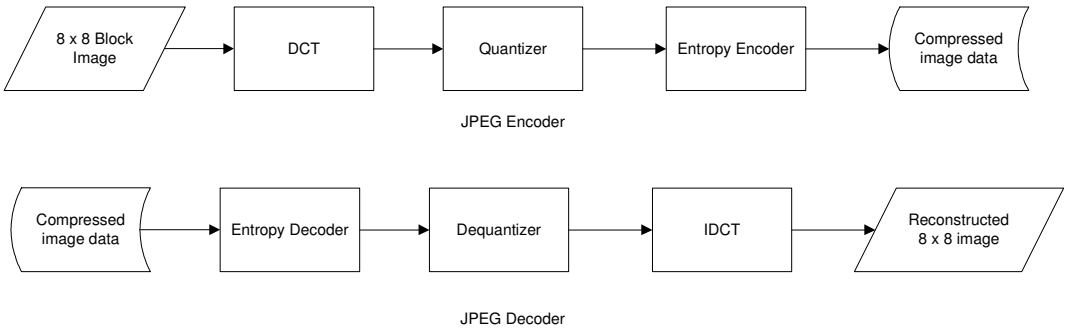


Figure 4.13. JPEG Encoder and decoder

To embed a watermark in the spatial domain, it is necessary to ensure that the embedded watermark will survive JPEG quantization process. JPEG processes images in 8 x 8 blocks, and so the method in which the watermark is embedded should be based on this same block structure. The process may be illustrated by encoding an 8 x 8 sub-image using JPEG. Consider if a '1' is embedded into the whole of the LSB plane of the 8 x 8 block as depicted by figure 4.14.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Figure 4.14. '1' bit embedded in 8x8 block

After the DCT transform of the block, figure 4.15 is the result.

8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4.15. DCT Transform of figure 4.14

To survive the quantization process, the value must be preserved through transformation and inverse transformation, that is

$$F_p = \tilde{F}_p \tag{4.10}$$

To achieve this, $\frac{F_p(v)}{Q(v)}$ must be the integer and have no effect on the rounding process. In particular the DC quantization coefficient should be equal to the dc component in order to preserve an integer result, and all other quantization coefficients should be scaled accordingly. For higher compression rate, to preserve an integer value, the embedded level must be increased, which will naturally have an effect on the quality of the image.

By designing the watermark embedding algorithm around the properties of the compression scheme, it is possible to preserve the watermark values. In this case, a priori knowledge of the quantization algorithm allows the DC coefficient to be unchanged through the compression/decompression process.

4.3.3 Method

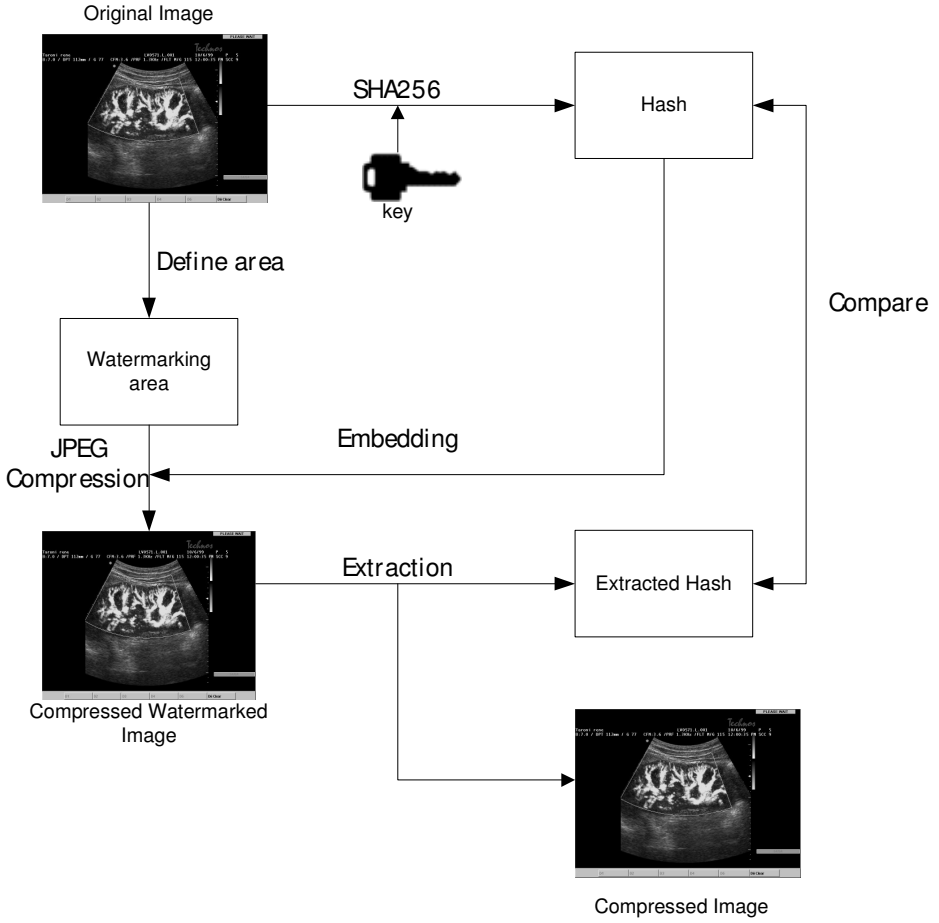


Figure 4.16. Watermarking scheme

The complete process is shown in figure 4.16 and comprises of the following steps:

1) **Define area**: This will define the Region of interest (ROI) where the smallest rectangle is obtained. Please refer to section 4.2.5.

2) **SHA256**: refer to section 4.2.5.

3) **Embedding**: Embed the hash value in the Region of Non-Interest (RONI) in the LSB. Since JPEG uses 8x8 blocks, we try to embed 1-bit in an 8x8 block. We only need 256 8x8 blocks to be able to embed the hash value.

4) **JPEG Compression**: Compression is performed on the watermarked image.

5) **Extraction**: The watermark is recovered from the watermarking area.

6) **Authentication**: The original hash and the extracted hash value are compared.

4.3.4 Experimental Results

An ultrasound image of 800x600x8 with a watermark embedded in RONI was subject to increasing levels of JPEG compression. The compression was performed using a quality factor in Matlab 6.5.1 to produce files with .jpg extension.

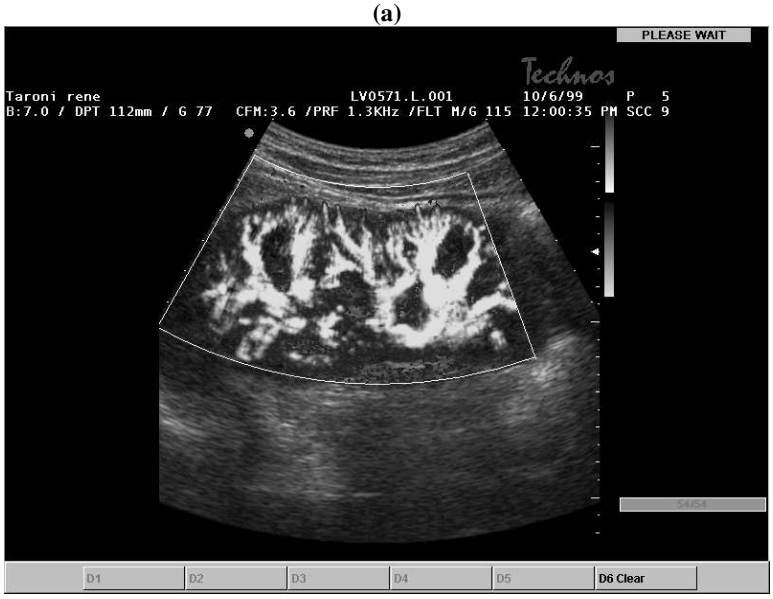
The following are the results. Table 4.4 shows that the watermark is robust to a high compression rate up to 90.6%. The JPEG image quality threshold is 60 for the least significant bit embedding. The image quality threshold is increased to 61 for 2nd and 3rd LSB manipulations.

Manipulation	Image Quality Threshold	Compression (%)	PSNR (dB)
1st LSB	60	90.6	40.75
2nd LSB	61	90.4	40.84
3rd LSB	61	90.4	40.84

Table 4.4. LSB Embedding and Image Quality Threshold

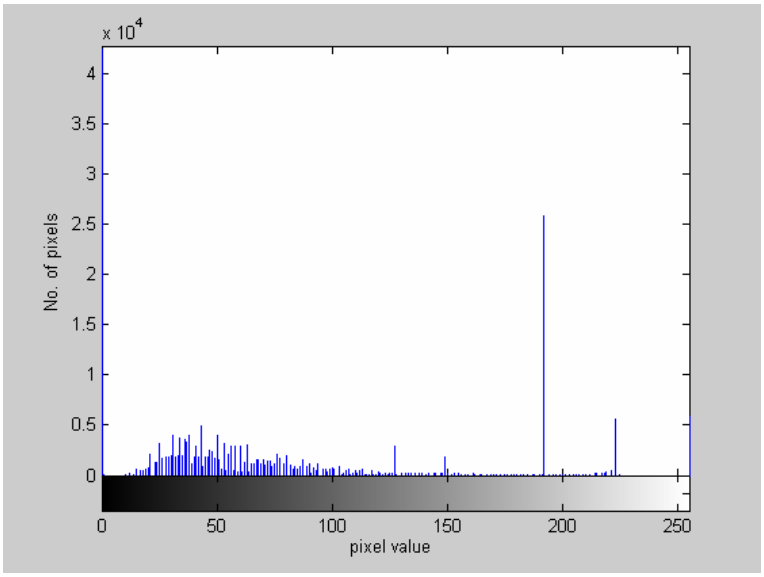
Figure 4.17 shows the original 800x600 US image and the compressed watermarked image with quality 60. This has the effect of changing some pixel values, with a marked effect on areas of abrupt change resulting in the increase of pixel values 2 – 10 (figure 4.18). The effect of adding the watermark is evident by the peaks of pixel value 0 and 1. JPEG loses definition, particularly at high frequencies, which has the effect of low pass or smoothing filter.

(a)

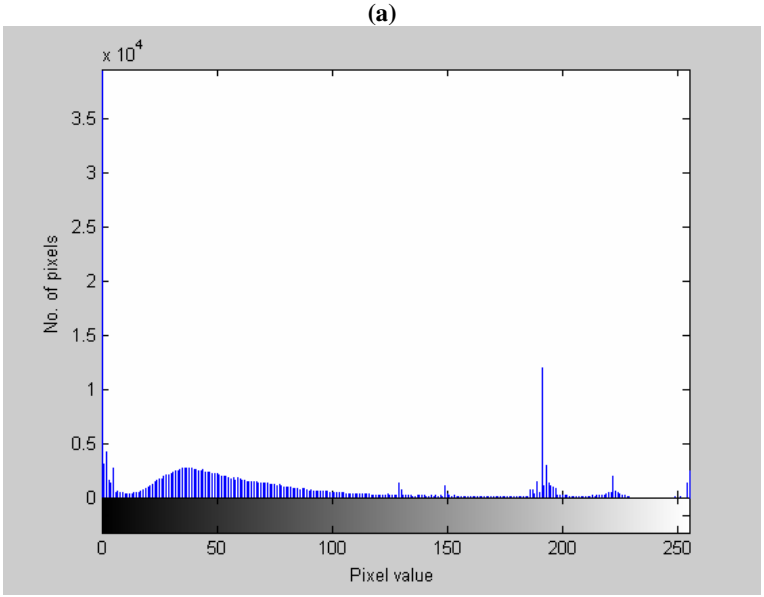


(b)

Fig. 4.17. a) Original 800x600 US image b) compressed watermarked image with quality 60



(a)



(b)

Figure 4.18. a) Image histogram of figure 4.17(a);
b) Image histogram of compressed image of figure 4.17(b)

An 800x600 ultrasound image was watermarked with its hash and then compressed with quality 60 in Matlab 6.5.1. The hash value of the original was recorded as "fcc29cbb8ea81be407cdd93e0326bf2bb68dca3d7872c9b6a033a981e184f989", and the hash value of the compressed image was extracted. Figure 4.19 shows (a) the original 800x600 ultrasound image, (b) the watermarked original image and (c) the watermarked image after compression with quality 60. The extracted hash value was

"fcc29cbb8ea81be407cdd93e0326bf2bb68dca3d7872c9b6a033a981e184f989", and is exactly the same as the original hash. This shows that the watermark can survive JPEG compression with Matlab 6.5.1 quality 60.

(a)

(b)

(c)

Figure 4.19. (a) original image (b) Watermarked image (c) the image after compression

4.3.5 Conclusion

A lossless watermarking scheme is proposed that is robust to lossy JPEG compression and at the same time is able to verify the authenticity and integrity of medical images. The watermarking scheme, including data embedding, extracting and verifying procedure were presented. Experimental results showed that such a scheme could embed and extract the watermark at a high compression rate. Combining cryptography and compression will add security to the medical images. In keeping the distortion level low, we could make sure that the watermarked image can still be valuable for other purposes, such as case studies in schools, but without disclosing a patient's confidential information.

Chapter 5

Authentication Watermarking with Tamper Detection and Recovery (AW-TDR)

5.1 Introduction

In this chapter, we present an efficient and effective watermarking method for image tamper detection and recovery. This chapter is structured as follows:

- Section 5.2 reviews authentication watermarking by Wong (1998) as a basis for discussion on vector quantization counterfeiting attacks.
- Section 5.3 describes vector quantization (VQ) counterfeiting attacks on block-wise independent watermarking schemes.
- Section 5.4 discusses a few techniques as countermeasures against VQ counterfeiting attacks. These include increasing block dimension, breaking block-wise independent and using a hierarchical watermarking technique.
- Section 5.5 proposes an authentication watermarking technique with tamper detection and recovery (AW-TDR).

5.2 Block-based Authentication Watermark

A block-based watermarking technique (Wong 1998) used an *M x N* image *X* and a binary watermark image *W*. In practice, this step is usually achieved by tiling the original image with a smaller logo image.

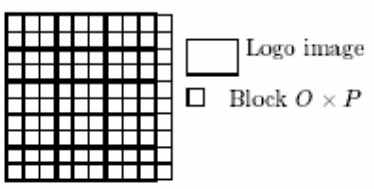


Figure 5. 1 Tiling of logo image in Wong's scheme

The original image X is partitioned into O x P pixel blocks, $\{X_1, X_2, \ldots\}$; where X_r denotes such blocks. Likewise, the watermark image is partitioned into blocks, W_r. For each block X_r, a corresponding block $\tilde{X}_r$ is formed by setting the least significant bit of each pixel to zero. A cryptographic hash (e.g.,. MD5 or SHA) of transformed block $\tilde{X}_r$ and image dimensions is computed.

$$H_r = \mathrm{H}(M, N, \tilde{X}_r) \tag{5.1}$$

The signature of a block is formed by XORing the computed hash with the watermark pattern and encrypting the result with a public key encryption algorithm.

$$S_r = Encrypt(H_r \oplus W_r, Key_{private}) \tag{5.2}$$

where $\oplus$ denotes the bitwise XOR operator. Finally, the signature S_r is inserted in X_r as the least significant bits of the block. Note that the application of this procedure independently on each block produces the watermarked image.

During watermark verification similar steps are followed. First the candidate image $\tilde{X}$ is partitioned into blocks $\tilde{X}_r$. Signature $\tilde{S}_r$ is read from the least significant bits of each

block, $\tilde{X}_r$. $\tilde{X}_r$s are formed by setting LSBs to zero and $\tilde{H}_r$s are calculated using image sizes and $\tilde{X}_r$s. Finally, watermark image blocks are recovered by XORing the hash values with decrypted signatures from each block.

$$\tilde{W}_r = Decrypt(\tilde{S}_r, Keypublic) \oplus \tilde{H}_r \tag{5.3}$$

Any changes in the pixel values of a block alter either the decrypted signature or (with very high probability) the output of the hash function. Theoretically, a randomly generated block may carry the correct watermark pattern. Nevertheless, the probability of such an occurrence is practically negligible. In either case, the recovered watermark block $\tilde{W}_r$ will be significantly different than the embedded watermark block. As a result, when a group of pixels in a spatial region are altered, the manipulation will be detected by the change in the corresponding region of the binary watermark image. On the other hand, it is possible to replace an entire image block with another without arousing suspicion, provided that both blocks bear the same watermark pattern. This observation is the basis for the vector quantization attack described in the next section.

5.3 Vector Quantization Counterfeiting Attack

A counterfeiting attack on block-wise independent watermarking schemes was proposed by Holliman and Memon (2000). The attacker approximates an image for which he wishes to create a forgery by using a collage of authentic blocks from watermarked images. Since the embedding and authentication processes are block-wise, the verification algorithm authenticates the collage image. Given a large enough database of watermarked images, the attacker can ensure that the counterfeit collage image has the same visual appearance as his original unwatermarked image.

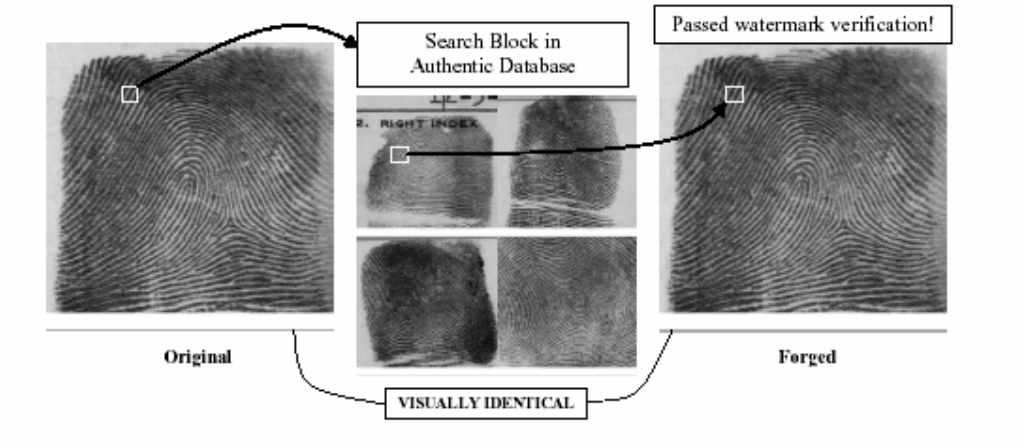


Figure 5.2 Vector quantization attack. The attacker approximates an image (on the left) by a collage of authentic blocks from watermarked images (center). The resulting image (right) is visually identical to the original and is deemed valid by the watermark detector.

5.4 Countermeasures Against Counterfeiting Attack

In this section a number of modifications on Wong's scheme that have been proposed as countermeasures against the vector quantization attack are discussed.

- Increasing Block Dimensions

  Vector quantization process depends on two key factors: the size and the number of image blocks in a codebook. Smaller size blocks can be approximated more accurately given a fixed size codebook. Similarly, better approximations can be obtained as the number of blocks in the codebook increases. Therefore, increasing the block dimensions used in the watermarking process can reduce the possibility of a reasonable forgery. Larger blocks also decrease the number of authentic blocks that can be obtained from one image and this will degrade the quality of forgery by reducing the codebook size.

  This countermeasure, however, does not thwart the attack completely. If the set of watermarked images available to the attacker is quite large, reasonable forgeries can still be produced. Moreover, using larger and larger blocks also impairs the tamper localisation accuracy of the watermark.

- Breaking Block-Wise Independence: Neighbourhood Dependent Blocks

An alternative method of eliminating the VQ attack is to eliminate the block wise independence of the watermark. In particular, the signature embedded in block Xr may be calculated using a larger support Xr, which overlaps the neighbouring blocks. This technique is very similar to block chaining modes used in block encryption techniques (e.g., CBC mode in DES – see Menezes, Oorchoot et al. 1997). Using this scheme, a collage of individually watermarked blocks of an image is no longer authenticated by the watermarking extraction process because the larger support covering the neighbouring blocks is not preserved.

- Hierarchical Block-Based Watermarking

A technique that embeds and extracts a watermark in a multilevel hierarchy was proposed (Celik et al. 2002). On the lowest level, the image, X is partitioned into O x P non-overlapping blocks. At each successive level, the image is partitioned into blocks that in turn are composed of 2 x 2 blocks at the preceding level of the hierarchy.

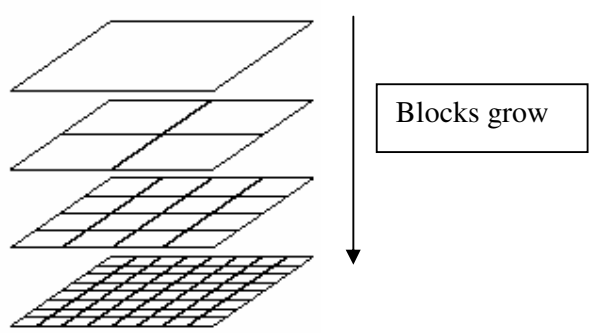


Figure 5.3 Partitioning of an image and the resulting four level hierarchical block structure

Although the method claimed it could eliminate the vulnerabilities of Wong's (1998) scheme to VQ attack, it is found that the method also compromises on the accuracy of localisation. For example, using an ultrasound image of size 800 x 600 pixels, the image will be partitioned, resulting in four level hierarchical block structure with the smallest block of 100 x 75 pixels.

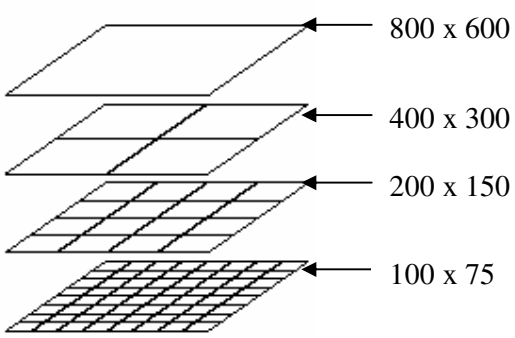


Figure 5.4 Partitioning of image size 800 x 600 pixels

5.5 Authentication Watermarking with Tamper Detection and Recovery (AW-TDR)

In this section, an efficient and effective digital watermarking method for image tamper detection and recovery is presented. The method is based on four concepts introduced from the literature: 1) block-based (Fridrich and Goljan 1999); 2) separating authentication bits and recovery bits (Lin and Chang 2001); 3) hierarchical (Celik et al 2002); and 4) average intensity as an image feature (Lou and Liu 2000). The method is efficient as it only uses simple operations such as parity check and comparison between average intensities. It is effective because the scheme inspects the image hierarchically with the inspection view increasing along with the hierarchy so that the accuracy of tamper localisation can be ensured. This scheme can perform both tamper detection and recovery for tampered images. Tamper detection is achieved through a block-based, inspection and recovery of a tampered block and relies on its feature information hidden in another block, which can be determined by a one-dimensional transformation.

5.5.1 Torus Automorphism

Torus automorphism is a kind of dynamic system. A dynamic system is a system whose states change with time t. When t is discreet, a dynamic system can be presented as iteration of a function f, $S_{t+1} = f(S_t)$, where $t \in Z = \{0, 1, 2, 3, \ldots\}$, S_t, S_{t+1} are the states at time t and $t+1$, respectively (Voyatzis and Pitas 1996a). A two-dimensional Torus automorphism can be considered as a permutation function or a spatial transformation

of a plane region. This transformation can be performed using a 2 x 2 matrix A with constant elements. More specifically, a state S_{i+1} or a point (x_{i+1}, y_{i+1}) can be transformed from another state S_i or another point (x_i,y_i) by

$$A = \begin{pmatrix} a1 & a2 \\ a3 & a4 \end{pmatrix}, \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = A \times \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod N, \tag{5.4}$$

where $a_i \in Z, |A| = 1$, and A has eigenvalues $\lambda_{1,2} \in R - \{-1,0,1\}$, R is the set of rational numbers. The detailed characteristics of A are described in (Voyatzis and Pitas 1996b, Voyatzis and Pitas 1996a). A set of successive points $\{S_0, S_1, S_2, \ldots\}$, generated by equation (5.4) composes an orbit φ of the system and the initial point $S_0 = (x_0,y_0)$ classifies φ into two categories. When x_0 and/or y_0 are irrational, φ is infinite. When both x_0 and y_0 are rational, φ is chaotic and periodic at every R times, $S_R = S_0$. R is called the recurrence time. Voyatzis and Pitas (1996) presented a one-parameter, two dimensional, discrete Torus automorphism as in equation (5.5), for creating a unique and random mapping of the pixels within an image:

$$A = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}, \quad \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = A \times \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod N, \tag{5.5}$$

where $(x_i, y_i) \in [0, N-1] \times [0, N-1]$ and $k \in [0, N-1]$. The recurrence time R depends upon the parameters k, N, and the initial point (x_0,y_0). In most cases, R is equal to N-1 or N+1, when N is prime (Voyatzis and Pitas 1996b, Voyatzis and Pitas 1996a).

5.5.2 Watermark Embedding

The watermarking embedding procedure is described in this section. Each image is of size M x N pixels where M and N are assumed to be a multiple of six and the number of grey levels is 256.

- Preparation

  We need to prepare a one to one block mapping sequence A → B→ C→ D → … → A for watermarking embedding, where each symbol denotes an individual block. The intensity feature of block A will be embedded in block B, and the intensity feature of block B will be embedded in block C, etc. Since the number of blocks in each dimension of most images can be hardly be a prime number, we cannot obtain a one to one mapping among the blocks by applying equation (5.5), based on the analysis in (Voyatzis and Pitas 1996b). Instead, a 1D transformation was used:

$$\vec{B} = [(k \times B) \bmod N_b] + 1, \tag{5.6}$$

  where $B, \vec{B}, k \in [1, N_b]$, k is a secret key (prime number), and N_b is the total number of blocks in the image.

  The generation algorithm of the block-mapping sequence is as follows:

  1. Divide the image into non-overlapping blocks of 6x6 pixels
  2. Assign a unique and consecutive integer $B \in \{1,2,3,...,N_b\}$ to each block from left to right and top to bottom, where N_b= (M/6) x (N/6)
  3. Randomly pick a prime number $k \in [1, N_b]$
  4. For each block number B, apply equation (5.6) to obtain $\vec{B}$, the number of its mapping block
  5. Record all pairs of B and $\vec{B}$ to form the block mapping sequence

k	B	1	2	3	4	5	6	7	8	21	22	23	24
23	$\vec{B}$	24	7	30	13	36	19	2	25	4	27	10	33
26	$\vec{B}$	27	13	39	25	11	37	23	9	27	13	39	25

Table 5.1 Mapping of blocks with k=23,26 and Nb=40

Note that the secret key, k, must be a prime in order to obtain a one to one mapping; otherwise, the period is less than N_b and a one to many mapping may occur. Table 5.1 lists some parts of the mapping sequence generated with Nb=40, k=23 and 26 respectively. In this table, $\vec{B}$ starts to repeat at B=21 when k=26, which is not a prime.

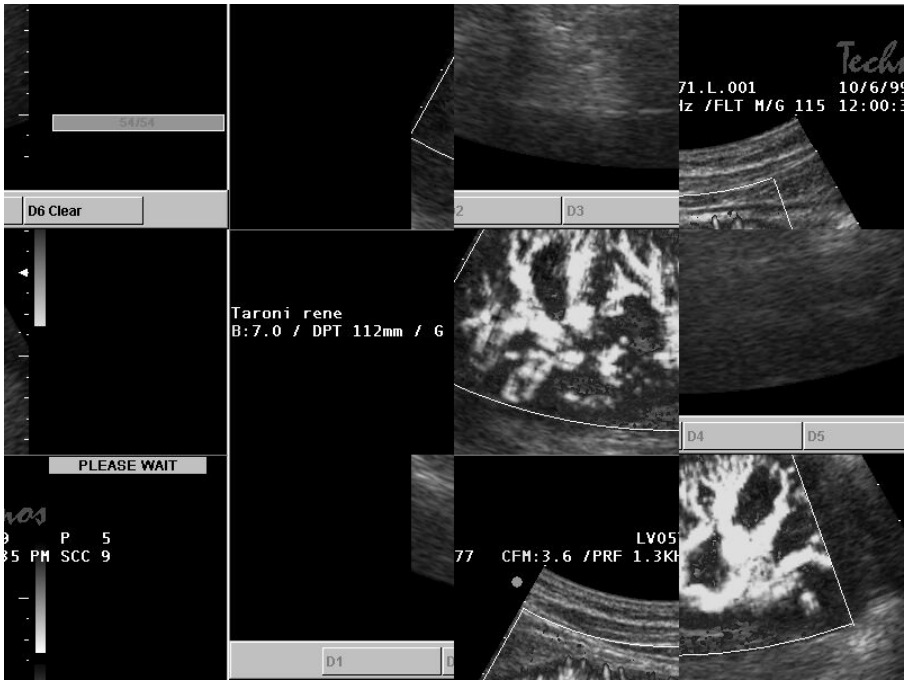


Figure 5.5 Image mapping using toral automorphism. Blocksize=200 k= 5

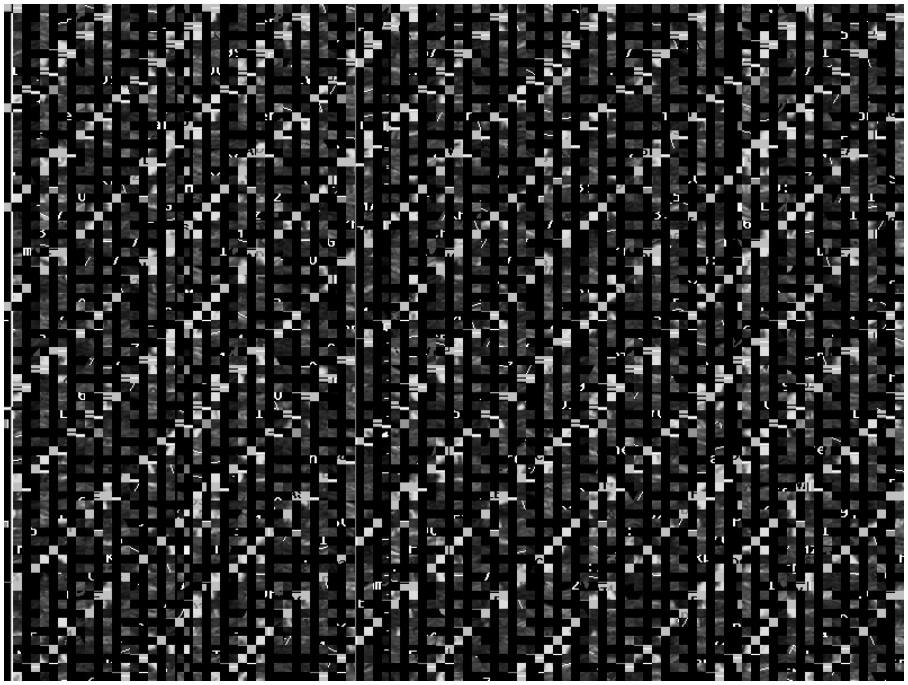


Figure 5.6 image mapping using toral automorphism. Blocksize=8 k= 3739

Figure 5.5 and 5.6 shows an 800x600 ultrasound image divided into equal size blocks and mapped using toral automorphism.

- Authentication watermark and recovery watermark generation

In the schemes proposed by Wong (1998) and Celik et al. (2002) a signature was generated for each block in order to localise tamper. Signature generation is computationally expensive and requires more bits for embedding, thus it will have an effect on the quality of the watermarked image.

In this section a case of using intensity average comparisons and parity bits as the authentication watermark is presented. To localise tamper in a block, the watermark needs to be embedded directly into that block. If a block is being tampered locally, the intensities of the pixels involved will be changed. This will also change the average intensity of the block concerned. To ensure that this is not changed, a parity check will be used. However, a parity check alone will not guarantee that the block has not been changed, because local tampering usually causes burst error (Cox et al. 2002), meaning that if more than one bit has been changed, a parity check is no

longer useful. Using ECC will help solve this issue, but again more watermark bits will be needed. To overcome this, the intensity comparison is used as another guard if a parity check fails. This feature will also be used to break block wise independent. To break block wise independent, the intensity of the block is compared to the intensity of a larger block. Let B denote the bigger block (figure 5.7) and the smaller or sub block as B_s, then the average intensity of B is

$$Avg\_B = \frac{(P_1 + P_2 + P_3 + \ldots + P_{15} + P_{16})}{16} \tag{5.7}$$

and the average intensity of sub block is

$$Avg\_B_s = \frac{(P_1 + P_2 + P_5 + P_6)}{4} \tag{5.8}$$

P_1	P_2	P_3	P_4
P_5	P_6	P_7	P_8
P_9	P_{10}	P_{11}	P_{12}
P_{13}	P_{14}	P_{15}	P_{16}

Figure 5.7 A 4x4 Block B

The intensity of each sub block will be used as the recovery watermark, and will be embedded in a block mapped by equation 5.6. This is to ensure that if the block is tampered with, the recovery bits will be highly likely to be available. In order to consider the block size suitable for recovery, the average intensities of 4x4, 3x3 and 2x2 blocks of the whole image were created to see the visual effect for each block size and the results of the signature image are presented in figure 5.6, 5.7 and 5.8 respectively. The 4x4 signature image loses fine details with clearly visible block effect. The 3x3 signature image has better quality, with fine details and less visible block effect. The 2x2 signature is the best without losing any fine details and no visible block effect.

Figure 5.8 Signature image using 4x4 block.

Figure 5.9. Signature image using 3x3 block

Figure 5.10. Signature image using 2x2 block. Quality is good. No visual block effect. A good candidate for recovery.

The choice of which signature image to use will depend on:

1. How many LSBs will be used, which is the answer to how much degradation is allowed for the watermark.
2. How will the recovered image be used? Will it be considered as authentic? If it is not, will it be used as an indication of the location and the nature of the tampering?

LSB is suggested, to minimise the degradation as medical images are very strict with the quality. The recovered image, however, will not be considered authentic and will not be used for any clinical purposes. One possibility for the purpose of recovery is to help in the investigation to find the motive and the person responsible for the tampering. A 3x3 sub block in a 6x6 block is suggested to accommodate two authentication bits and seven recovery bits to be embedded in the LSB of each pixel.

- Embedding

For each block B of 6x6 pixels, divide it into four sub-blocks of 3x3 pixels. The watermark in each sub-block is a 3-tuple (v, p, r), where both v and p are 1-bit authentication watermark, and r is a 7-bit recovery watermark for the corresponding sub-block within block A mapped to B. The following algorithm describes how the 3-tuple watermark of each sub-block is generated and embedded:

1. Set the LSB of each pixel within the block to zero and compute the average intensity of the block and each of its four sub-blocks, denoted by avg_B and avg_B_s, respectively.

2. Generate the authentication watermark, v, of each sub-block as:

$$v = \begin{cases} 1 & if\ avg\_Bs \geq avg\_B, \\ 0 & otherwise, \end{cases} \tag{5.9}$$

3. Generate the parity check bit, p, of each sub-block as :

$$p = \begin{cases} 1 & if\ num\ is\ odd, \\ 0 & otherwise, \end{cases} \tag{5.10}$$

where num is the total number of 1s in the seven MSBs of avg_B_s.

4. From the mapping sequence generated in the preparation step, obtain block A whose recovery information will be stored in block B.
5. Compute the average intensity of each corresponding sub-block As within A, and denote it avg_A_s.
6. Obtain the recovery intensity, r, of A_s by taking the seven MSBs in avg_A_s.
7. Embed the 3-tuple watermark (v, p, r), 9 bits in all, onto the LSB of of each pixel in Bs.

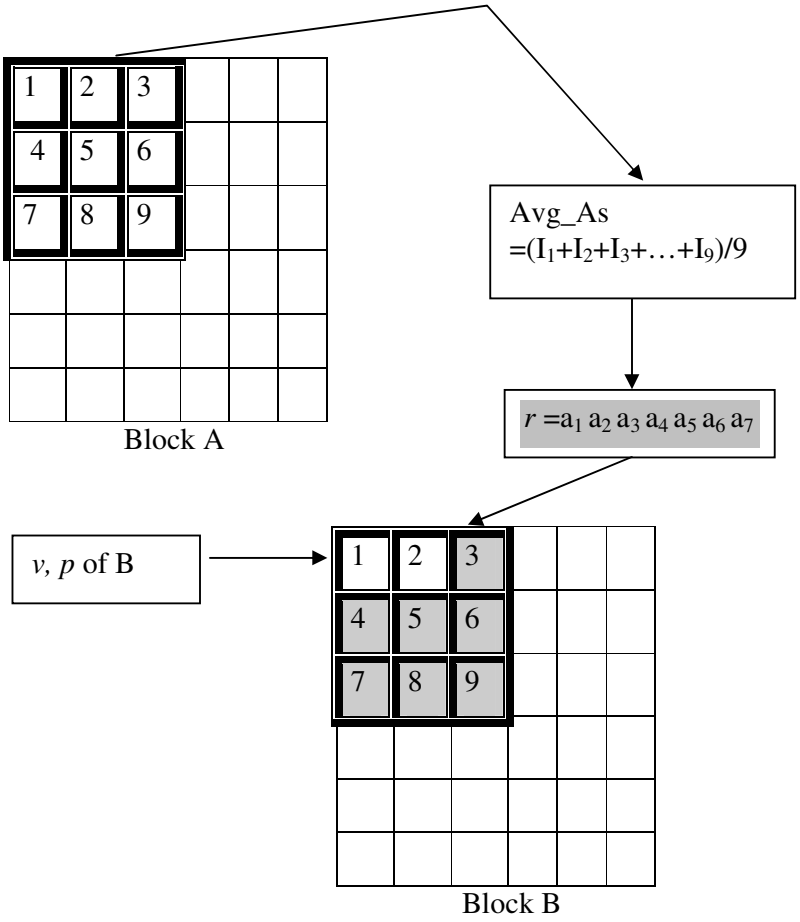


Figure 5.11(a) Watermark generation and embedding location

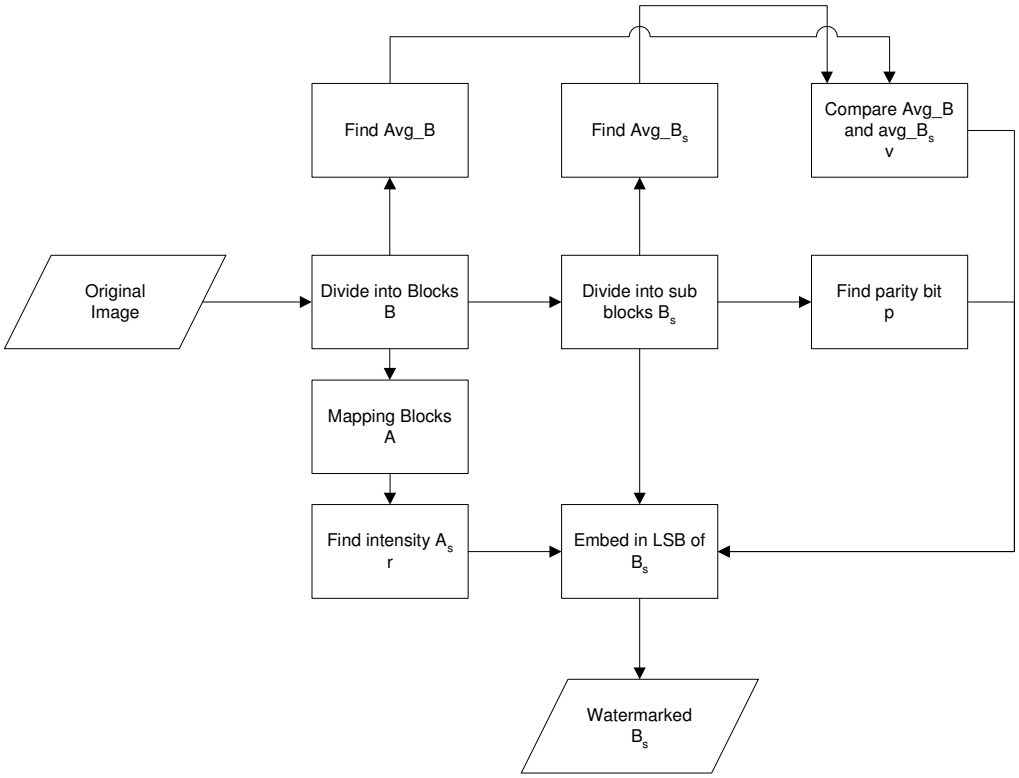


Figure 5.11(b) AW-TDR embedding scheme

5.5.3 Tamper detection

The test image is first divided into non-overlapping blocks of 6x6 pixels, as in the watermarking embedding process. For each block denoted as $\vec{B}$, the LSBs of each pixel in $\vec{B}$ were set to zero and compute its average intensity, denoted as avg_$\vec{B}$. A 2-level detection is then performed. In level-1 detection, each 3x3 sub-block within one block is examined. In level-2 detection, a 6x6 block is treated as one unit. Level-3 detection is for VQ attack resilience only. The procedure of our hierarchical tamper detection scheme is described in the following:

- Level-1 detection.

For each sub-block $\vec{B}$s of 3x3 pixels within the block $\vec{B}$, perform the following steps:

1. Extract v and p from $\vec{B}_s$.

2. Set the LSBs of each pixel within each $\vec{B}_s$ to zero and compute the average intensity for each sub-block $\vec{B}_s$, denoted as avg_$\vec{B}_s$.
3. Count the total number of 1s in avg_$\vec{B}_s$ and denote it as P_s.
4. Set the parity check bit p' of $\vec{B}_s$ to 1 if P_s is odd, otherwise, set it to 0.
5. Compare p' with p. If they are not equal, mark $\vec{B}_s$ as tampered and complete the detection for $\vec{B}_s$.
6. Set the algebraic relation v'=1 if avg_$\vec{B}_s$>=avg_$\vec{B}$, otherwise, set it to 0.
7. Compare v' with v. If they are not equal, mark $\vec{B}_s$ as tampered and complete the detection for $\vec{B}_s$; otherwise mark it valid.

- Level-2 detection.

For each block of size 6x6 pixels, mark this block tampered if any of its sub-blocks is marked tampered; otherwise mark it valid.

- Level-3 detection.

For each valid block $\vec{B}$ of size 6x6 pixels, perform the following steps:

1. Find the block number of block C, where block C is the one in which the intensity feature of block $\vec{B}$ is embedded.
2. Locate block C.
3. If block C is marked tampered, assume block $\vec{B}$ is valid and complete the test.
4. If block C is valid, perform the following steps:
    a. Obtain the 7-bit should-be intensity of each $\vec{B}_s$ by extracting the LSBs from each pixels in the corresponding block within block C, padding one zero to the end to make an 8-bit value.
    b. Compare with avg_$\vec{B}$s and mark $\vec{B}$ tampered if they are different.

5.5.4 Image Recovery

After the detection stage, all the blocks are marked either valid or tampered. Only the tampered blocks are recovered and the valid blocks are left as they are. For

convenience, we call the tampered block, block B and the block embedded with its intensity, block C. The restoration procedure for each tampered block is described as follows:

1. Calculate the block number for block C.
2. Locate block C
3. Obtain the 7-bit intensity of each sub-block within block B, padding one zero to the end to make an 8-bit value.
4. Replace the new intensity of each pixel within the sub-block with this new 8-bit intensity.
5. Repeat step 3 and 4 for all sub-blocks within block B.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48

(a)

48	23	46	21	44	19	42	17
40	15	38	13	36	11	34	9
32	7	30	5	28	3	26	1
24	47	22	45	20	43	18	41
16	39	14	37	12	35	10	33
8	31	6	29	4	27	2	25

(b)

Figure 5.12. (a) An 8x6 block with block 18,19,26 and 27 tampered, (b) Recovery bits location

Figure 5.12(a) shows an 8x6 block and each block is given a number from 1 to 48. By using the transformation given by equation 5.3, with k=23, the transformation is given in figure 5.12(b). If, for example, blocks 18, 19, 26 and 27 were tampered with, all

blocks will be recovered since from the mapping block, as the recovery bits are stored in blocks 6, 23, 31 and 46 respectively.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48

(a)

48	23	46	21	44	19	42	17
40	15	38	13	36	11	34	9
32	7	30	5	28	3	26	1
24	47	22	45	20	43	18	41
16	39	14	37	12	35	10	33
8	31	6	29	4	27	2	25

(b)

Figure 5.13 (a) An 8x6 block with blocks 1,24 and 48 are tampered,
(b) Recovery bits stored in block 1,24 and 25

If three blocks 1, 24 and 48 were tampered with, the only block that will be recovered is block 24. The reason being that information for block 1 is embedded in block 24, which is tampered with resulting in a loss of information. The same applies to block 48, where the recovery bits were embedded in block 1 which has been tampered with. The recovery bits for block 24 however were embedded in block 25 that has not been tampered with.

5.5.5 Experimental Results

- Missing detection

In evaluating the proposed authentication watermarking with tamper detection and recovery (AW-TDR), different manipulations on an ultrasound image, two fingerprint

images and a military image were tested to obtain the miss detection rate for level-1 and level-2 detection.

The watermarked ultrasound image was manipulated using cloning tool from Adobe Photoshop CS. The manipulated area is ~30 x 50 pixels.

Figure 5.14 Original image

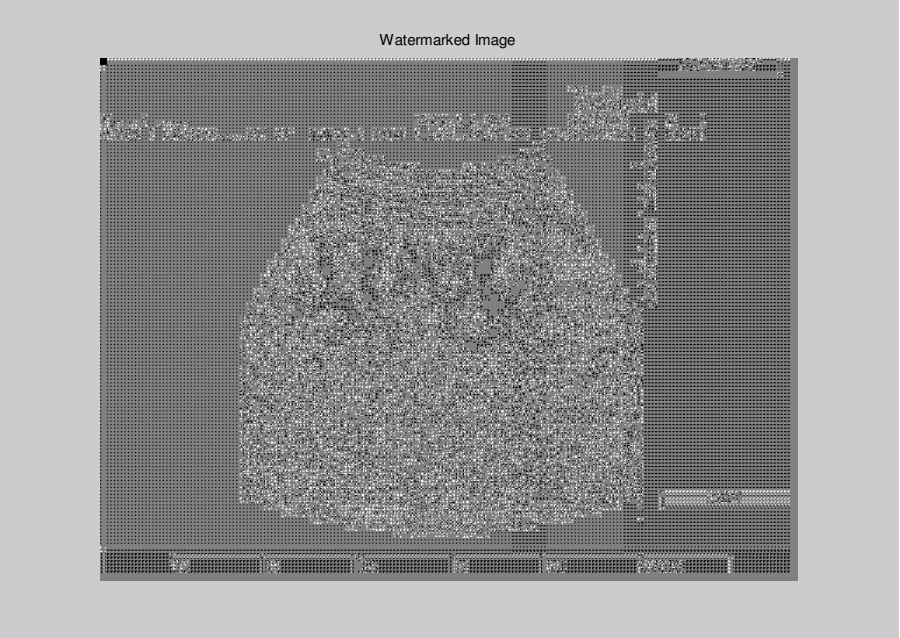


Figure 5.15 Watermark embedded PSNR = 54.1483 dB

Figure 5.16 Tampered image

Figure 5.17 Level 1 detection with some areas undetected

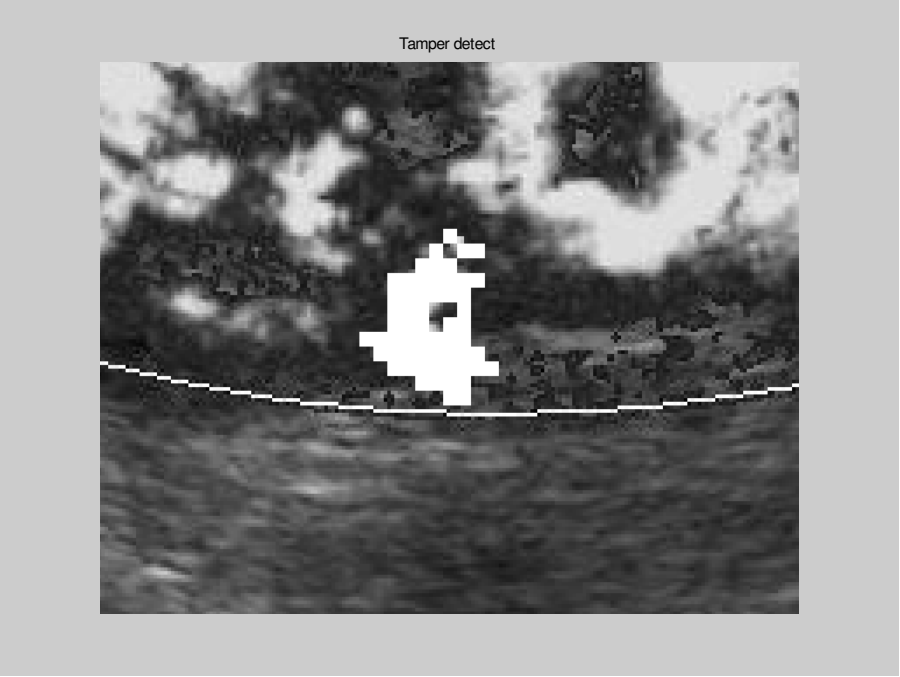


Figure 5.18 Some areas undetected magnified

Figure 5.19 level 2 detection

Figure 5.20. Magnified Level 2 detection

Figure 5.21. Original fingerprint1 (from National Institute of Science and Technology [NIST] Science and Technical database http://www.nist.gov/srd/nistsd4.htm)

Figure 5.22. Watermarked fingerprint1 PSNR = 54.5262 dB

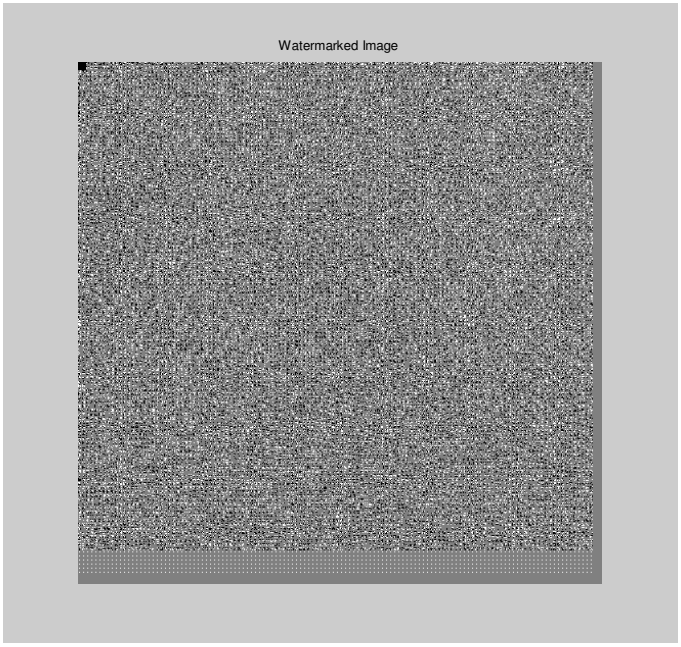


Figure 5.23. Watermark embedded in fingerprint1

Figure 5.24. Tampered watermarked fingerprint1

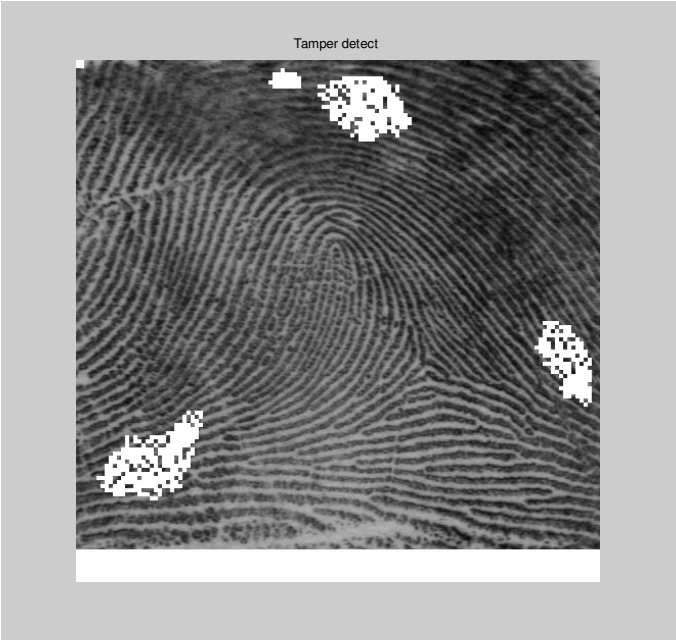


Figure 5.25. Level 1 detection- fingerprint1

Fingerprint1 was manipulated using healing brush tool and cloning tool. The manipulated sizes are ~60 x 50 and ~100 x 100 pixels.

Figure 5.26. Level 2 detection- fingerprint1

Figure 5.27. Watermarked fingerprint2 PSNR = 54.9982 dB

Figure 5.28 Tampered watermarked fingerprint2

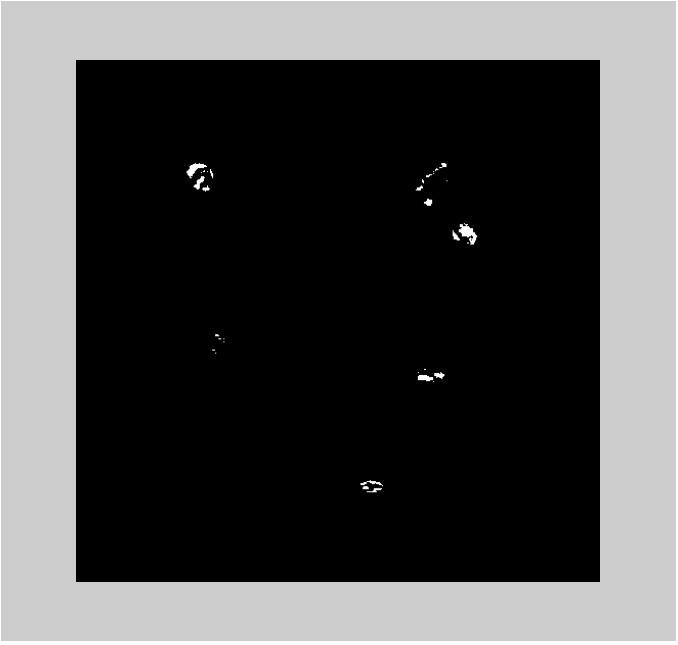


Figure 5.29. Image Difference

Fingerprint2 was manipulated using cut and paste and cloning tool. This time the manipulation size is smaller ranging from ~ 10 x10 to 40 x 100 pixels.

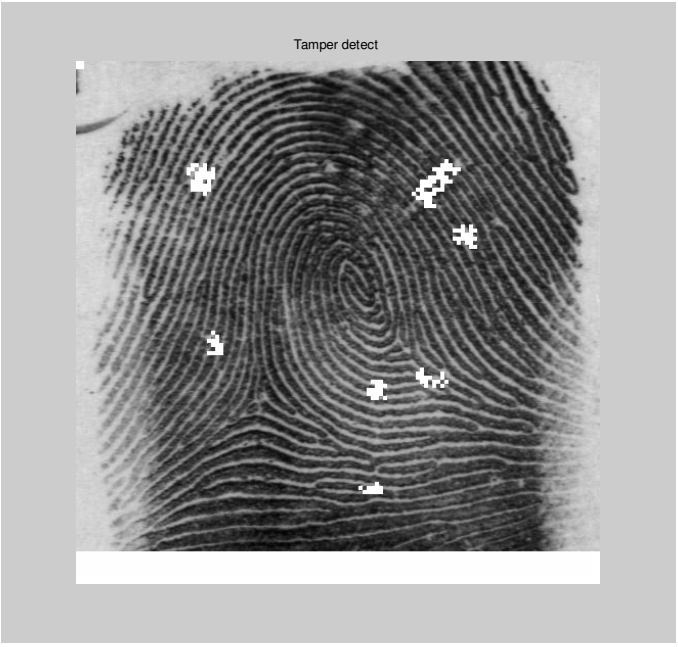


Figure 5.30. Level 1 detection – fingerprint2

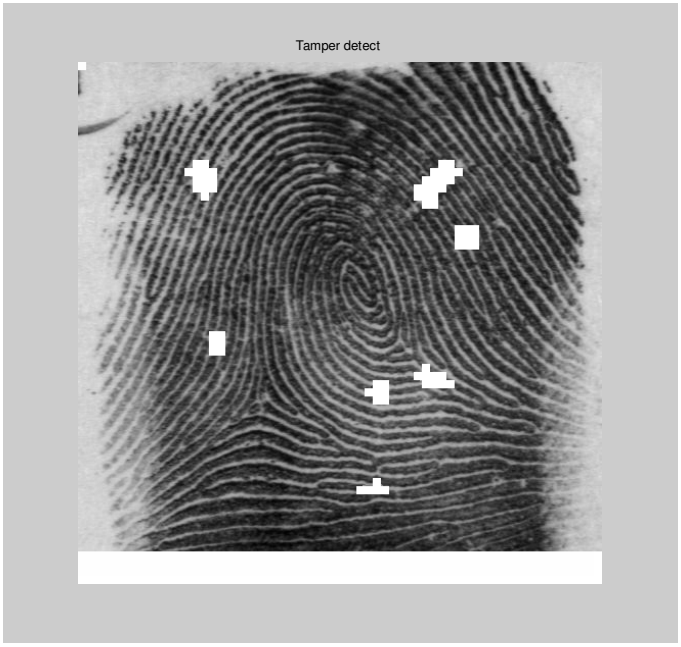


Figure 5.31. Level 2 detection – fingerprint2

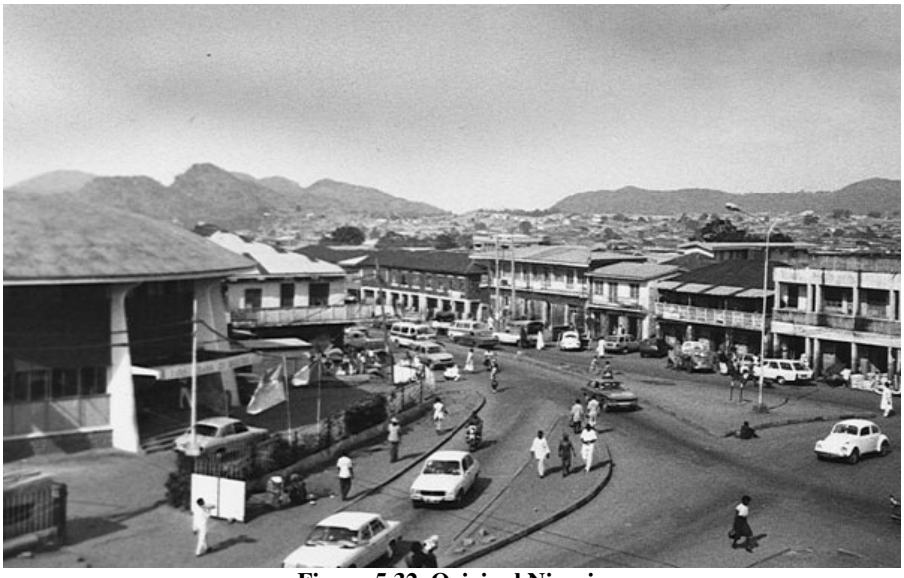


Figure 5.32. Original Nigeria

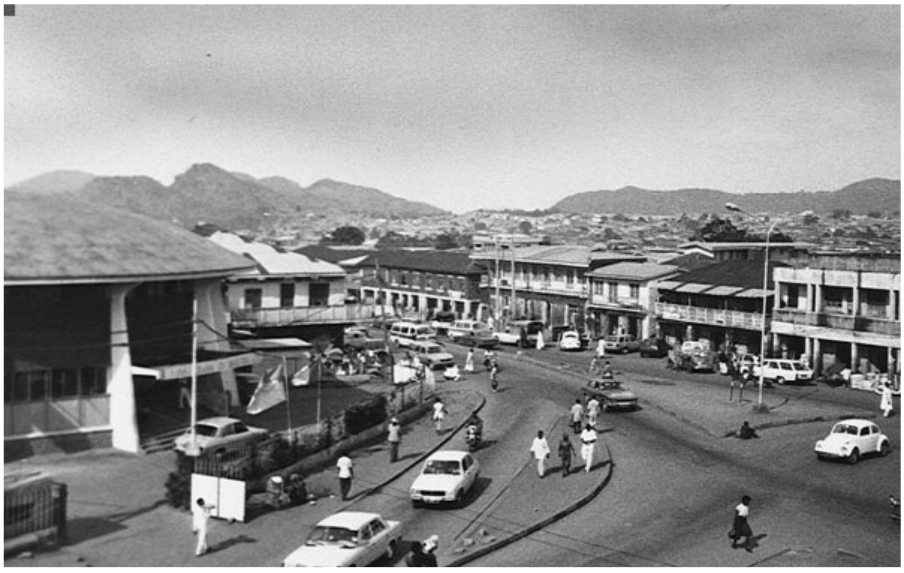


Figure 5.33. Watermarked Nigeria

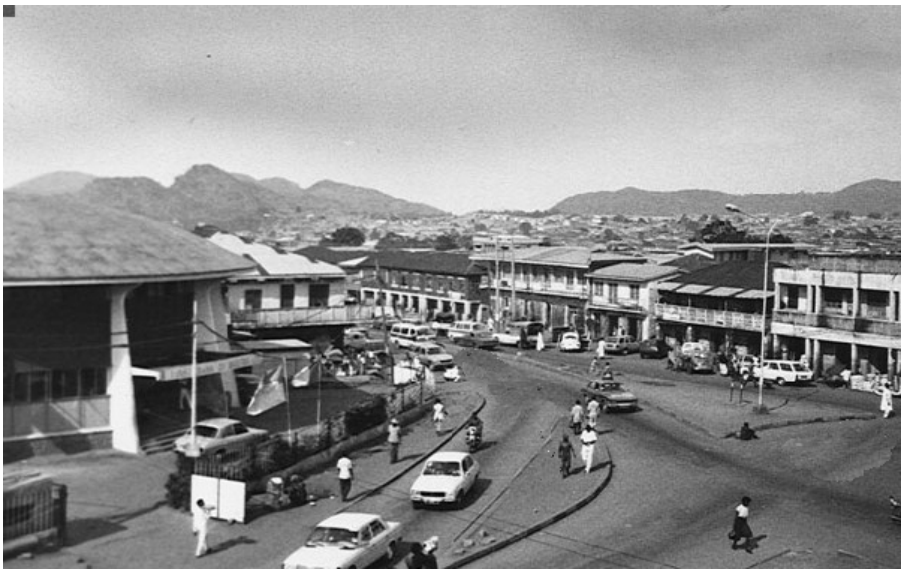


Figure 5.34. Tampered Nigeria

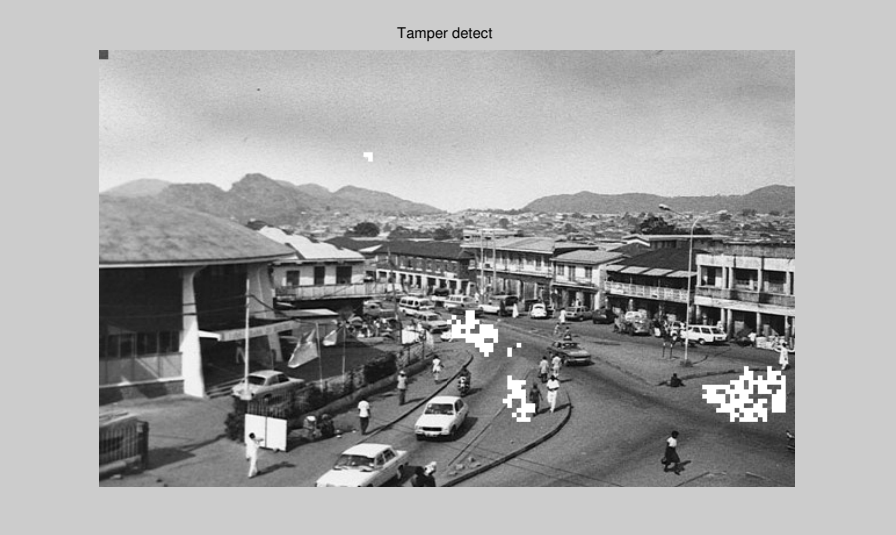


Figure 5.35. Level 1 detection- Nigeria

Nigeria was manipulated by removing some objects from the image. Four people were removed including one person and his shadow at the centre. A Volkswagen at the bottom right corner was also removed. A small area in the sky was also manipulated. Paintbrush, healing brush and cloning tools were used.

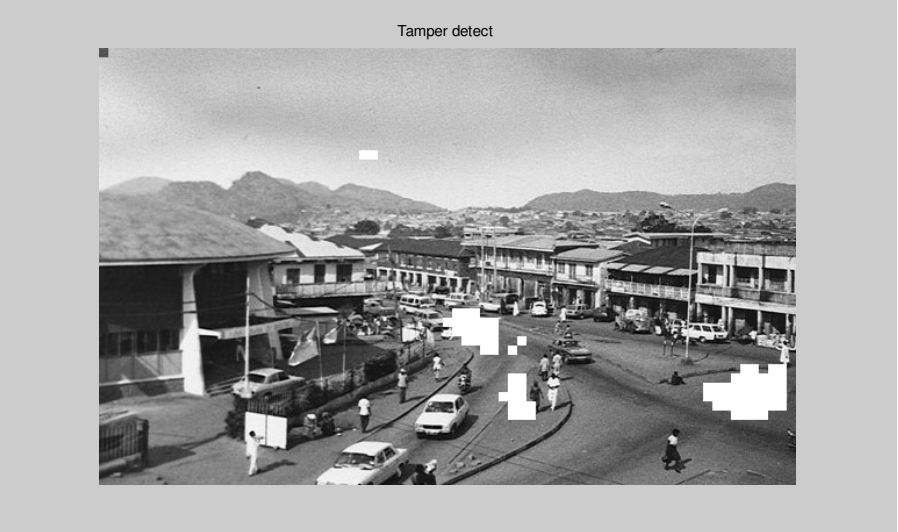


Figure 5.36. Level 2 detection - Nigeria

Table 5.2 shows the missing detection rate using level-1 and level-2 detection. For level-1 detection, we have a maximum of 16% of missing detection rate. We achieved at least 99.9% detection rate for level-2 detection.

	Ultrasound (800x600)	Fingerprint1 (512x512)	Fingerprint2 (512x512)	Nigeria (600x376)
Level1	10%	15%	13%	16%
Level2	0.1%	0.06%	0.02%	0.03%

Table 5.2. Miss detection rate

- Recovery in the middle region

We carried out another experiment to test the performance of our recovery algorithm to test when the tamper is made near to the centre of the image. For ultrasound images, this is highly likely because the region of interest happens to be in the centre of the image. We tampered with a watermarked image (k=3739) with tampering size of 20x20 pixels (figure 5.37) and 100x100 pixels (figure 5.39). For the 20x20 tamper, one block situated at the middle is not recovered. For 100x100 tamper, seven blocks were not recovered including the block in the middle of the image. The analysis of this will be discussed in chapter 6.

Figure 5.37. Tamper in the middle 20 x 20 pixel

Figure 5.38. Recovered image of figure 5.37

Figure 5.38 shows that the block in the middle is not recovered, while the blocks around it have been recovered.

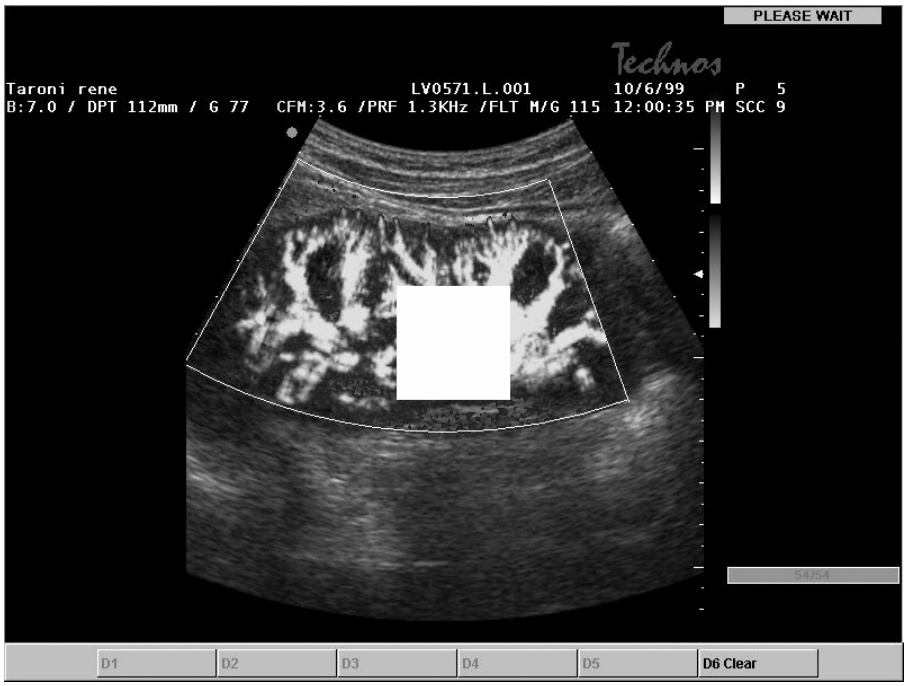


Figure 5.39. Tamper in the middle 100 x 100

Figure 5.40. Recovered image of figure 5.39

Figure 5.40 maintains that the block in the middle cannot be recovered by our method. In addition, some blocks at the vertical edge also cannot be recovered using our method.

- Recovery rate

An experiment to see the distribution of tampering, the size of the tampering effect and the rate of recovery was also carried out. We tampered with the watermarked (k=3739) image using spread-tampered blocks and single tampered blocks with a tampered area ranging from 10% to 50% as shown in figure 5.41 and figure 5.42. The spread-tampered blocks are the same size as the embedding blocks. Figure 5.43 shows the number of blocks that were not recovered from the single tampered block. We also changed the direction column-wise to see the effect. For a 10% column-wise single tampered block, we have a 100% recovery as in figure 5.44. This shows that the distance for those blocks and the mapped blocks were more than 1/10 of the image size. The results will be discussed further in chapter 6.

Figure 5.44 shows the number of un-recovered blocks for a single-tampered chunk. We obtained a 100% recovery for spread tampered blocks. The analysis will be discussed in chapter 6. Please see Appendix C for the recovered images.

5.5.6 Conclusion

We proposed a watermarking scheme that can detect and localise tampered and recovered images. The purpose is to verify the integrity and authenticity of medical images. We presented our watermarking procedures that include data embedding, tamper detection and recovery procedure. The experimental results demonstrate that the precision of tamper detection and localisation is close to 100% after level-2 detection. The tamper recovery rate is better than 86% for a less than half a tampered image.

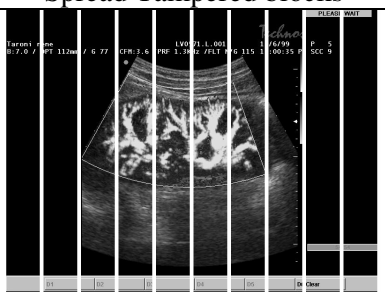
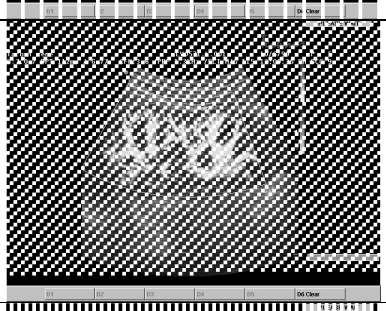
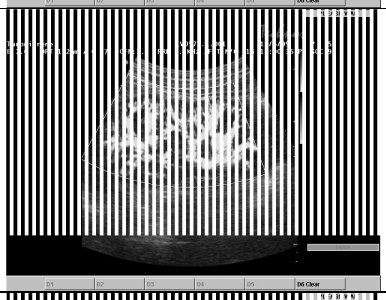
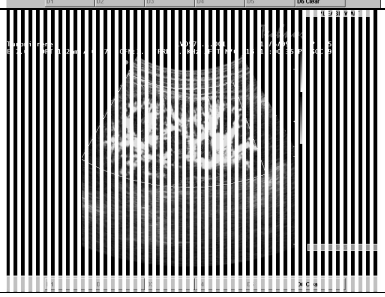
Tampering rate	Spread Tampered blocks	Recovered Image
10%		
20%		
30%		
40%		
50%		

Figure 5.41. Spread Tamper and recovered images

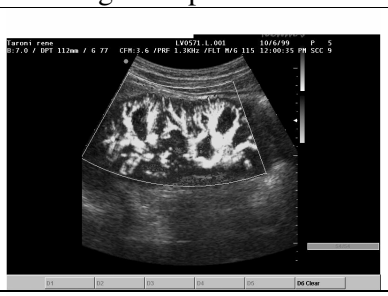
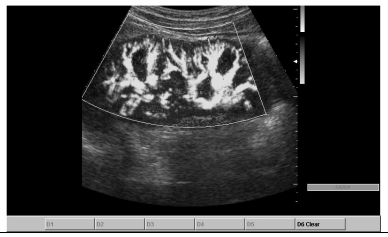
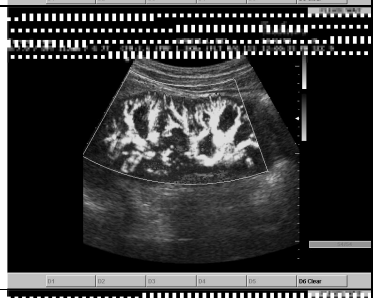
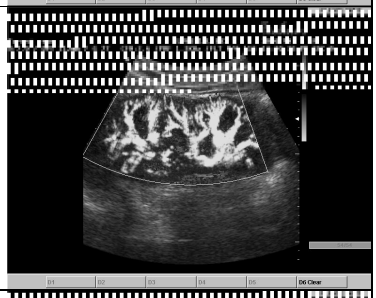
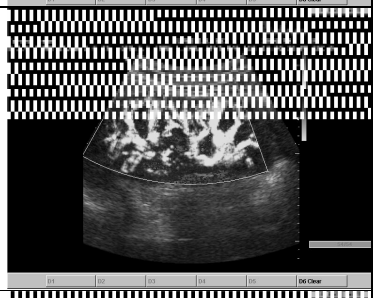
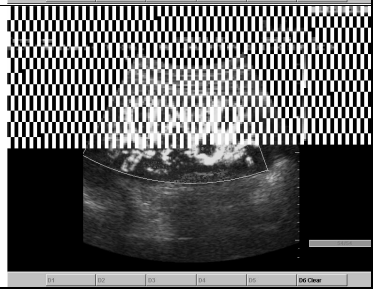
Tampering rate	Single tampered block	Recovered Image
10%		
20%		
30%		
40%		
50%		

Figure 5.42. Block tamper and recovered images

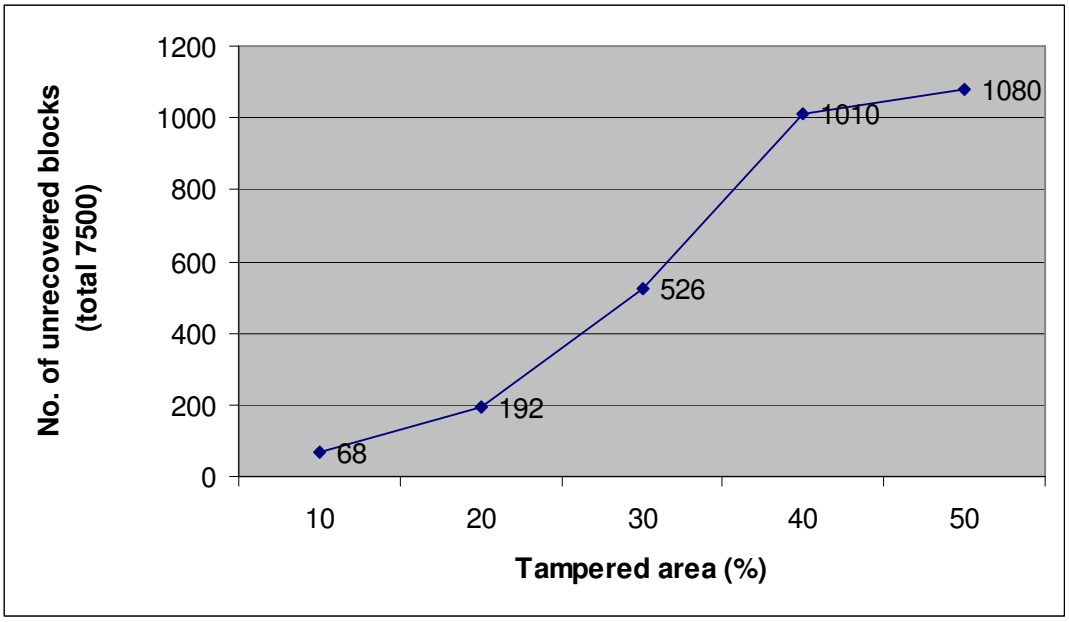


Figure.5.43. The number of un-recovered blocks for single tampered blocks

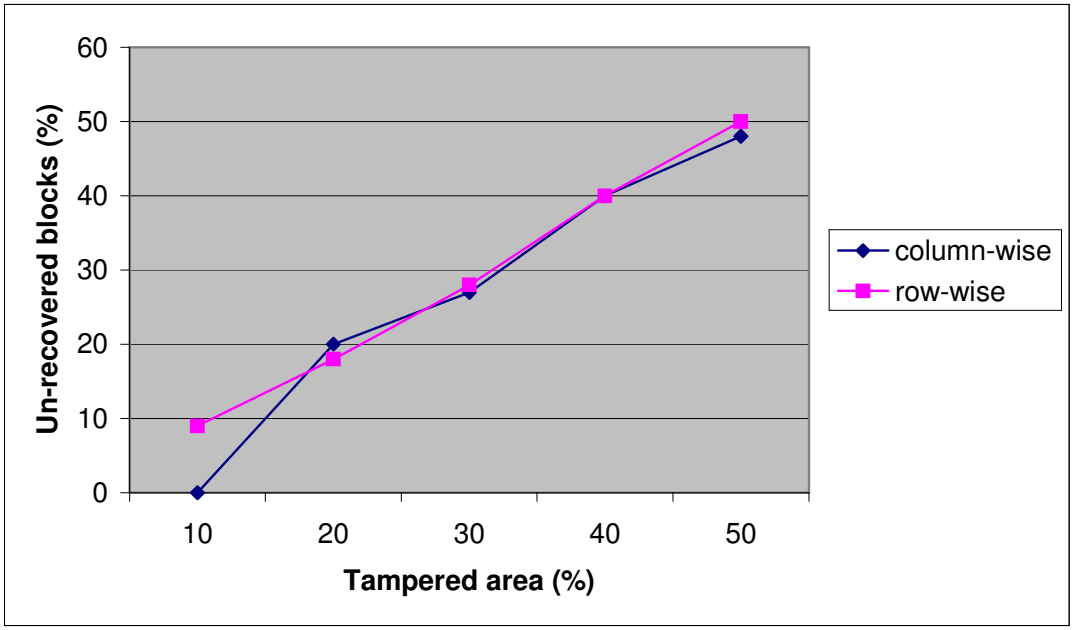


Figure 5.44. Percentage of un-recovered blocks for column and row- wise tampered

Chapter 6

Research Evaluation and Discussion

6.1 Introduction

This chapter discusses the evaluation of each of the proposed techniques and gives the final evaluation of the thesis. This chapter is structured as follows:

- Section 6.2 highlights the criteria to be used for evaluating the thesis
- Section 6.3 discusses experimental results from strict authentication watermarking (SAW) and evaluates the technique
- Section 6.4 discusses experimental results from the strict authentication watermarking with JPEG (SAW-JPEG) and evaluates the technique
- Section 6.5 discusses experimental results from authentication watermarking with tamper detection and recovery (AW-TDR) and evaluates the technique
- Section 6.6 presents the overall evaluation of the thesis

6.2 Evaluation Criteria

We evaluate our watermarking system according to the requirements outlined by Tong Liu, Zheng-ding Qiu (2002) and Lin, Chang (2000):

- Invisibility: The embedded watermark is invisible. It is the basic requirement of keeping the quality of marked images. The marked image must be perceptually identical to the original one under normal observation. It is a question of making sure that the visual impact of watermarking is as weak as possible so that the watermarked image remains identical to the originals.
- Detect tampering: An authentication watermarking system should detect any tampering in a marked image. This is the most fundamental property to reliably test the authenticity of the image. The system must be sensitive to malicious manipulations such as altering the image in specific areas.
- Security: The embedded watermark cannot be forged or manipulated. In such systems, the marking key is private and should be difficult to deduce from the detection of information. Insertion of a mark by unauthorised parties should be difficult.
- Identification of manipulated area or localization: The authentication watermark should be able to detect the location of altered areas, and verify other areas as authentic. The detector should also be able to estimate what kind of modification had occurred.
- Reconstruction of altered regions: The system may need the ability to restore, even partially, altered or destroyed regions in order to allow the user to know what the original content was of the manipulated areas.

6.3 Strict Authentication Watermarking (SAW)

- Invisibility and image quality

Invisibility is achieved as the maximum difference to the original image is only by one grey level. Figure 6.1 indicates how much visual difference for each grey level.

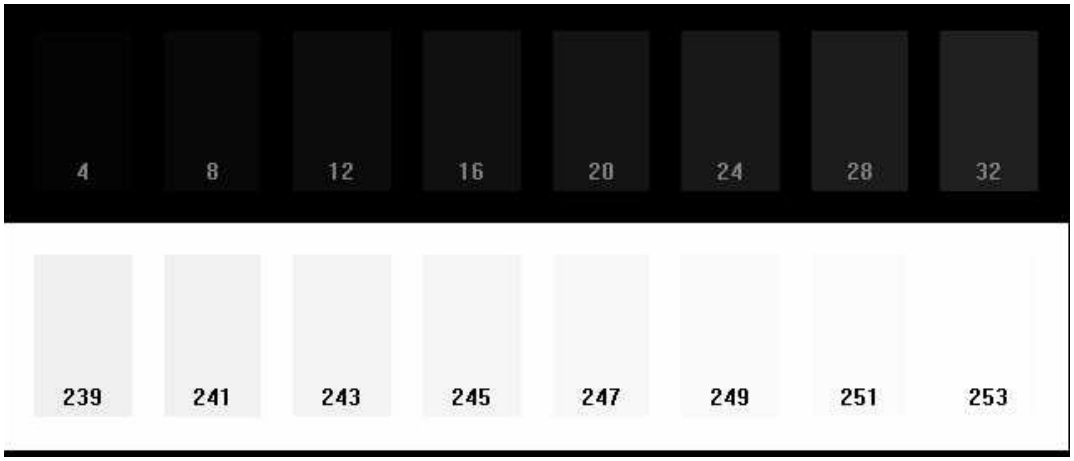


Figure 6.1. Grey levels

With only 256 bits embedded, distortion is very low with PSNR at 2.6 x 10^6 dB. However, the objective of any medical image would lie in a specific region of interest. Although as non-medics we do not know where the exact area of interest is for an ultrasound image, it is apparent that the region of interest only lies where the object projected by the ultra sound lies. We have made sure that the areas concerned are not included for embedding purpose. If the way the ultrasound image is taken and stored is changed then the technique will not be relevant anymore.

- Security

The security for this technique depends on the security of the key used. Bigger key space will increase security, but will result in difficulty to manage them. As Tong and Zheng-ding (2002) stressed, any algorithm alone cannot guarantee the security of the system. It is necessary to define a set of scenarios and specifications describing the operation and rules of the system, such as management of the keys or the communication protocols between consultants, doctors, technicians and so forth.

- Tamper detection

The technique will detect tamper by comparing the signature produced by hashing the region of interest. Any changes inside the area will have a significant change in the signature produced by the hash function. The system however will not detect changes made outside the region of interest. If this is of concern, then the system can be made to

produce a signature of the whole image, which is embedded in the region of non-interest. At the receiver's side the watermark, which is the signature, is extracted, and reverts the values to the original state. The signature is then calculated and compared to the received.

- Reversibility

The reversibility is achieved by exploiting the characteristic of ultrasound images. With plenty of redundant areas outside the region of interest with pixel values of zeros, this helps to achieve reversibility without having to employ a sophisticated technique unique from the literature.

- Capacity

The SAW embedding scheme achieves a high capacity for watermarks to be embedded. Table 4.3 shows that 510,000 bits could be embedded in an 800x600x8 image with distortion less than 32 dB. This gives an embedding rate of 1.06 bits/pixel. This makes the scheme superior to that of Guo and Zhuang (2003) where their embedding rate is 0.0054 bits/pixel. In applications where the watermark is embedded in a RONI, potentially an even higher embedding rate could be achieved.

The time to calculate the digital signature for a large image could be a disadvantage to this method. It is very compute intensive. For example Cao et al. (2003) noted that the time required for the sending and receiving sites for processing a digital mammogram could range from 40s for the segmented image of 7 Mb to 3 min for original 36 Mb image using Sun Sparc 690MP multiprocessor machine.

- Recovery

This method, although capable of detecting single bit changes within an entire image, has no capability to determine where the tamper has occurred or restore the tampered image to its original.

6.4 Strict Authentication Watermarking with JPEG Compression (SAW-JPEG)

- Invisibility and image quality

In this technique, the same amount of information as in SAW, is embedded into the image, however the number of embedded bits is significantly higher, in this case 64 times. Invisibility is maintained as only one grey level is involved. Using only the least significant bit, which is the eighth bit, the distortion level is kept very low with PSNR at 6.1×10^4 dB.

- Security, tamper detection and reversibility

The security of this technique also depends on the key. The technique is reversible and has excellent tamper detection, but no capability for reconstruction. This is the same with SAW.

- Robust to JPEG

The technique is robust to compression and was tailored to JPEG. It survives compression up to a specific level for a watermark embedded in the LSB. This approach appears unique than that reported in the literature.

- Informed authentication

The SAW technique uses informed authentication, that is it calculates the digital signature using information from the original image. In order to authenticate, the received digital signature should be compared against a new digital signature calculated for the original image. In telemedicine applications, the original image may not be available and so the digital signature must be found for the received image and compared to the received digital signature. The scheme is shown in Figure 6.2.

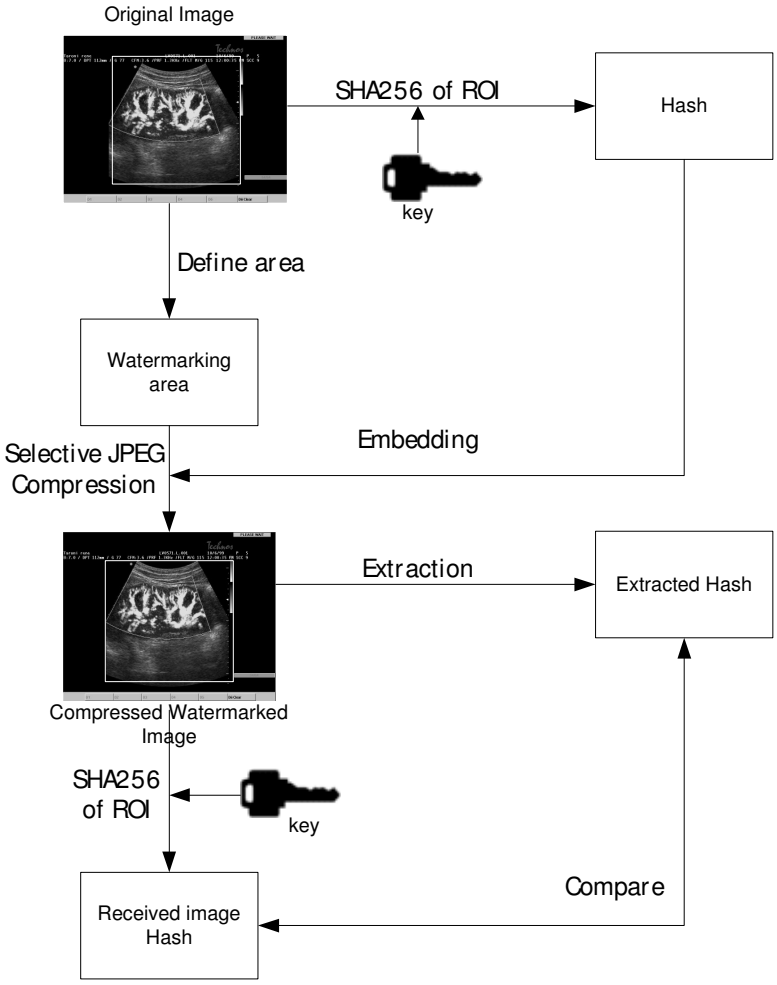


Figure 6.2 Final scheme for SAW-JPEG

6.5 Authentication Watermarking with Tamper Detection and Recovery (AW-TDR)

- Invisibility and image quality

Invisibility is achieved in this technique, by restricting modification to only the LSB. The embedding rate is 1 bit per pixel. The quality of the watermarked image is good with PSNR at 54 dB. Unlike the previous techniques the authentication and recovery bits are embedded in the ROI as well as RONI.

This scheme will be unacceptable in applications where there must be no modification to the image, and in such cases the watermarks could be embedded into a RONI if such a suitable region exists. It is clear that the image is modified, but the effect is minimal and such a change should be imperceptible to clinicians and would not affect diagnostic accuracy. There are currently no standards or guidelines for acceptable changes to watermarked images. Acceptable limits could be determined through clinical validation. This would require comparison of a sufficiently large number of images by separate clinicians to determine whether perceptible differences exist between images with and without watermarks and if such differences affect clinical decisions.

The design of such a study is described in Appendix A, but it was beyond the scope of this work to carry the study out.

- Security

The strength of this technique depends on the key and the use of k < Nb may not provide sufficient security. For an 800 x 600 image, there are approximately 1600 keys. This can easily be defeated with brute force attack. Using k > Nb will result in loss of key uniqueness, where more than one key can produce equivalent watermarks for the same image.

- Tamper detection and tamper localization

Tamper detection depends on the probability of getting the parity bit, p and average intensity, v. The probability of miss for level 1 detection is ½ x ½ = ¼ = 0.25. So the probability of missing detection for 6x6 block at level 2 detection would be $(¼)^4$ = 1/256 = 0.0039. For level-1 detection, if the type of error is parity error, then we are sure that the sub-block is indeed tampered. For level-2 detection, if the type of error is an intensity relationship error, we cannot be sure whether the sub-block under inspection is tampered with or other sub blocks within the same block are tampered. However, some pixels within the block must be in error. Thus, in case the tampered sub-block is not detected in level 1 inspection, the whole block will be marked tampered after level 2 inspections.

From the experimental results in Table 5.2 we find that the maximum missing rate after level-1 detection is 16% (probability of 0.16) and after level 2 detection is 0.1%. From the results, we can conclude that our method can detect tamper of size 3 x 3 pixels with a probability of 0.84 and tamper of size 6 x 6 pixels with a probability of 0.99. Using a mean intensity parity bit , it also ensures that we do not have a false alarm.

- Reconstruction

Reconstruction is achieved by embedding the recovery bits in a block some distance away from the original block as suggested by Fridrich and Goljan (1999). From the experimental results, it showed that the recovery bits were not embedded in blocks situated in the same column, but with some percentage in the same row. Those in the same row must have an odd distance from the original, because the way we spread the tamper was by using the same size, as the block use for embedding and the distance from each other were at least one block. If we change the tamper block size in the spread-tampered blocks, then we may have a different result.

42	21	22	23	24	25	26	43
41	20	7	8	9	10	27	44
40	19	6	1	2	11	28	45
39	18	5	4	3	12	29	46
38	17	16	15	14	13	30	47
37	36	35	34	33	32	31	48

(a)

31	28	3	26	1	24	47	6
8	5	42	17	40	15	22	29
33	30	19	48	23	38	45	4
10	7	44	21	46	13	20	27
35	32	9	34	11	36	43	2
12	37	14	39	16	41	18	25

(b)

Figure 6.3 (a) Spiral numbering of blocks (b) mapping with k=23, shaded blocks will not be recovered for 4x4 blocks tamper

We could also change the way we order the block number. Will it make a difference in the distribution of blocks? For example we could start the block at the centre and move in a spiral manner.

We anticipate the tamper is likely to be in the middle as the feature of ultrasound images has the region of interest in the middle. Our preliminary results show that the block spiralling and starting in the middle will have a greater chance of recovery compared to our proposed method. If we tamper with the 2x2 block in the middle, we will have two blocks that cannot be recovered, giving us 2/4 =50% recovery rate (refer to Figure 5.12(b)). With the spiral method we will have a 100% recovery for 2x2 block tamper in the middle of the image as in figure 6.3(b). If we have 4x4 blocks tampered, the proposed method will only have a 5/16 = 31% recovery rate, while the spiral method will give a higher recovery rate of 12/16 = 75%.

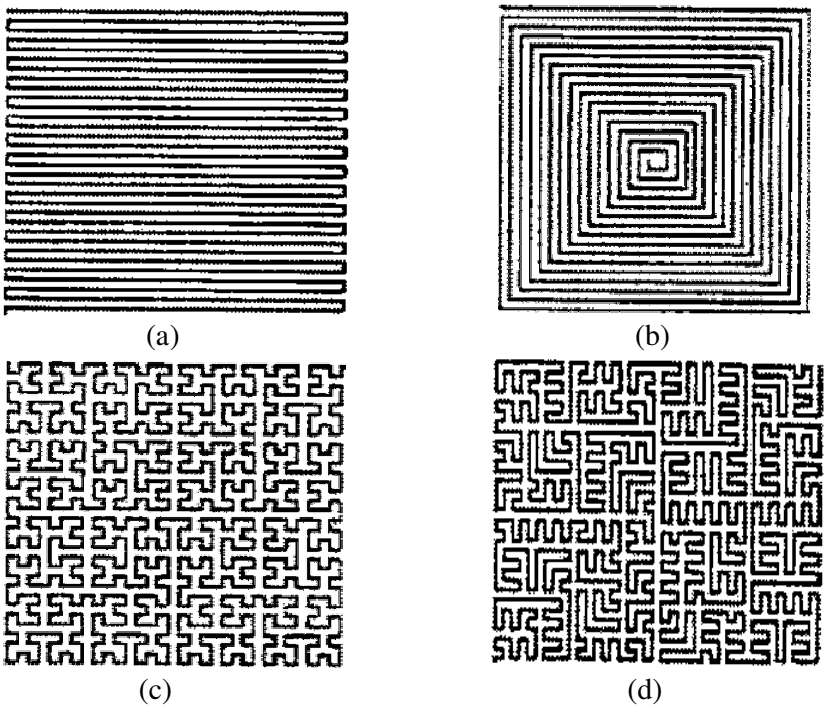


Figure 6.4 (a-b) Typical scans, (c-d) Key generated Peano scan

Considering the scan technique may further strengthen the method. In place of the simple raster scan, other methods such as those shown in Figure 6.4 could be used. Figure 6.4(a) is a reverse raster scan and appears to offer few advantages. Figure 6.4(b) is a spiral scan. Although advantageous for blocks in the middle, it retains the

weaknesses of raster scan at the edges. The scan method of Figure 6.4(c) is the so-called Peano scan with the Peano-Hilbert variant shown in Figure 6.4(d). These are localised scan methods and could ensure blocks are relocated a minimum distance away. However, work on scanning has been conducted before and applied in other fields of research. The type of ordering or scanning called the Peano-Hilbert plane-filling curve as shown in figure 6.4(d) has been applied in a compression technique by Lempel and Ziv (1986). The possibilities of using the Peano-Hilbert scan for our watermarking technique can be explored further to find the optimal recovery point.

- VQ counterfeiting attack

The scheme is considered to be robust against a VQ counterfeiting attack by adding another level of authentication. Although the attack will successfully defeat level-1 and level-2 inspection, the attack will not survive level-3 detection (Section 5.5.3) as long as the key is kept secret.

6.6 Final proposal for AW-TDR

This section presents the final proposal for AW-TDR, describing the preparation of blocks, the embedding algorithm and the location plan for authentication bits and recovery bits.

Figure 6.5 shows the preparation of blocks B for embedding authentication bits and blocks C for embedding recovery bits. Blocks B will be mapped on to blocks C using an invertible function, as described in chapter 4 and chapter 5. Blocks B and C can be of different sizes.

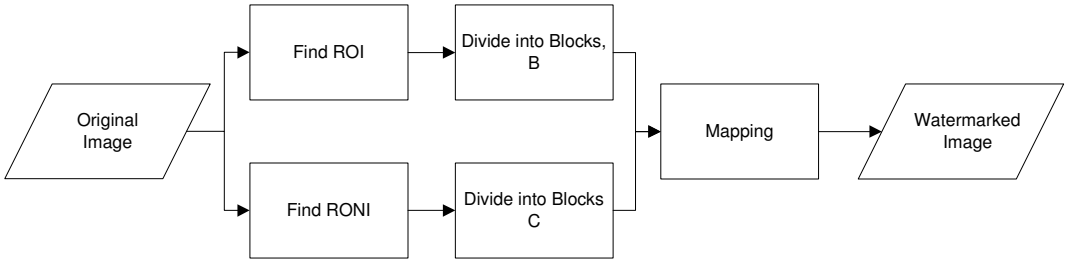


Figure 6.5 Mapping blocks in RONI for intensity embedding

Figure 6.6 shows the final proposal for the embedding algorithm for AW-TDR. The difference to the earlier version is that only the ROI is considered for the authentication process. The rest of the image will be used to embed the recovery bits.

Figure 6.7 shows the authentication bits, v and p will be embedded in the ROI and the recovery or reconstruction bits will be embedded in the RONI. We suggest the block size in ROI to be 4 x 4 pixels, with a sub-block of 2 x 2 pixels and the block size in RONI where the recovery bits to be embedded to be 2 x 1 pixels.

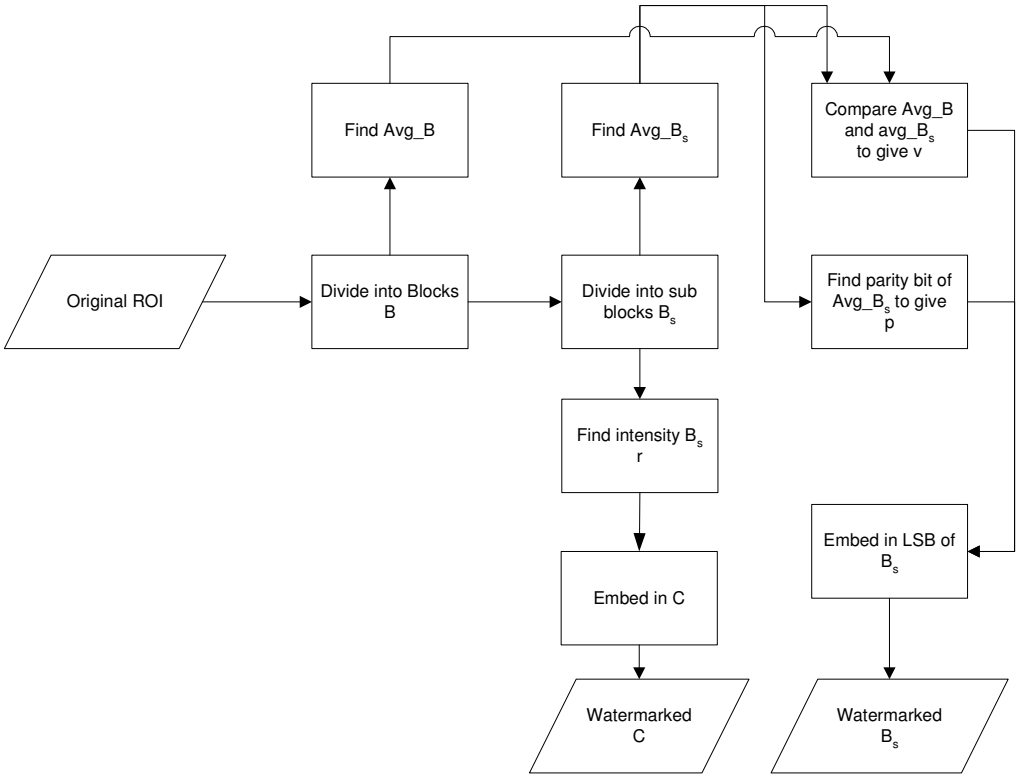


Figure 6.6 Final AW-TDR embedding

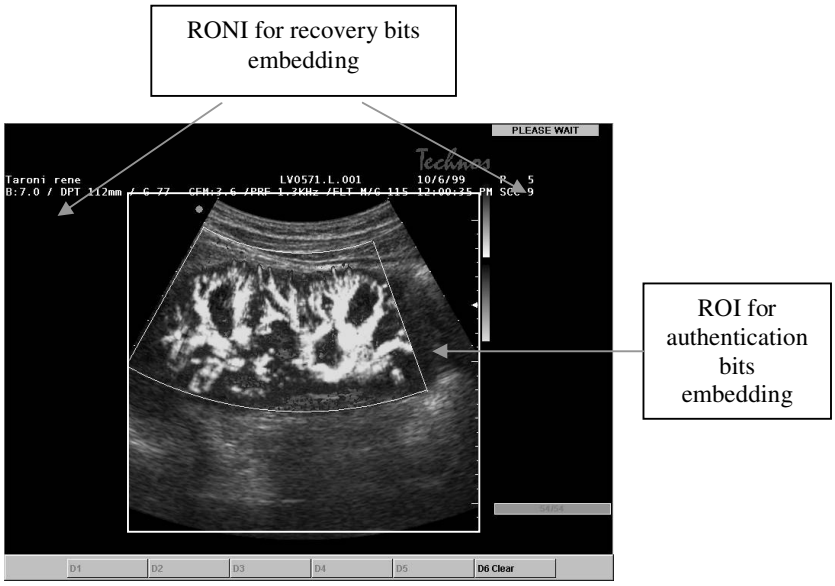


Figure 6.7 Location of bits for embedding

Figure 6.8 shows the location of the authentication bits, v and p in the LSB of pixels *p1* and *p2* respectively. The recovery bits, *r = r1r2r3r4r5r6r7*, form a seven-bit value that is the average of the sub-block, Bs. The seven bits *r1, r2, r3 and r4* are then embedded in the four LSBs of pixel *c1* and *r5, r6 and r7* are embedded in the four LSBs in pixel *c2*. For example if *p1*= 153, *p2*= 155, *p3*= 200 and *p4*= 180, *r* = 172 = 10101100_2. If *c1* and *c2* is 0 initially, *c1* will be 1010 and *c2* will be 1100 after embedding.

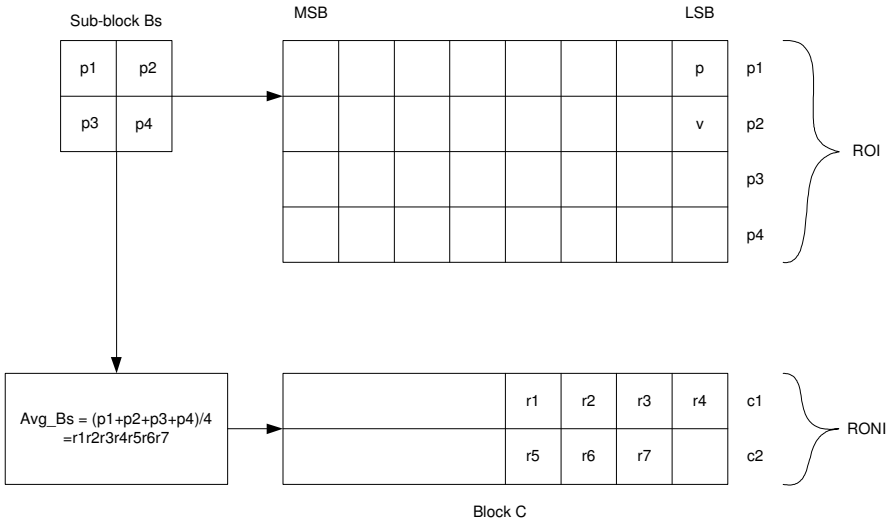


Figure 6.8 Location of bits in the corresponding pixels

The final proposal will enhance AW-TDR in three aspects:

1. Image quality in the ROI will be improved as the maximum change is only 2 bits in every 4 pixels, or embedding rate of 0.5 bits/pixel
2. Recovery rate will also be better since the recovery bits are located outside the region of interest. The disadvantage is that, only manipulation done in the ROI will be detected
3. The quality of the reconstructed image will be enhanced since the average of 2 x 2 pixels (please refer to figure 5.10) would be used to reconstruct the tampered image.

6.6 Summary

This chapter reviewed and evaluated the proposed schemes SAW, SAW-JPEG and AW-TDR.

SAW, a reversible watermarking scheme being capable of verifying authenticity and integrity of ultrasound images is proposed. It also allows recovery of the original image at the receiver. The SAW embedding scheme has a high capacity for embedding a watermark, in ultrasound images at around 1.06 bits/pixel. This makes the scheme superior to that of Guo and Zhuang (2003), which had an embedding rate of 0.0054 bits/pixel. Since the watermark is embedded in RONI, potentially an even higher embedding rate could be achieved.

SAW-JPEG is also a strict-authentication watermarking scheme and is robust to certain levels of JPEG compression. This work appears unique and there are no reports of embedding a watermark in the LSB to be robust against JPEG compression. This is because it is almost impossible for any image to survive their least significant bits after the quantization process. This technique is only unique to images with some areas of constant pixel values such as in ultrasound images. The method is based on exploiting the image feature that is able to survive compression and so may be modified to be robust over other compression schemes.

	Requirements	SAW	SAW-JPEG	AW-TDR
MANDATORY	Invisibility	Yes	Yes	Yes
	Detect tamper	Yes	Yes	Yes
	Security	Key	Key	Key
DESIRABLE	Reversibility	Yes	Yes	No
	Compression	No	Yes	No
	Localise tamper	No	No	Yes
	Reconstruction	No	No	Yes
OTHER	Distortion	2.6×10^6 dB	6.1×10^4 dB	54 dB

Table 6. 1. Summary of proposed watermarking

AW-TDR is a watermarking scheme that can detect and localise tamper and recover the image. The experimental results demonstrate that the precision of tamper detection and localisation is close to 100% after level-2 detection. The tamper recovery rate is better than 86% for a less than half tampered image.

The three schemes have been implemented on ultrasound images and the results have shown to be successful authentications of ultrasound images with the respective capabilities shown in Table 6.1. The mandatory requirements for watermarking identified in Table 2.2 were met and additional functionalities were developed.

From the evaluation and comparison of the three schemes proposed, this chapter determines current weaknesses and proposes modifications for enhanced versions. This includes modifying SAW-JPEG for blind authentication and a scheme for AW-TDR to have minimal embedding in the ROI.

Chapter 7

Conclusions and Reflections

7.1 Introduction

This chapter is structured as follows:

- Section 7.2 summarises the research
- Section 7.3 highlights the contributions and limitations of this thesis
- Section 7.4 gives suggestions for continuing the research in future work
- Section 7.5 summarises this chapter
- Section 7.6 reflects on the PhD process

7.2 Summary of Research

While the purpose of fragile watermarking and digital signature systems are similar, watermarking systems offer several advantages compared to signature systems (Memon et al. 1999) at the expense of requiring some modification (watermark insertion) of the image data. As a watermark is embedded directly into the image data, no additional

information is necessary for authenticity verification. This is unlike digital signatures, since the signature itself must be bound to the transmitted data. The critical information needed in the authenticity testing process is discreetly hidden and more difficult to remove than a digital signature, or even if it is removed, it remains possible to detect that it has been tampered with. Also, digital signature systems view an image as an arbitrary bit stream and do not exploit its unique structure. Therefore a signature system may be able to detect that an image had been modified but cannot characterise the alterations. Many watermarking systems can determine which areas of a marked image have been altered and which areas have not, as well as estimate the nature of the alterations.

7.2.1 Summary

The advantages of watermarking compared to digital signature may be summarised as:

- No additional information/overhead needed
- Able to localise tamper or alterations
- Able to restore tampered images

The topic of watermarking in medical images has received relatively little research and analysis of the literature identifies that works remains to be undertaken on:

- Methods that can be used to solve the problem of watermarking medical images
- Methods that can be used to detect tampering
- Methods that can be used to recover tampered images

It is proposed that watermarking is reversible or conducted in the region of non-interest to make sure it will not change the diagnosis. This research is concerned with the issue of authenticating medical images. The issue of tamper detection and recovery are also of interest in this research.

7.2.2 Statement of the Problem

A major concern of the users of medical digital images is that it would be easy to modify the contents. Current cryptography methods can detect tampering by generating

an authentication for the image, but at the expense of an overhead for its storage. It may also be separated from the image.

The aim of this research was to develop a method where the authentication may be embedded within the image itself - digital watermarking. The work considered watermarking methods that might be robust against the effect of applying lossy compression (e.g. JPEG) to such images. The methods were then enhanced to provide information on the location of the tampering and have an ability to return an approximate rendering of original image.

7.2.3 Purpose of the Study

The purpose of this study was to investigate and develop watermarking techniques suitable for medical imaging. This includes:

- The development of watermarking algorithms for:
  - Strict authentication
  - Strict authentication with JPEG compression
  - Tamper detection and recovery
- An implementation of the techniques on selected medical image modality.

7.3 Contributions and Limitations

The contributions of this thesis will be highlighted from each proposed scheme; Strict Authentication Watermarking (SAW), Strict Authentication Watermarking with JPEG Compression (SAW-JPEG) and Authentication Watermarking with Tamper Detection and Recovery (AW-TDR).

- Strict Authentication Watermarking (SAW)

The contributions of this scheme (SAW) cover three different elements of the research process: theory, practice and outcome. The integration of the digital signature as the watermark, the use of region of non-interest together with random mapping as the watermarking region is a novel approach in authentication watermarking. The scheme

detects tamper by comparing the signature produced by hashing the region of interest. Any changes inside the area will have a significant change in the signature produced by the hash function. The system however will not detect changes made outside the region of interest. If this is of concern, then the system can be made to produce a signature of the whole image, which is embedded in the region of non-interest. At the receiver's side the watermark, which is the signature, is extracted and reverts the values to the original state. The signature is then calculated and compared to the received version.

Reversible watermarking for ultrasound images provides a lossless authentication watermark, which ensures the integrity of the image data without permanent loss of image fidelity. The reversibility is achieved by exploiting the characteristic of the ultrasound image. The abundance of redundant areas outside the region of interest with pixel values of zeros helps to achieve reversibility without having to employ a sophisticated technique and again is unique from studies reported in the literature

The SAW embedding scheme achieves a high capacity for watermarks to be embedded. Table 4.3 shows that 510,000 bits could be embedded in an 800x600x8 image with distortion less than 32 dB. This gives an embedding rate of 1.06 bits/pixel. This makes the scheme superior to that of Guo and Zhuang (2003) where their embedding rate is 0.0054 bits/pixel. In applications where the watermark is embedded in a RONI, potentially an even higher embedding rate could be achieved.

- Strict Authentication Watermarking with JPEG Compression.

The contribution that emerges from SAW-JPEG is an embedding technique in the LSB that can survive JPEG quantization process. The use of knowledge of the quantization algorithm to allow the DC coefficient to be unchanged through the compression/decompression process is used for the proposed watermarking scheme. There has been no attempt in studies from the literature to embed watermark in the LSB to be robust against JPEG compression before, as it is almost impossible for any image to survive their least significant bits after quantization process.

The SAW-JPEG technique uses informed authentication, that is it calculates the digital signature using information from the original image. In order to authenticate, the received digital signature should be compared against a new digital signature calculated from the original image. In telemedicine applications, the original image may not be available and so the digital signature must be found for the received image and compared to the received digital signature. The new scheme is shown in Figure 6.2.

- Authentication Watermarking with Tamper Detection and Recovery (AW-TDR)

AW-TDR is an efficient and effective digital watermarking method for image tamper detection and recovery. The contribution of this method is the integration of four concepts derived from the literature; 1) block-based (Fridrich and Goljan 1999); 2) separating authentication bits and recovery bits (Lin and Chang 2001); 3) hierarchical (Celik et al 2002); and 4) average intensity as image feature (Lou and Liu 2000). The method is efficient as it only uses simple operations, such as a parity check and comparison between average intensities. It is effective because the scheme inspects the image hierarchically with the inspection view increasing along with the hierarchy so that the accuracy of tamper localisation can be ensured. This scheme can perform both tamper detection and recovery for tampered images. Tamper detection is achieved through a block-based, inspection and recovery of a tampered block. It relies on its feature information hidden in another block that can be determined by a one-dimensional transformation.

The scheme is considered to be robust against a VQ counterfeiting attack by adding another level of authentication. Although the attack will successfully defeat level-1 and level-2 inspection, the attack will not survive level-3 detection (Section 5.5.3) as long as the key is kept secret.

A modification for AW-TDR to have minimal embedding in the ROI was proposed. The final proposal will enhance AW-TDR in three aspects:

1. Image quality in the ROI will be improved as the maximum change is only 2 bits in every 4 pixels, or embedding rate of 0.5 bits/pixel

2. Recovery rate will also be improved since the recovery bits are located outside the region of interest. The disadvantage is that only manipulation done in the ROI will be detected
3. The quality of the reconstructed image will be enhanced since the average of 2 x 2 pixels (please refer to figure 5.10) would be used to reconstruct the tampered image.

Limitations:

- SAW and SAW-JPEG are only applicable to images with RONI (e.g., ultrasound images).
- SAW and SAW-JPEG do not allow any bit change in the ROI. This implies that any legitimate image processing that changes the spatial value of the image will result in the image being considered as tampered.
- Security of AW-TDR depending on keys only. The use of $k < Nb$ may not provide sufficient security. For an 800 x 600 image, there are approximately 1600 keys, which can easily be defeated with a severe attack.
- The three proposed schemes only consider LSB as the watermarking domain and so remain as fragile schemes
- AW-TDR is designed to detect local manipulations such as cut and paste, repainting and erasing. Global manipulations such as compression will result in the whole image is considered tampered.

Table 7.1 shows the summary of the contributions from the thesis.

Research Process	Contribution
Theory	1. The integration of a digital signature as a watermark and region of non-interest and random mapping as embedding area. 2. The use of an image feature for reversible watermarking. 3. The use of knowledge of the quantization algorithm to allow the DC coefficient to be unchanged through the compression/decompression process for watermarking. 4. The integration of four concepts introduced from the literature; block-based; separating authentication bits and recovery bits; hierarchical detection; and average intensity as image feature for detection and recovery.
Practice	1. Development of a scheme that is able to authenticate medical images with reversible capability. 2. Development of a scheme that is able to authenticate medical images and can survive a certain level of JPEG compression. 3. Development of a hierarchical scheme that is able to localise tamper with recovery capability.
Outcome	1. Strict Authentication Watermarking (SAW) 2. Strict Authentication Watermarking with JPEG Compression (SAW-JPEG) 3. Authentication Watermarking with Tamper Detection and Recovery (AW-TDR)

Table 7.1 Thesis Contributions

7.4 Further Research

The research has opened up a number of possibilities for future work. The suggested list is provided below:

- Improvement on security for AW-TDR. The use of k < Nb may not provide sufficient security. For an 800 x 600 image, there are approximately 1600 keys, which can easily be defeated with a severe attack.
- A variety of different error correction codes can be applied to improve on the quality of recovery bits (e.g., Hamming codes, turbo codes, and trellis codes). This metadata can be represented as a watermark. For example, a Reed Solomon ECC code can be used to generate parity bytes for each row and column of an image (Lee and Won 1999, Lee and Chen 2002). These parity bytes can be embedded as a watermark in the two significant bit planes of the image.
- To include reversible watermarking techniques, for example the one proposed by Goljan et al (2001) and at the same time maintain tamper detection and recovery for authentication bits embedded in the ROI.
- The possibilities of using the Peano-Hilbert scan for the AW-TDR watermarking technique can be explored further to find the optimal recovery point.
- As compression is acceptable in a medical standard such as DICOM, investigation on embedding in other domains such as DCT (used in JPEG) and wavelet (used in JPEG2000) should be considered to make sure the watermark is robust against those compression schemes.
- As in a radiology image lossy compression (Wong et al 1995), there exists no legal standards for regulating how much distortion induced by watermarking system can be accepted. To be acceptable, a watermarking system requires thorough clinical validation tests. Such tests must be carried out on a large number of images and should involve a number of clinicians to assure the diagnostic accuracy is not jeopardised by such distortion. We propose a study in Appendix A to find out whether or not our watermarking scheme interferes with clinical diagnosis.
- Application on other image modalities such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), single photon emission computerised tomography (SPECT), nuclear medicine (NM), digital subtraction angiography (DSA), and digitalflurography (DF).

- Issues in practical application in a real-world hospital environment need to be investigated before watermarking could possibly be used.

7.5 Summary

This research has demonstrated that watermarking can provide authentication for medical images. Three fragile watermarking schemes SAW, SAW-JPEG and AW-TDR have been investigated.

This research has extended current technology in fragile watermarking by providing a high capacity, reversible authentication service for medical images. SAW-JPEG demonstrates a technique to embed information in the LSB that can survive JPEG quantization process.

A hierarchical image authentication watermark (AW-TDR) is proposed that is able to validate the source of the image, verify its integrity, and when the integrity verification fails, determine the altered image regions. This approach overcomes the security problems associated with previous independent block-based authentication watermarks, while retaining their tamper localisation properties. The algorithm has been shown to provide security against vector-quantization (collage) counterfeiting attacks and accurate localisation of altered image regions.

Three schemes have been implemented on ultrasound images and the results have shown successful authentication of ultrasound images with the respective capabilities shown in table 6.1. The mandatory requirements for watermarking, identified in Table 2.2, were met with some additional functionality.

From the results and evaluation, it can be concluded that this research has met the objectives outlined in chapter 1.

7.5.1 Watermarking Future

Watermarking is still not a fully mature and understood technology, and many questions remain unanswered. However, the interest in watermarking is high, both from the academia and industry. The interest from academia is reflected in the number of publications on watermarking and in the number of conferences being held on watermarking and data hiding. The interest from industry is evident from the number of companies that have funded research in the field.

There exist enough applications where watermarking can provide working and successful solutions. Specifically for audio and video, it seems that watermarking technology will become widely deployed (MusicTrace 2005). The DVD industry standard, for example, will use watermarking for copy protection system (DRM Watch Staff 2004). Similarly, plans exist to use watermarking for copy protection for Internet audio distribution. Broadcast monitoring using watermarking is another application that will probably be widely deployed for both audio and video (Digimark 2001). Whether the development of watermarking technology will become a success story or not, remains to be seen, but it is a research area that is fast developing.

7.6 Personal Remarks

7.6.1 My PhD Journey

I am a lecturer at a university in Malaysia. I started my PhD when I was 35 years old, married with 5 children. It was not a straightforward decision to do a PhD. Although the university encourages people to do their PhD as early as possible in order to increase and enrich research activities in the country. A decision to leave your home for a period of over 3 years has to be based on a strategic plan as it will not only involve me, as the PhD candidate, but my family members too. So the initial plan was I would pursue my PhD, my husband will pursue his sub-specialty and three of the children would follow us.

I arrived in London in January, when the temperature was below 10 degree Celsius. It was soaring 33 degree at home! My first worry was the thought that I will not survive in this weather, not the PhD. We went through the immigration office who asked some absurd questions such as 'How much money do you have?' and 'Are you going back to your country after you have completed your studies?' The impression we got was that we are not welcome here. My first big hurdle was to find suitable accommodation for a family with three children. Many landlords turned us down because we have children. Once we lost our deposit money to an agent when the landlord did not allow us to move in at the last minute. We felt as if the system in this country was all against us. I felt like taking a drastic decision to quit the idea of doing a PhD. We eventually found accommodation after 3 tearful weeks with the help of a colleague from the department. We have to make do with a small space.

Being in a different country having a different culture is difficult, but it also enriches our learning process. Those experiences make us stronger as a person. I learnt that I had to take responsibility for my own learning. I was not used to deciding for myself. I felt lost, like being left in a forest and asked to find my own way. I would ask 'But where should I go?' and 'what path should I choose?' Nobody can answer those questions for me.

The process of completing the PhD took me through a series of emotions, not just the mental vigour to grasp what other people were doing in your field and to find out methodologies and approaches in trying to answer your questions. I will not forget incidences such as the Iraq war since March 2003, the Tsunami in December 2004 and recently the London bombings of 7th July 2005. I am writing them here because they have affected me in many ways and these events keep coming back to my mind when I sat writing this thesis.

So what will happen to me after the PhD? I will return to the university that sponsored me, Kolej Universiti Kejuruteraan Teknologi Malaysia (KUKTEM). Hopefully I will be able to pursue the research area of digital watermarking in medical images. I have already established contacts with radiologists from the Medical Faculty of the

International Islamic University Malaysia to become collaborators for the clinical evaluation of watermarked images. The hospital will be implementing teleradiology, so there will be medical images transferred across the network. The appropriate ethical approval will be sought from the university's ethic committee when I return to Malaysia.

7.6.2 My Conclusion on Security of Medical Images

A few people have asked me, "Why do you need to watermark medical images?" and a few others have asked me, "Who would want to forge medical images?". Here I will try to answer these two questions.

There is public concern regarding medical images being viewed and used by inappropriate parties, including relatives. This is of particular concern in telemedicine applications (Tachakra et al 1996) where images are shared outside a single organisation. Watermarking offers a method to embed patient details within an image, but in a way invisible to unauthorised persons. This may go some way to address these issues.

The approach taken in developing security techniques usually sees everybody as a potential criminal. This is really pathetic as the reasons behind it can be fictitious. This is to me like waiting in the battlefield waiting for an enemy that may never exist. To answer the second question, I could make a few fictitious criminals - the manipulation that can be achieved by adding or removing some parts of the image. The first person could be someone who wants to make false insurance claim by forging a medical image. The second person sells a forged medical image of a famous person to a tabloid newspaper. The third person is really vicious; he/she is trying to get away from his/her crime (homicide) by not just forging medical images, but the whole medical data to show that the death is through natural causes. But who gives them access to the data? An unauthorised person having access to the data, meaning that all security measures have failed. These include access to the building, to the room where the computer is located, the hospital network and the server where the data is stored. Planning to break

all of these security measures require a lot of resources. Maybe it is cheaper and easier to bribe a person who has access to the file than to break the code. I shall leave this to the scriptwriter to keep the suspense.

A technology solution to provide privacy, confidentiality and security of medical data is important. However, technology can do very little to ensure that the person receiving information will handle it according to standards. That depends on ethics and an effective supervision and legal structure that provides sanctions against detected misuse. As the demand for sophisticated IT in healthcare grew over US$25 billion in 2000 (Anderson 2000), technology must also be made comprehensible to the clinicians and medical personnel; otherwise they will resist it.

Glossary

Active attack Any attempt to thwart the purpose of a watermarking system by modifying content. This includes unauthorised removal and unauthorized embedding.

Adversary Anyone who attempts to thwart the purpose of a watermarking system. Depending on the application, adversaries might attempt a variety of attacks, including unauthorised removal, unauthorised detection and unauthorised embedding. Other terms from the literature that have been used for an adversary include pirate, hacker, attacker and traitor.

Asymmetric key watermarking Any method of watermarking in which embedding and detection require the use of different watermarking keys.

Authentication The process of verifying the integrity of a watermark or the watermarked image.

Blind Authentication Authentication without any knowledge of the original, unwatermarked content.

Cryptography The study and practice of keeping message secure.

Digital signature The digital equivalent of a traditional signature. They are used to verify the identity of the sender. A digital signature can be

constructed by encrypting a one-way hash of a message with the sender's private key.

Discrete Cosine Transform (DCT) A transform commonly used in image and video compression. The basic functions in this transform are real-valued cosine waves.

Error correction code (ECC) A mapping of messages into sequences of symbols such that not every possible sequence represents a message. In decoding such a code, sequences that do not correspond to messages are interpreted as corrupted code words. By defining the mapping between messages and code words in an appropriate way, it is possible to build decoders that can identify the code word closest to a given, corrupted sequence.

Exact authentication Verification that every bit of a given image has remained unchanged. This is in contrast to selective authentication.

False negative A type of error in which a detector fails to detect a watermark in a watermarked image.

False positive A type of error in which a detector incorrectly determines that a watermark is present in an image that was never watermarked.

Fragile watermark A watermark that becomes undetectable after even minor modifications of the image in which it is embedded. These are unsatisfactory for most applications, but can be useful for authentication.

Hash function A mapping of a variable length string into a fixed-length string called a hash. Typically, the hash of a string is shorter than the original.

Imperceptible Undetectable by a human perceptual system. This is often defined statistically.

Information hiding The art and science of hiding information. The fields of steganography and watermarking are examples of information hiding, but the term covers many other subjects, such as anonymous communications and preventing unauthorized database inference.

JPEG Joint Picture Experts Group- JPEG is a standard image compression technique based on block DCT quantization. JPEG2000 is a multi-scale wavelet-based image compression standard.

Key management Procedures for ensuring the integrity of keys used in cryptographic systems. This can include key generation, key distribution and key verification.

LSB watermarking The practice of embedding watermarks by placing information in the least significant bits of the image.

Message authentication Code (MAC) A one-way hash of a message that is then appended to the message. This is used to verify that the message is not altered between the time the hash is appended and the time it is tested.

One-way hash A hash function reasonably inexpensive to calculate, but prohibitively expensive to invert. That is, given an input string, it is easy to find the corresponding output. However, given a desired output, it is virtually impossible to find a corresponding input string.

Reversible watermark A watermark that can be exactly removed from the watermarked image, thereby obtaining a bit-for-bit copy of the original unwatermarked image. Such watermarks are more commonly referred to as invertible or erasable.

Robustness The ability of watermarks to survive signal processing operations.

Security In watermarking, the ability of a watermark to resist intentional tampering. More generally, the ability of an entire system (which may incorporate watermarking) to resist intentional tampering.

Semi-fragile watermark A watermark that is fragile against certain distortions but robust against others. This is useful for selective authentication.

Steganography The art of concealed communication by hiding messages in seemingly safe objects. The very existence of a steganographic message is secret. This term is derived from the Greek words *steganos*, which means covered, and *graphia*, which means writing.

Watermark A general term that can refer to an embedded message, a reference pattern, a message pattern or an added pattern.

Watermark key A secret key or key pair used for watermark embedding and detection. A watermark key can be used in conjunction with a cipher key.

Watermarking The practice of imperceptibly altering an image to embed a message about that image.

References

ACHARYA, R., ANAND, D., BHAT, S. and NIRANJAN, U.C., 2001. Compact storage of medical images with patient information, *IEEE Transactions Information Technology in Biomedicine*, **5**, pp. 320-323.

ANDERSON, J.G., 2000. Security of the distributed electronic patient record: a case-based approach to identifying policy issues, *International Journal of Medical Informatics*, **60**(2), pp. 111-118.

BARNI, M., BARTOLINI, F., CAPPELLINI, V., PIVA, A. and SALUCCO, F., 2001. Text-based geometric normalization for robust watermarking of digital maps, *IEEE International Conference on Image Processing (ICIP) 2001, Oct 7-10 2001 Thessaloniki,* IEEE Computer Society, pp.1082-1085.

BENEDENS, O. and BUSCH, C., 2000. Towards blind detection of robust watermarks in polygonal models. *Computer Graphics Forum,* **19**(3), pp. 199-208.

BHATTACHARJEE, S. and KUTTER, M., 1998. Compression tolerant image authentication, *IEEE International conference on image processing*, **1**, Chicago, IEEE, pp. 435-439.

BRASSIL, J.T., LOW, S. and MAXEMCHUK, N.F., 1999. Copyright protection for the electronic distribution of text documents. *Proceedings of the IEEE,* **87**(7), pp. 1181-1196.

CAO, F., HUANG, H.K. and ZHOU, X.Q., 2003. Medical image security in a HIPAA mandated PACS environment. *Computerized Medical Imaging and Graphics,* **27**(2-3), pp. 185-196.

CELIK, M.U., SHARMA, G., TEKALP, A.M., 2002. Hierarchical watermarking for secure image authentication with localization, *IEEE Transactions on Image Processing*, **11**(6), pp.585-594.

CHAO, H.M., HSU, C.M. and MIAOU, S.G., 2002. A data-hiding technique with authentication, integration, and confidentiality for electronic patients records, *IEEE Transactions Information Technology in Biomedicine*, **6**, pp. 46-53.

CHO, Y., AHN, B., KIM, J.S., KIM, I.Y. and KIM S.I., 2001. A study for watermark methods appropriate to medical images, *Journal of Digital Imaging*, **14**(2) supplement 1, pp.184-186.

CLUNIE, D.A., 2000. Lossless compression of grayscale medical images - effectiveness of traditional and state of the art approaches, *Medical Imaging 2000 - PACS Design and Evaluation: Engineering and Clinical Issues, Feb 15-Feb 17 2000 Bellingham, WA, USA,* Society of Photo-Optical Instrumentation Engineers pp.74-84.

COATRIEUX, G., MAITRE, H., SANKUR, B., ROLLAND, Y. and COLLOREC, R., 2000. Relevance of Watermarking in Medical Imaging, *2000 IEEE EMBS Conf. On Information Technology Applications in Biomedicine, November 2000 Arlington, USA*, IEEE, pp. 250-255.

COX, I.J., MILLER, M.L. and BLOOM, J.A., 2002. *Digital Watermarking*. San Francisco, CA: Morgan Kaufmann Publishers.

DIFFIE, W. and HELLMAN, M.E., 1976. New directions in cryptography. *IEEE Transactions on Information Theory,* **22**(6), pp. 644-654.

DIGIMARK, 2001. Press Releases: Philips Digital Networks Licenses Digimarc Digital Watermark Patents for Video Broadcast Monitoring [Homepage of Digimarc Corporation], [Online]. Available: http://www.digimarc.com/about/release.asp?newsID=59 [October 24, 2005].

DOBBERTIN, H., 1996. The Status of MD5 After a Recent Attack. *Crypto Bytes*, **2**(2), pp. 1 and 3, available : ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto2n2.pdf [June 6, 2005].

DRM WATCH STAFF, 2004. Philips Releases Turnkey System for DVD Watermarking [Homepage of DRM], [Online]. Available: http://www.drmwatch.com/drmtech/article.php/3427971 [October 24, 2005].

EGGERS, J.J., IHLENFELDT, W. and GIROD, B., 2001. Digital watermarking of chemical structure sets, *Proceedings of the 4th Information Hiding Workshop '01, 25-27 April 2001 Pittsburgh, PA, USA*.

FEDERATION BUREAU OF INVESTIGATION (FBI), October 2000, 2000-last update, forensic audio, video & image analysis unit [Homepage of FBI], [Online]. Available: http://www.fbi.gov/hq/lab/org/faviau.htm [May 6, 2005].

FRIDRICH, J., GOLJAN, M. and BALDOZA, A.C., 2000. New fragile authentication watermark for images, *International Conference on Image Processing (ICIP 2000), Sep 10-13 2000 Vancouver, BC,* IEEE Computer Society pp. 446-449.

FRIDRICH, J., 1998. Image watermarking for tamper detection, *Proceedings of the 1998 International Conference on Image Processing, ICIP. Part 2 (of 3), Oct 4-7 1998 Los Alamitos, CA, USA,* IEEE Comp Society, pp. 404-408.

FRIDRICH, J. and GOLJAN, M., 1999. Images with self-correcting capabilities. *IEEE International Conference on Image Processing,* **3**, pp. 792-796.

FRIEDMAN, G.L., 1993. The Trustworthy Digital Camera: Restoring Credibility to the Photographic image. *IEEE Transactions on Consumer Electronics,* **39**(4), pp. 905-910.

GARFINKEL, S. and SPAFFORD, G., 1996. *Practical Unix and Internet Security*. Devon, UK: O'Reilly & Associates.

GIAKOUMAKI, A., PAVLOPOULOS, S. and KOUTSOURIS, D., 2003. A medical image watermarking scheme based on wavelet transform, *Engineering in Medicine and*

Biology Society. Proceedings of the 25th Annual International Conference of the IEEE, Sep 17-21 2003 Cancun, Mexico, IEEE, **1**, pp. 856-859.

GIROD, B., 1993. What's Wrong with Mean-squared Error? In: Watson, A. B., ed. *Digital Images and Human Vision*, Cambridge, MA:MIT Press, pp. 207-220.

GOLJAN, M., FRIDRICH, J. and DU, R., 2001. Distortion-free data embedding for images, *Proceedings of 4th International Workshop on Information Hiding, Lecture Notes in Computer Science*, **2137**, pp. 27-41.

GREEN, D. M. and SWETS, J.A. 1974. *Signal Detection Theory and Psychophysics*, Huntington, New York: Robert E. Krieger Publishing Co.

GUO, X. and ZHUANG, T., 2003. A lossless watermarking scheme for enhancing security of medical data in PACS, *Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation, Feb 18-20 2003 San Diego, CA, USA,* The International Society for Optical Engineering, pp.350-359.

HAWKES, P., PADDON, M. and ROSE, G.G., 2005, Musings on the Wang et al. MD5 collision [Homepage of International Association of Cryptologic Research (IACR)], [Online]. Available: http://eprint.iacr.org/2004/264.pdf [May 10, 2005].

HERNANDEZ, J.R., AMADO, M. and PEREZ-GONZALEZ, F., 2000. DCT-domain watermarking techniques for still images: detector performance analysis and a new structure. *IEEE Transactions on Image Processing,* **9**(1), pp. 55-68.

HOLLIMAN, M. and MEMON, N., 2000. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Transactions on Image Processing,* **9**(3), pp. 432-41.

HONSINGER, C.W., JONES, P.W., RABBANI, M. and STOFFEL, J.C., 2001. Lossless recovery of an original image containing embedded data. Patent No. US06278791, United States.

HORII, S.C., 1997-last update, A nontechnical introduction to DICOM [Homepage of Radiological Society of North America], [Online]. Available: http://www.rsna.org/REG/practiceres/dicom/nontechintro.html [07/08, 2003].

HUANG, H.K., 2003. Enterprise PACS and image distribution. *Computerized Medical Imaging and Graphics,* **27**(2-3), pp. 241-53.

ITU, 2000. *Methodology for the Subjective Assessment of the Quality of Television Pictures: Recommendation ITU-R BT.500-10*, Radiocommunication Assembly.

JOHNSON, N.F. and KATZENBEISSER, S.C., 2000. A Survey of Steganographic Techniques, In: S. KATZENBEISSER and PETITCOLAS, FABIEN A. P., eds, *Information Hiding Techniques for Steganography and Digital Watermarking.* Norwood, MA: Artech House, pp. 43-75.

KUNDUR, D. and HATZINAKOS, D., 1998. Towards a telltale watermarking technique for tamper-proofing, *Proceedings of IPCIP'98 International Conference on Image Processing, 4-7 Oct. 1998 Chicago, IL, USA,* IEEE Computer Society, pp. 409-13.

KUNDUR, D. and HATZINAKOS, D., 1996. Blind image deconvolution. *IEEE Signal Processing Magazine,* **13**(3), pp. 43-64.

KUTTER, M. and HARTUNG, F., 2000. Introduction to watermarking techniques, In: S. KATZENBEISSER and PETITCOLAS, FABIEN A. P., eds., *Information Hiding Techniques for Steganography and Digital Watermarking.* Norwood, MA: Artech House, pp. 97-120.

KWON, K., KWON, S., LEE, S., KIM, T. and LEE, K., 2003. Watermarking for 3D polygonal meshes using normal vector distributions of each patch, *Proceedings: International Conference on Image Processing, ICIP-2003, Sep 14-17 2003 Barcelona, Spain,* IEEE Computer Society, pp. 499-502.

LANGELAAR, G.C., SETYAWAN, I. and LAGENDIJK, R.L., 2000. Watermarking digital image and video data. *IEEE Signal Processing Magazine,* **17**(5), pp. 20-46.

LEE, J. and CHEE SUN WON, 2000. Image integrity and correction using parities of error control coding, *IEEE International Conference on Multimedia and Expo (ICME 2000), Jul 30-Aug 2 2000 New York, NY, USA,* IEEE, pp. 1297-1300.

LEE, J. and WON, C.S., 1999. Authentication and correction of digital watermarking images. *Electronics Letters,* **35**(11), pp. 886-887.

LEE, W. and CHEN, T., 2002. A public verifiable copy protection technique for still images. *Journal of Systems and Software,* **62**(3), pp. 195-204.

LEMPEL, A., ZIV, J., 1986, Compression of two-dimensional data, *IEEE Transactions on Information Theory*, **32**(1), pp. 2-8.

LI, C., LOU, D. and LIU, J., 2003. Image integrity and authenticity verification via content-based watermarks and a public key cryptosystem. *Journal of the Chinese Institute of Electrical Engineering, Transactions of the Chinese Institute of Engineers, Series E/Chung KuoTien Chi Kung Chieng Hsueh K'an,* **10**(1), pp. 99-106.

LI, W. and XUE, X., 2003. An audio watermarking technique that is robust against random cropping. *Computer Music Journal,* **27**(4), pp. 58-68.

LIN, C.-. and CHANG, S.-., 2001. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology,* **11**(2), pp. 153-168.

LIN, C. and CHANG, S., 2000. Semi-fragile watermarking for authenticating JPEG visual content, *Security and Watermarking of Multimedia Contents II, Jan 24-Jan 26 2000 Bellingham, WA, USA,* Society of Photo-Optical Instrumentation Engineers, pp. 140-151.

LOU, D. C. and LIU, J. L., 2000. Fault resilient and compression tolerant digital signature for image authentication. *IEEE Transactions on Consumer Electronics*, **46**(1), pp. 31-39.

MACQ, B. and DEWEY, F., 1999. Trusted Headers for Medical Images, *DFG VIIII-DII Watermarking Workshop,* Erlangen, Germany.

MEMON, N., SHENDE, S. and WONG, P., 1999. On the security of the Yueng-Mintzer Authentication Watermark, *Final Program and Proceedings of the IS&T PICS 99, April 1999 Savannah, GA, USA*, The Society for Imaging Science and Technology, pp. 301-306.

MENEZES, A., OORCHOT P. VAN and VANSTONE, S., 1997. *Handbook of Applied Cryptography*. Boca Raton, FL:CRC.

MINTZER, F., BRAUDAWAY, G.W. and BELL, A.E., 1998. Opportunities for Watermarking Standards. *Communications of the ACM,* **41**(7), pp. 55-64.

MINTZER, F., BRAUDAWAY, G.W. and YEUNG, M.M., 1997. Effective and ineffective digital watermarks, *Proceedings of the 1997 International Conference on Image Processing. Part 3 (of 3), Oct 26-29 1997 Los Alamitos, CA, USA,* IEEE Comp Soc, pp. 9-12.

MONDEN, A., IIDA, H., MATSUMOTO, K., INOUE, K. and TORII, K., 2000. Practical method for watermarking Java programs, *2000 IEEE 24th Annual International Computer Software and Applications Conference (COMPSAC 2000), Oct 25-Oct 27 2000 Los Alamitos, CA, USA,* IEEE Computer Society, pp.191-197.

MURRAY, T.D., 1996-last update, the wizard of watermarks [Homepage of Virginia Tech.], [Online]. Available: http://ebbs.english.vt.edu/gravell/wizard/wizard.html [June 30, 2004].

MUSICTRACE, 2005. Watermark embedding for audio signals [Homepage of MusicTrace], [online]. Available: http://www.musictrace.de/products/contentmark.en.htm [October 24, 2005].

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION, 2003. *Digital Imaging and Communications in Medicine (DICOM).* PS 3.1-2003, available:

http://www.amicas.com/pacsed/DICOM%20Strategy_2002-03-28.doc [6 July 2004, 2004].

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 1995. *Secure Hash Standard (SHA-1).* Federal Information Processing Standards Publication #180-1, available: http://www.itl.nist.gov/fipspubs/fip180-1.htm [June 30, 2004].

OKADA, H., SHIITEV, A., SONG, H., FUJITA, G., ONOYE, T. and SHIRAKAWA, I., 2002. Error detection by digital watermaking for MPEG-4 video coding. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* **E85-A**(6), pp. 1281-1288.

O'RUANAIDH, J.J.K. and PUN, T., 1997. Rotation, scale and translation invariant digital image watermarking, *Proceedings of IEEE International Conference on Image Processing*, *October 1997 Santa Barbara, CA, USA*, IEEE, (**1**), pp. 536-539.

PAQUET, A.H., WARD, R.K. and PITAS, I., 2003. Wavelet packets-based digital watermarking for image verification and authentication. *Signal Processing,* **83**(10), pp. 2117-2132.

PETITCOLAS, FABIEN A. P., 2000. Introduction to Information Hiding. In: S. KATZENBEISSER and PETITCOLAS, FABIEN A. P., eds., *Information Hiding Techniques for Steganography and Digital Watermarking.* Norwood, MA: Artech House, pp. 1-11.

RIVEST, R.L., April 1992. *The MD5 Message Digest Algorithm.* Internet Request For Comments: MIT Laboratory for Computer Science and RSA Data Security, Inc.

RIVEST, R.L., SHAMIR, A. and ADLEMAN, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM,* **21**(2), pp. 120-126.

SAID, A. and PEARLMAN, W.A., 1996. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology,* **6**(3), pp. 243-50.

SCHNEIDER, M. and CHANG, S.F., 1996. Robust content based digital signature for image authentication, *Proceedings of IEEE International Conference on Image Processing*, *September 16-19 1996 Lausanne, Switzerland*, IEEE, **3**, pp. 227-230.

SHANNON, C.E., 1948. A Mathematical Theory of Communication, *Bell System Technical Journal*, **27**(4), 373-423, 623-656.

STALLINGS, W., 2003. *Network Security Essentials Applications and Standards*. Second ed. Upper Saddle River, New Jersey: Prentice Hall.

STINSON, D.R., 1995. *Cryptography: Theory and Practice*. Boca Raton, FL: CRC Press.

SU, J.K., HARTUNG, F. and GIROD, B., 1998. Digital watermarking of text, image, and video documents. *Computers & Graphics,* **22**(6), pp. 687-695.

SUN, S. and CHANG, P., 2003. Video watermarking synchronization based on profile statistics, *Proceedings: 37th Annual 2003 International Carnahan Conference on Security Technology, Oct 14-16 2003 Taipei, Taiwan,* IEEE, pp. 410-413.

TACHAKRA, S., MULLETT S.T.H., FREIJ, R. and SIVAKUMAR, A., 1996. Confidentiality and ethics in telemedicine, *Journal of Telemedicine and Telecare*, **2**(1), pp. 68-71.

TONG, LIU and ZHENG-DING, QIU, 2002. The survey of digital watermarking-based image authentication techniques, *Proceedings of International Conference on Signal Processing (ICSP), 26-30 Aug. 2002 Beijing, China,* IEEE, pp. 1556-1559.

TRICHILI, H., BOUHLEL, M., DERBEL, N. and KAMOUN, L., 2002. A new medical image watermarking scheme for a better telediagnosis, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Oct 6-9 2002 Yasmine Hammamet, Tunisia*, IEEE, pp. 557-560.

TSAI, P., HU, Y. and CHANG, C., 2004. A color image watermarking scheme based on color quantization. *Signal Processing,* **84**(1), pp. 95-106.

UTKU CELIK, M., SHARMA, G., SABER, E. and MURAT TEKALP, A., 2002. Hierarchical watermarking for secure image authentication with localization. *IEEE Transactions on Image Processing,* **11**(6), pp. 585-95.

VAN SCHYNDEL, R.G., TIRKEL, A.Z. and OSBORNE, C.F., 1994. A digital watermark, *Proceedings of 1st International Conference on Image Processing, 13-16 Nov 1994 Austin, TX, USA,* IEEE Computer Society Press, pp. 86-90.

VOYATZIS, G. and PITAS, I., 1996a. Applications of toral automorphisms in image watermaking, *Proceedings of the 1996 IEEE International Conference on Image Processing, ICIP'96. Part 2 (of 3), Sep 16-19 1996 Los Alamitos, CA, USA,* IEEE, pp. 237-240.

VOYATZIS, G. and PITAS, I., 1996b. Chaotic mixing of digital images and applications to watermarking, *Proceedings of European Conference on Multimedia Applications, Services and Techniques, 28-30 May 1996 Louvain la Neuve, Belgium,* Univ. Catholique Louvain, pp. 687-94.

WAKATANI, A., 2002. Digital Watermarking for ROI Medical Images by Using Compressed Signature Image, *35th Annual Hawaii International Conference on System Sciences (HICSS-35'02), 7-10 Jan 2002 Big Island, Hawaii*, IEEE, pp. 2043-2048.

WALLACE, G.K., 1991. The JPEG Still Picture Compression Standard. *Communications of the ACM,* **34**(4), pp. 30-44.

WALTON, S., 1995. Information authentication for a slippery new age. *Dr. Dobbs Journal,* **20**(4), pp. 18-26.

WANG, X., FENG, D., LAI, X. and YU, H., August 2004, 2004-last update, collisions for hash functions MD4, MD5, HAVAL-128, and RIPEMD. Available: http://eprint.iacr.org/2004/199 [May 10, 2005].

WOLFGANG, R.B. and DELP, E.J., 1996. A watermark for digital images, *Proceedings of International Conference on Image Processing, ICIP'96, Sep 16-19 1996 Los Alamitos, CA, USA, IEEE*, **3**, pp. 219-222.

WOLFGANG, R.B., PODILCHUK, C.I. and DELP, E.J., 1999. Perceptual watermarks for digital images and video. *Proceedings of the SPIE - The International Society for Optical Engineering, Security and Watermarking of Multimedia Contents, 25-27 Jan. 1999 San Jose, CA, USA*, Society of Photo-Optical Instrumentation Engineers, **3657**, pp. 40-51.

WOLFGANG, R.B. and DELP, E.J., 1999. Fragile watermarking using the VW2D watermark. *Proceedings of SPIE - The International Society for Optical Engineering, Security and Watermarking of Multimedia Contents, 25-27 Jan. 1999 San Jose, CA, USA***,** Society of Photo-Optical Instrumentation Engineers**, 3657**, pp. 204-213.

WONG, P., 1999. A watermark for image integrity and ownership verification, *Final Program and Proceedings of the IS&T PICS 99, April 1999 Savannah, GA, USA*, The Society for Imaging Science and Technology, pp.374-379.

WONG, P.W., 1998. Public key watermark for image verification and authentication, *Proceedings of the 1998 International Conference on Image Processing, ICIP. Part 1 (of 3), Oct 4-7 1998 Los Alamitos, CA, USA,* IEEE Computer Society, pp. 455-459.

WONG, P.W. and MEMON, N., 2000. Secret and public key authentication watermarking schemes that resist vector quantization attack, *Security and Watermarking of Multimedia Contents II, Jan 24-Jan 26 2000 Bellingham, WA, USA,* Society of Photo-Optical Instrumentation Engineers, pp.417-427.

WONG, S., ZAREMBA, D., GOODEN, D. and HUANG, H.K., 1995. Radiologic image compression-a review*, Proceedings of IEEE*, **83(2)**, pp. 194-219.

WU, M. and LIU, B., 1998. Watermarking for image authentication, *Proceedings of the 1998 International Conference on Image Processing, ICIP. Part 2 (of 3), Oct 4-7 1998 Los Alamitos, CA, USA*, IEEE Computer Society, pp. 437-441.

XIE, L. and ARCE, G.R., 1998. Joint wavelet compression and authentication watermarking, *Proceedings of the 1998 International Conference on Image Processing, ICIP. Part 2 (of 3), Oct 4-7 1998 Los Alamitos, CA, USA,* IEEE Computer Society, pp. 427-431.

YAN, F., JI, B., ZHANG, D. and FANG, H., 2004. Robust quadri-phase audio watermarking. *Acoustical Science and Technology,* **25**(1), pp. 106-108.

YANG, Y. and BAO, F., 2003. An invertible watermarking scheme for authentication of electronic clinical brain atlas, *IEEE International Conference on Accoustics, Speech, and Signal Processing, Apr 6-10 2003 Hong Kong,* IEEE, pp.533-536.

YEUNG, M.M. and MINTZER, F., 1997. An invisible watermarking technique for image verification, *Proceedings of International Conference on Image Processing, Oct 1997 Santa Barbara, CA, USA*, IEEE, **2**, pp. 680-683.

YEUNG, M.M. and MINTZER, F.C., 1998. Invisible watermarking for image verification. *Journal of Electronic Imaging,* **7**(3), pp. 578-591.

ZHANG, X., FENG, J. and LO, K., 2003. Image watermarking using tree-based spatial-frequency feature of wavelet transform. *Journal of Visual Communication and Image Representation,* **14**(4), pp. 474-491.

ZHOU, X.Q., HUANG, H.K. and LOU, S.L., 2001. Authenticity and integrity of digital mammography images. *IEEE Transactions on Medical Imaging,* **20**(8), pp. 784-791.

ZHU, B., SWANSON, M.D. and TEWFIK, A.H., 1996. Transparent robust authentication and distortion measurement technique for images, *7th IEEE Digital Signal Processing Workshop Proceedings*, *Sep 1-4 1996 Loen, Norway*, IEEE, pp.45-48.

Appendices

Appendix A – Clinical Assessment of Ultrasound Images

Clinical assessment

Digital watermarking introduces identifiers to guard against false ownership claims and fabrication. Through visual inspection by the naked eye, these images appear to be unaltered and therefore medical diagnoses should not be any different regardless of the presence or absence of the identifiers. Technically speaking, the image pixels are preserved despite the introduction of the identifiers and therefore the assumption that the clinical diagnoses remain unchanged is technically sound. However, clinical assessment of these images would in some way add further evidence to the conclusions reached so far, but more importantly, such a study would reduce anxieties and fears that may arise among clinicians as the result of this technique.

Methodology and Statistical Analysis

This study involves subjecting assessors to two sets of images, the original (group O) and those digitally watermarked (group DW). Both groups view ultrasound images that essentially are similar, except the latter has been digitally watermarked with this new technique. Group 0 would be regarded as a control against for group DW to be compared against. This study is conducted in a blind manner in that the assessors do not know which of the images to be assessed has been watermarked. The assessors will consist of four radiologists at consultant level to achieve authority and consistency in assessment, who would therefore be familiar with ultrasound images used in clinical practise.

Fifteen images are used as controls in group O. The same images are digitally watermarked to represent group DW. All images (O and DW) would be randomly

assigned to the assessors who are blind to the group the image belongs to. Each image will carry a clinical stem to prompt clinical diagnosis. In cases where a clinical diagnosis is not possible, reasons for this are sought, but which could be owing to poor image quality. The assessors are also given the opportunity to add comments on all aspects of the image being assessed should they need to do so.

All images assessed would be re grouped into groups O and DW and a Chi Square test is employed to detect any significant difference between them. A value $P<0.05$ is taken as the level of significance. Further analysis would also be carried out on images (if incorrectly diagnosed) to explain this finding.

The study would also look into comments made by assessors in all aspects of the images, as these comments would also form the basis of any conclusion formed from this clinical assessment.

List of images for evaluation:

1. Abdominal aortic aneurysm
2. Adrenal mass
3. Liver cysts
4. Liver metastases
5. Cholecystitis / cholelithiasis
6. Achilles tendon tear
7. Forearm abscess
8. Muscle mass
9. Patellar tendon tear
10. Rotator cuff tear
11. Breast cyst
12. Abnormal endometrium
13. Adnexal mass
14. Ovarian cyst
15. Greater saphenous vein thrombosis.

CLINICAL ASSESMENT OF ULTRASOUND IMAGES

Investigator: Jasni M Zain
Brunel University

Thank you for agreeing to participate in this study.

The aim is to find out whether or not embedded ultrasound images alter clinical diagnosis when compared to the original ones.

In this study, ultrasound images underwent a process called watermarking, mainly to add security during image transfer. By visual inspection, the images do not change as the result of the process; this study will go one step further by subjecting these images to objective clinical assessment.

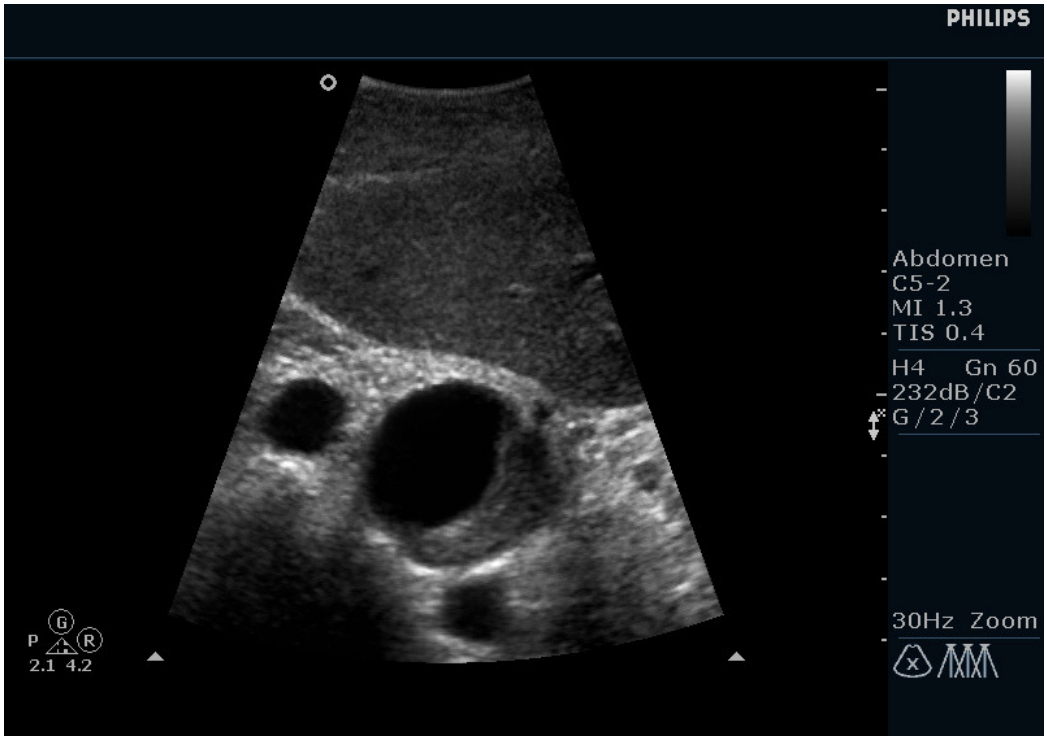
You will be given 10 ultrasound images, a mixture of the original and the embedded, each with a brief clinical summary to help you arrive at the most likely diagnosis. You will not be able to differentiate whether or not these images have been embedded.

Your task is to enter the most likely diagnosis based on the clinical summary and the ultrasound image provided. Please do not write a descriptive report. If you cannot arrive at a single diagnosis, please choose a box to state the reason.

Thank you for your kind help.

Clinical summaries and the ultrasound images

1. Abdominal ultrasound of a 72-year-old man with chronic abdominal discomfort.

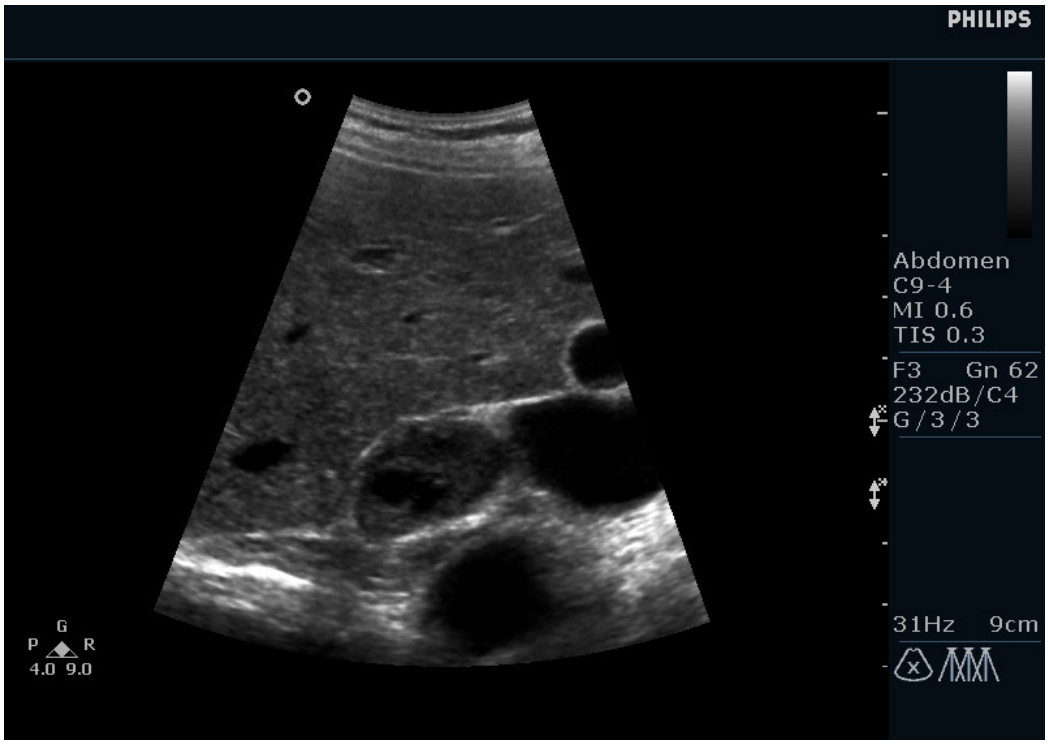


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

2. A screening abdominal ultrasound of a woman with confirmed lung malignancy showing the left adrenal.

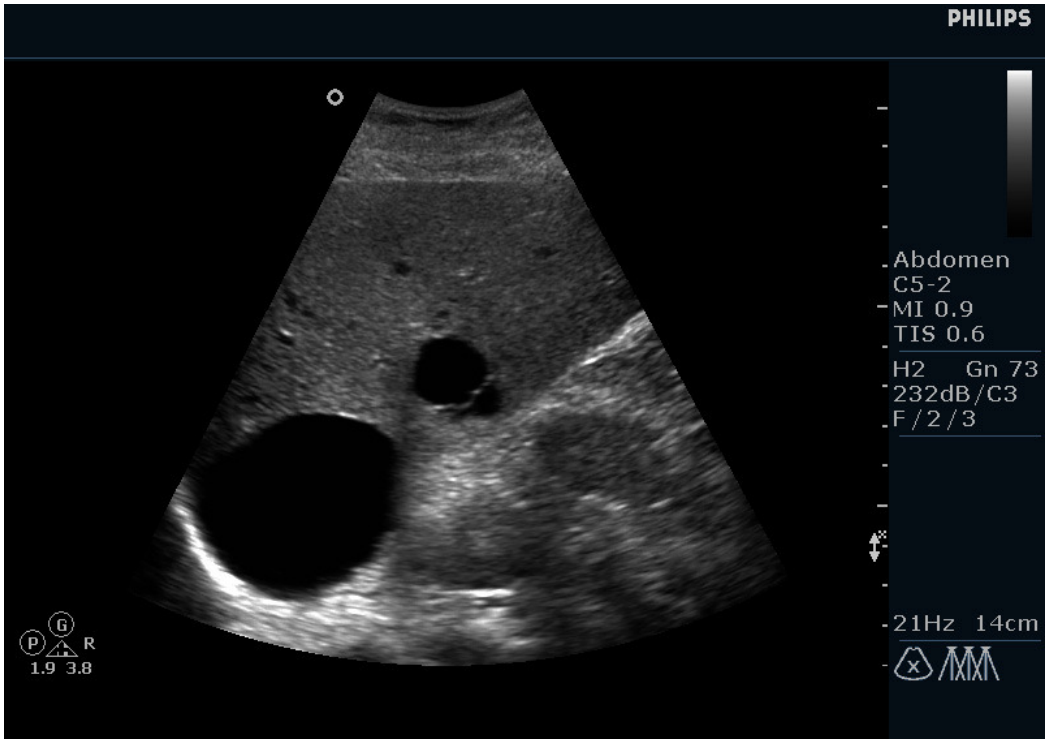


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

3. A young woman with one-day history of right upper quadrant pain that resolved on arrival to the casualty department. The liver functions tests and total white cell count were all normal. This is an ultrasound image of the liver.

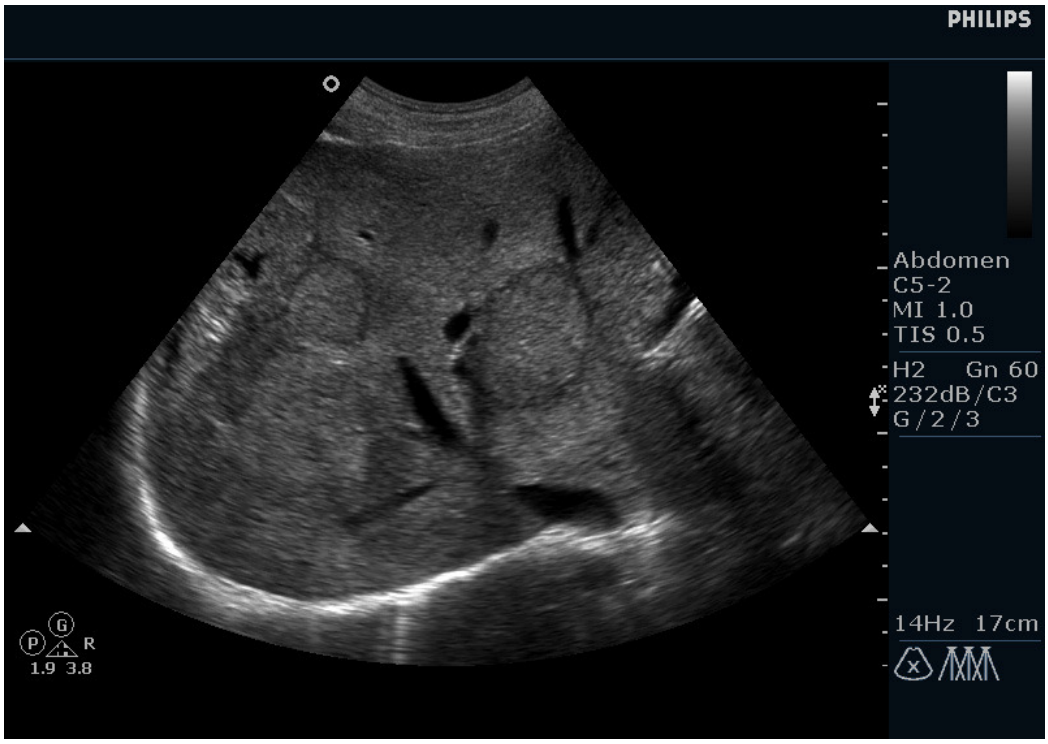


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

4. A liver ultrasound of a man with adenocarcinoma of her left lung.

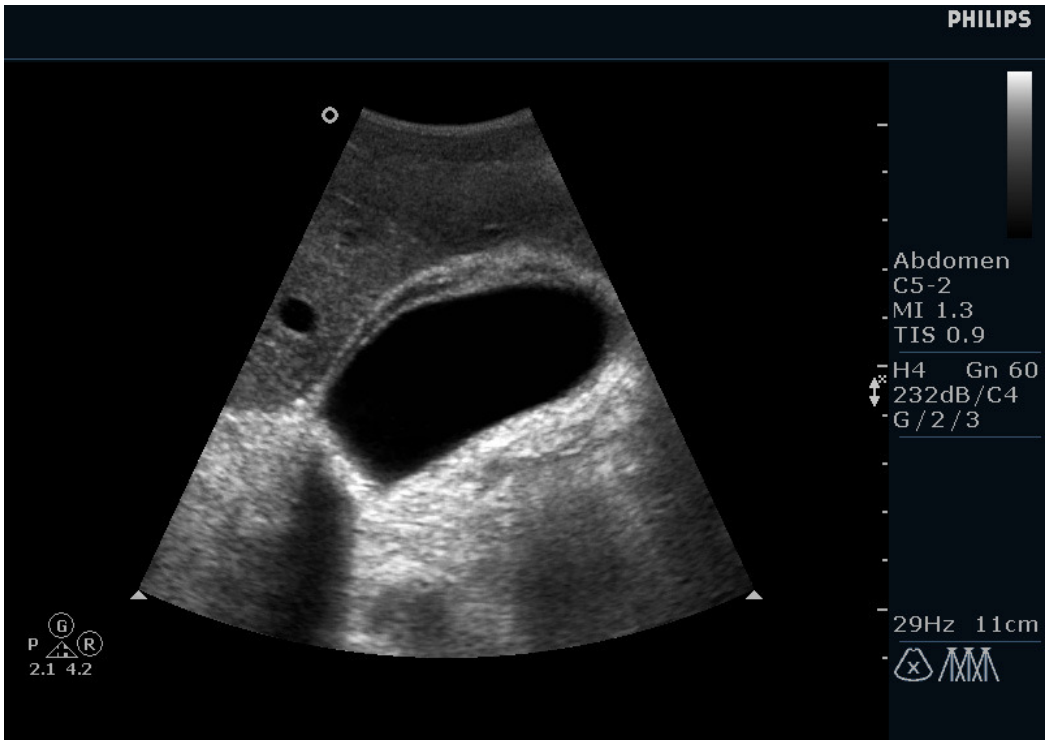


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

5. An obese female patient with 2 days history of fever and constant right upper quadrant pain.



The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

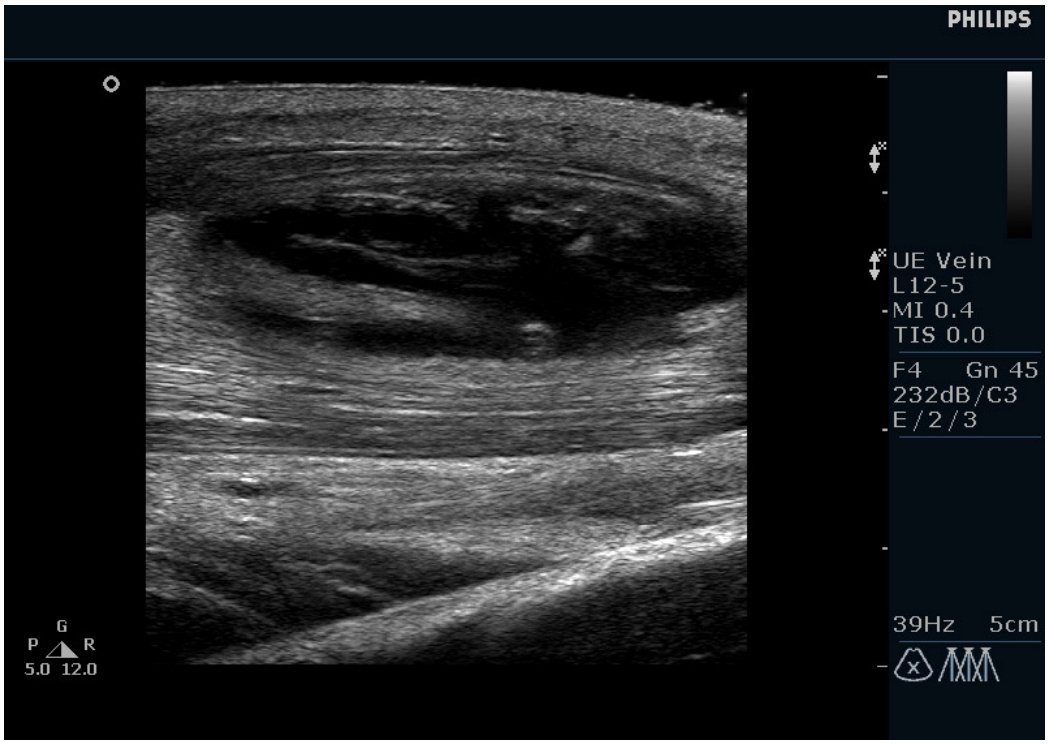
6. A left heel ultrasound of an amateur rugby player who could not walk following a tackle.

The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

7. A poorly controlled diabetic man with a swollen and tender forearm. Blood tests showed elevated white cells with neutrophils predominance. This is the ultrasound of his forearm.

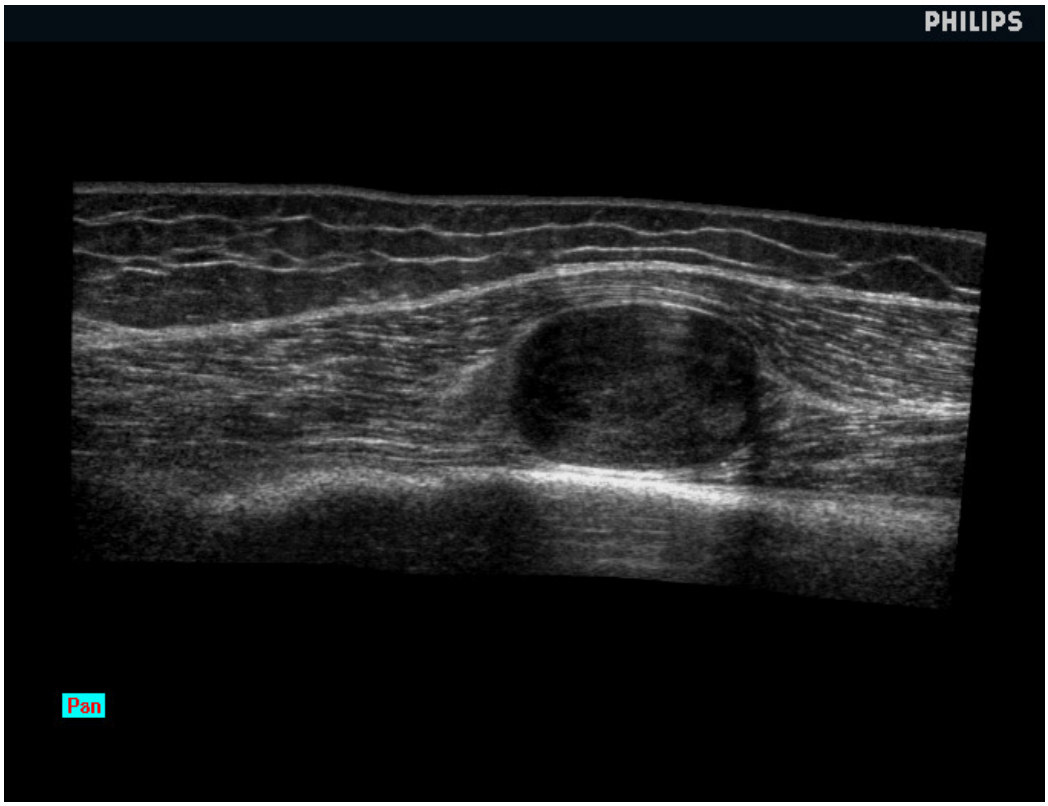


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

8. A middle-aged man with recent history of thigh swelling. This image was from the swollen area.

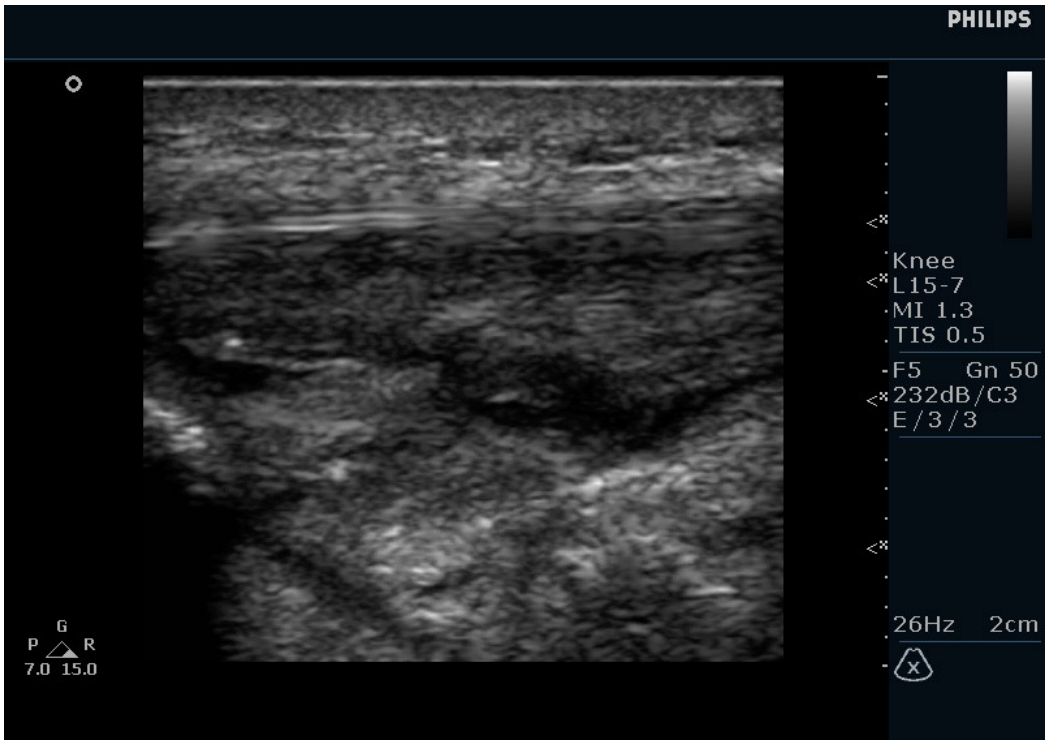


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

9. A knee ultrasound of a man with a tender kneecap following a football game.

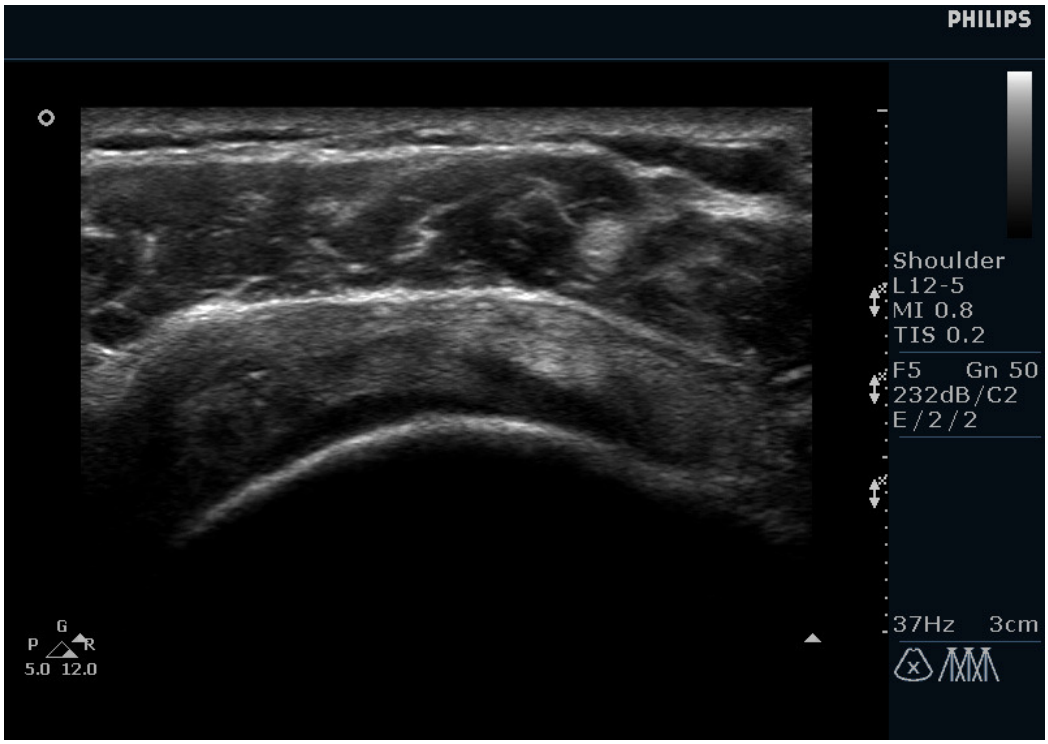


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

10. A shoulder ultrasound done on a javelin thrower with severe shoulder pain following a big javelin throw.

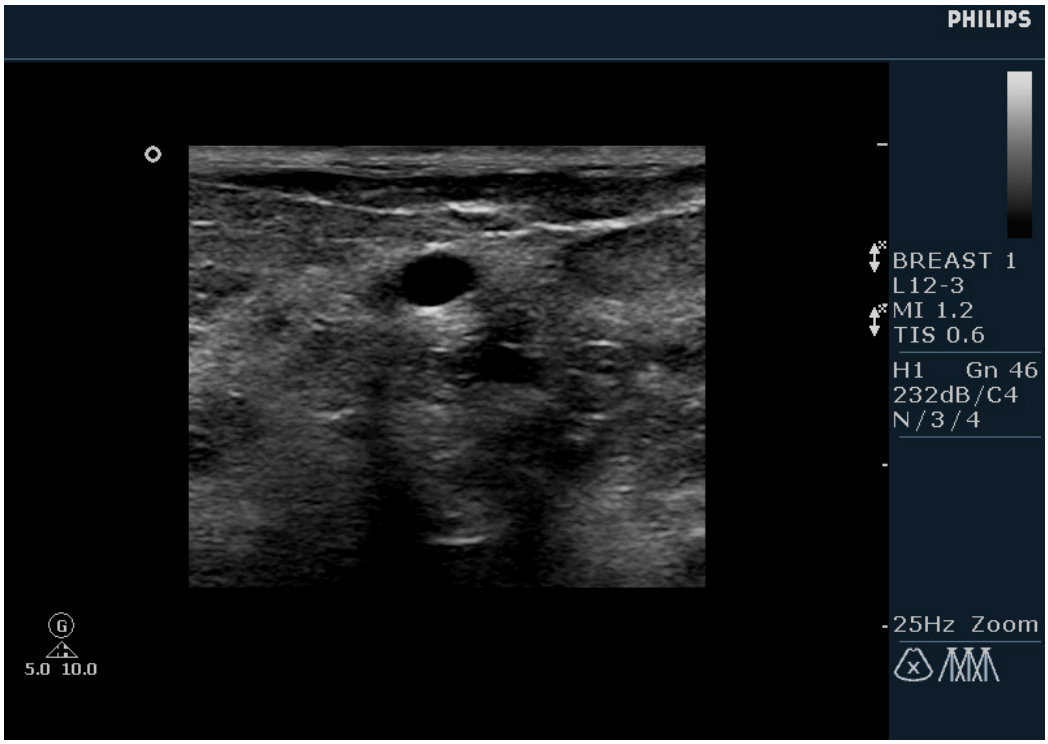


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

11. A breast ultrasound of an asymptomatic woman.

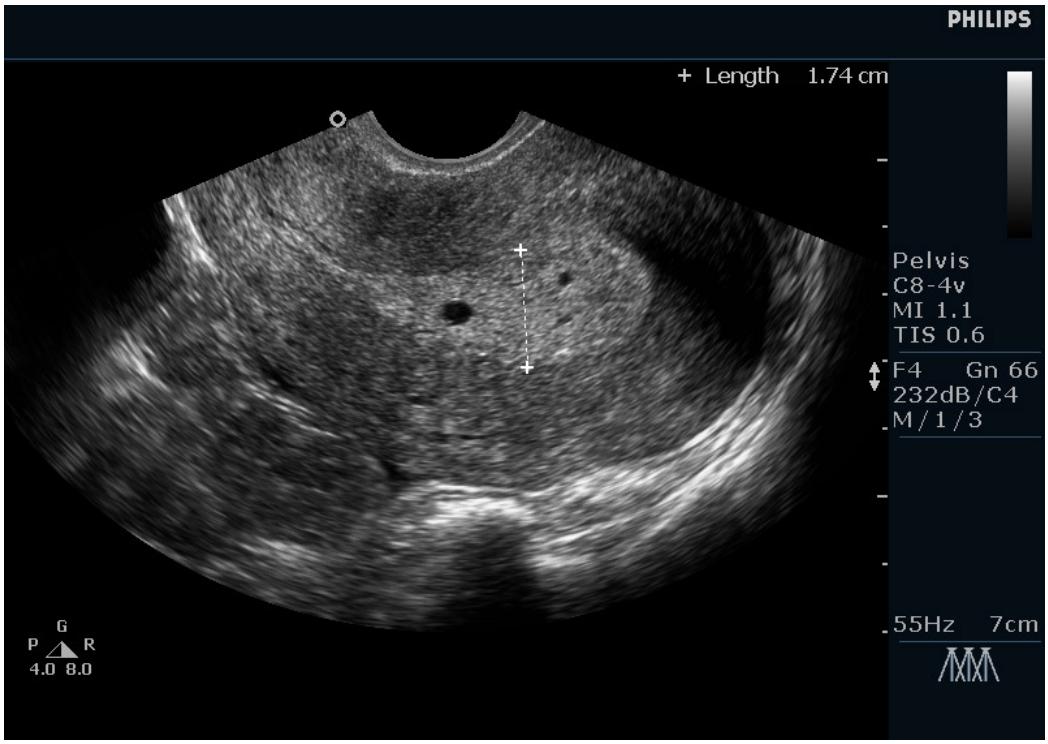


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

12. A lady with frequent heavy periods underwent a pelvic ultrasound.

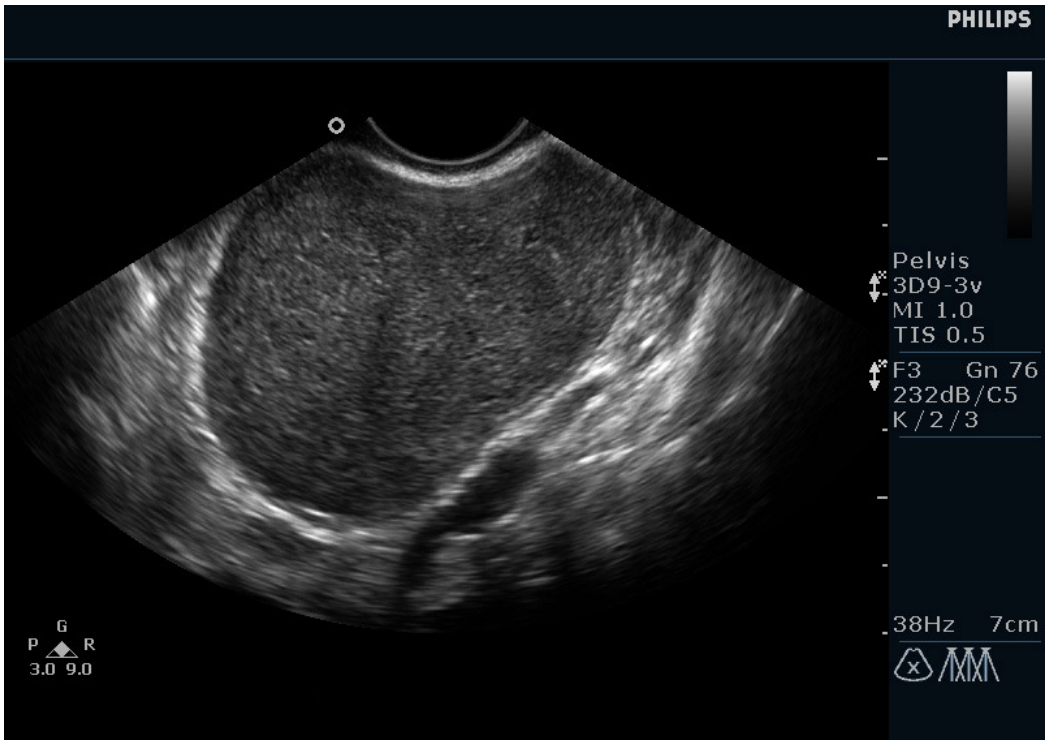


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

13. An adnexal image from a pelvic ultrasound of an elderly lady with malignant ascites.

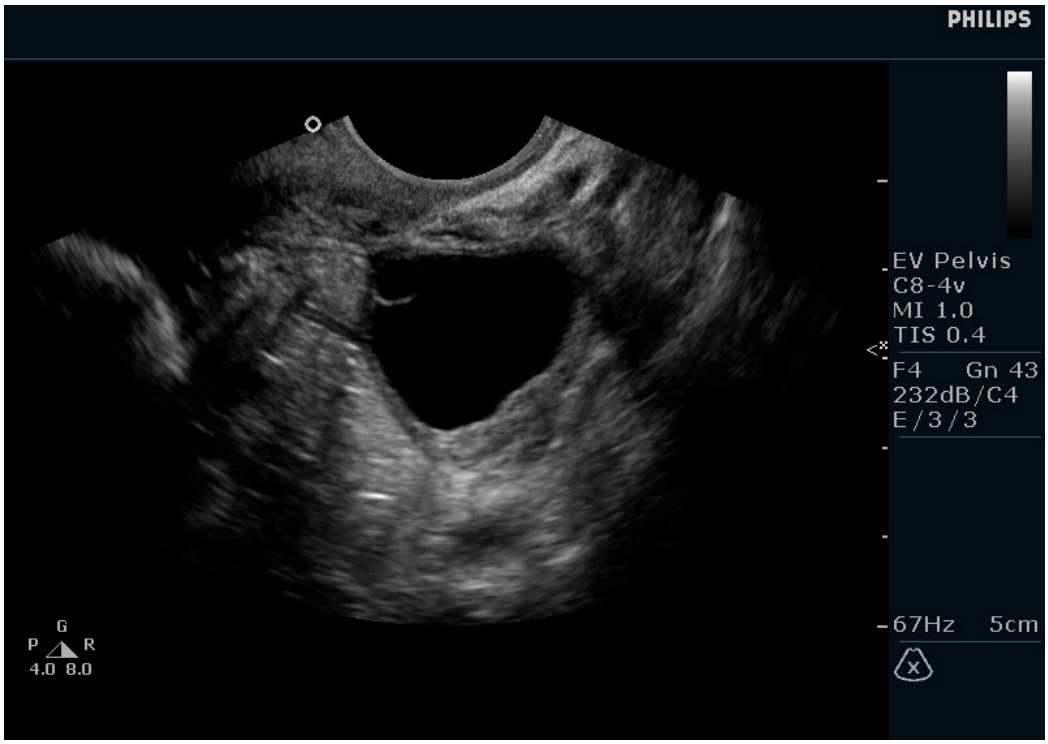


The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

14. A young woman with an intermittent left iliac fossa pain. This image is from her left ovary.



The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

15. A right inner thigh ultrasound of an obese woman with chronic varicose veins and previous history of right deep venous thrombosis. She presented with an acutely swollen right leg.

The most likely diagnosis is

Please choose the appropriate box if no diagnosis is entered

Don't know	
Inadequate clinical information	
Poor image quality	
Two or more diagnoses are equally likely	
Others	

Appendix B – Program Listing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Function:     Calculates the PSNR (Peak Signal to Noise Ratio)
%                   of images A and A', both of size MxN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function A = psnr(image,image_prime,M,N)

   % convert to doubles
   image=double([image]);
   image_prime=double([image_prime]);

   % avoid divide by zero nastiness
   if (sum(sum(image-image_prime)) == 0)
      error('Input vectors must not be identical')
   else
      psnr_num=M*N*max(max(image.^2));              % calculate numerator
      psnr_den=sum(sum(image-image_prime).^2);      % calculate denominator
      A=psnr_num/psnr_den;                          % calculate PSNR
   end

return
```

```
  function [PSNR,mse]=psnr(X,Y)
% function [PSNR,mse]=psnr(X,Y)
% Peak signal to noise ratio of the difference between images and the
%mean square error
%    If the second input Y is missing then the PSNR and MSE of X itself
% becomes the output (as if Y=0).

if nargin<2, D=X;
else
  if any(size(X)~=size(Y)), error('The input size is not equal to each other!'); end
  D=X-Y;
end

mse=sum(D(:).*D(:))/prod(size(X))
PSNR=10*log10(255^2/mse)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:              Jasni Zain
% Project:          Strict Authentication Watermarking
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hare = dicomread ('JZI944SA.DCM');
info= dicominfo ('JZI944SA.DCM');
%F15 = imread('nhs.jpg', 'jpeg');
%n =4; % Number of bits to replace 1 <= n <= 7
%Hare= rgb2gray(Hare);
Hare = double(Hare);
%a=mat2str(Hare);
%z= md5(a)
Hare1 = Hare;
for i = 241:248
   for j =9:16
      for b=2:-1:1

         Hare1(i,j)= bitset(Hare1(i,j),b,1);

      end
   end
end
for i = 241:248
   for j =17:24
      for b=2:-1:1

         Hare1(i,j)= bitset(Hare1(i,j),b,1);

      end
   end
end
for i = 241:248
   for j =25:32
      for b=2:-1:1

         Hare1(i,j)= bitset(Hare1(i,j),b,0);
         %k=k+1;
      end
   end
end
for i = 241:248
   for j =33:40
      for b=2:-1:1

         Hare1(i,j)= bitset(Hare1(i,j),b,0);
         %k=k+1;
```

```
      end
   end
end

a=Hare1(241:248,9:40)
%Stego = uint8(double(RemoveLSB(Hare, n)) + double(F15) / 2^(8 - n));

%Extracted = uint8(double(RemoveMSB(Stego, n))*2^(8-n));
psnr_num=800*600*max(max(Hare.^2));              % calculate numerator
      psnr_den=sum(sum(Hare-Hare1).^2);      % calculate denominator
      psnr=psnr_num/psnr_den
Hare1=uint8(Hare1);
dicomwrite(Hare1, 'c:\temp\wm5.dcm');
imview(Hare1, [])
%figure,imshow(Extracted)
Hare2=dicomread('c:\temp\wm5.dcm');

Hare2 = double(Hare2);
Hare3 = Hare2;
Arr1=[];
for i = 241:248
   for j =9:40
      for b=2%2:-1:1

         Arr2= [bitget(Hare3(i,j),b)];
         Hare3(i,j)=bitset(Hare3(i,j),b,0);
         Arr1=[Arr1 Arr2];
      end
   end
end
Arr1
Hare3=uint8(Hare3);
%x= mat2str(Hare3);
%y= md5(x)
%dicomwrite(Hare3,'c:\temp\result4.dcm');
%imview(Hare3, [])
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   This program implements the Secure Hash Standard SHA-256 as set forth by
%   the Federal Information Processing Standards Publication 180-2.
%
%
%
%Inputs:
%         1. Input File - name of the input file. The file must be text
%            file in ASCII format.
%         2. Output File - name/location of the output file.
%
%Outputs:
%         1. Output File- The final hash value is
%            placed in this file as a Hex value on the first line.
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function sha256()

%Ask for user input
fname=input('Input File (in ASCII format)? ','s');
hash_foutname=input('Output File for SHA256 Hash? ','s');

%Open the input file and get the first line of data
fid=fopen(fname);
M = fread(fid);
fclose(fid);

%Convert the input message from ASCII to 8-bit binary values for each character
% M=dec2bin((abs(input(1))),8);
% for i = 2:length(input)
%     M=strcat(M,dec2bin(abs(input(i)),8));
% end

%Get Constants - K256 and initial hash values, H0
[K256 H]=constants(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 PREPROCESSING SECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%PAD THE MESSAGE

%Calculate the number of zeros needed to pad the message up to 448
len=8*length(M);
k=mod(448-mod(len,512)-1,512);

if (k > 0)
   M(length(M)+1)=128;
end

for i=2:(k+1)/8
   M(length(M)+1)=0;
end

%Append the bit value of the length of the message to fill up to 512
len_bin=dec2base(len,2,64);
for i=1:8
   M(length(M)+1) = bin2dec(len_bin(1,(8*i-7):(8*i)));
end


%PARSING THE PADDED MESSAGE

%Calculate the number of blocks in the message
N=length(M)/64;

%Split the message into N 512-bit blocks of message
%Each N block has 16 32-bit blocks
cnt = 1;
for i=1:N
   for j=1:16

M_parsed(i,j)=bitshift(M(cnt),24)+bitshift(M(cnt+1),16)+bitshift(M(cnt+2),8)+M(cnt+
3);
      cnt = cnt + 4;
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              PROCESSING SECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Process each 512 bit block of Message individually
for i = 1:N
```

```
%PREPARE THE MESSAGE SCHEDULE
%The first 16 blocks of message schedules are 32-bit blocks of the N block
for t = 1:16
   W(t)=M_parsed(i,t);
end

%The next 48 message schedules are calculated as follows from the initial 16
message schedules
for t = 17:64

   W(t) = add4num32(gam1(W(t-2)),W(t-7),gam0(W(t-15)),W(t-16));

end

%Intialize the eight working variables to initial hash values
a = H(i,1);
b = H(i,2);
c = H(i,3);
d = H(i,4);
e = H(i,5);
f = H(i,6);
g = H(i,7);
h = H(i,8);

%Compute all 64 iterations of the eight working variables
for t = 1:64

   T1 = add5num32(h,eps1(e),Ch(e,f,g),K256(t),W(t));
   T2 = mod(eps0(a) + Maj(a,b,c),2^32);
   h = g;
   g = f;
   f = e;
   e = mod((d + T1),2^32);
   d = c;
   c = b;
   b = a;
   a = mod((T1 + T2),2^32);

end

%Compute the i-th hash values for N block
H(i+1,1) = mod((a + H(i,1)),2^32);
H(i+1,2) = mod((b + H(i,2)),2^32);
H(i+1,3) = mod((c + H(i,3)),2^32);
H(i+1,4) = mod((d + H(i,4)),2^32);
H(i+1,5) = mod((e + H(i,5)),2^32);
```

```
    H(i+1,6) = mod((f + H(i,6)),2^32);
    H(i+1,7) = mod((g + H(i,7)),2^32);
    H(i+1,8) = mod((h + H(i,8)),2^32);

end

%Open the output file and store hex values of each hash as one line

fid=fopen(hash_foutname,'at');
fprintf(fid,'%s',dec2hex(H(N+1,1),8));
fprintf(fid,'%s',dec2hex(H(N+1,2),8));
fprintf(fid,'%s',dec2hex(H(N+1,3),8));
fprintf(fid,'%s',dec2hex(H(N+1,4),8));
fprintf(fid,'%s',dec2hex(H(N+1,5),8));
fprintf(fid,'%s',dec2hex(H(N+1,6),8));
fprintf(fid,'%s',dec2hex(H(N+1,7),8));
fprintf(fid,'%s',dec2hex(H(N+1,8),8));
fclose(fid);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                FUNCTIONS SECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Function: gam0
%
%  Defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = gam0(x)

result1=rotr(x,7);
result2=rotr(x,18);
result3=shr(x,3);
result = bitxor(bitxor(result1,result2),result3);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
%
%  Function: gam1
%
%  Defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = gam1(x)

result1=rotr(x,17);
result2=rotr(x,19);
result3=shr(x,10);
result = bitxor(bitxor(result1,result2),result3);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Function: eps0
%
%  Defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = eps0(x)

result1=rotr(x,2);
result2=rotr(x,13);
result3=rotr(x,22);
result = bitxor(bitxor(result1,result2),result3);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Function: eps1
%
%  Defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = eps1(x)

result1=rotr(x,6);
result2=rotr(x,11);
result3=rotr(x,25);
result = bitxor(bitxor(result1,result2),result3);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Function: shr
%
%  Shifts a binary number x positions to
%  the right, placing '0' values in the
%  x left positions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = shr(x,n)

result = bitshift(x,-n,32);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Function: rotr
%
%  Shifts a binary number x positions to
%  the right, rotating the shifted values
%  back into the left.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = rotr(x,n)

result = bitor(bitshift(x,-n,32),(bitshift(x,32-n,32)));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  function: add4num32
%
%  Adds 4 32-bit numbers
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = add4num32(x1,x2,x3,x4)

result=mod(x1+x2+x3+x4,2^32);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  function: add5num32
%
%  Adds 5 32-bit numbers
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = add5num32(x1,x2,x3,x4,x5)

result=mod(x1+x2+x3+x4+x5,2^32);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  function: Ch
%
%  defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = Ch(x,y,z)

temp1 = bitand(x,y);
temp2 = bitand(bitcmp(x,32),z);
result = bitxor(temp1,temp2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  function: Maj
%
%  defined SHA256 function
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [result] = Maj(x,y,z)

temp1 = bitand(x,y);
temp2 = bitand(x,z);
temp3 = bitand(z,y);
result = bitxor(bitxor(temp1,temp2),temp3);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  function: constants
%
%  produces SHA256 constants
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [K256,H] = constants(temp);

%SHA-256 Constant Definitions
%1
K256(1)=1116352408;K256(2)=1899447441;K256(3)=3049323471;K256(4)=3921009
573;
K256(5)=961987163;K256(6)=1508970993;K256(7)=2453635748;K256(8)=28707632
21;
%2
K256(9)=3624381080;K256(10)=310598401;K256(11)=607225278;K256(12)=142688
1987;
K256(13)=1925078388;K256(14)=2162078206;K256(15)=2614888103;K256(16)=324
8222580;
%3
K256(17)=3835390401;K256(18)=4022224774;K256(19)=264347078;K256(20)=6048
07628;
K256(21)=770255983;K256(22)=1249150122;K256(23)=1555081692;K256(24)=1996
064986;
%4
K256(25)=2554220882;K256(26)=2821834349;K256(27)=2952996808;K256(28)=321
0313671;
K256(29)=3336571891;K256(30)=3584528711;K256(31)=113926993;K256(32)=3382
41895;
%5
K256(33)=666307205;K256(34)=773529912;K256(35)=1294757372;K256(36)=13961
82291;
K256(37)=1695183700;K256(38)=1986661051;K256(39)=2177026350;K256(40)=245
6956037;
%6
K256(41)=2730485921;K256(42)=2820302411;K256(43)=3259730800;K256(44)=334
5764771;
K256(45)=3516065817;K256(46)=3600352804;K256(47)=4094571909;K256(48)=275
423344;
%7
K256(49)=430227734;K256(50)=506948616;K256(51)=659060556;K256(52)=883997
877;
K256(53)=958139571;K256(54)=1322822218;K256(55)=1537002063;K256(56)=1747
873779;
```

```
%8
K256(57)=1955562222;K256(58)=2024104815;K256(59)=2227730452;K256(60)=236
1852424;
K256(61)=2428436474;K256(62)=2756734187;K256(63)=3204031479;K256(64)=332
9325298;

%Intial Hash Values

H(1,1)=1779033703;
H(1,2)=3144134277;
H(1,3)=1013904242;
H(1,4)=2773480762;
H(1,5)=1359893119;
H(1,6)=2600822924;
H(1,7)=528734635;
H(1,8)=1541459225;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:                Jasni Zain
%Project:       Image relocation using toral automorphism
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% save start time
start_time=cputime;

blocksize=4;                % set the blocksize

% read in the cover object
file_name='sig4.bmp';
cover_object=(imread(file_name));
%cover_object=rgb2gray(cover_object);
cover_object=double(cover_object);
% determine size of cover image
Mc=size(cover_object,1);            %Height
Nc=size(cover_object,2);            %Width
%ABB=[];
Br=floor(Nc/blocksize);  % Blocks per row
Bc= floor(Mc/blocksize); % Blocks per column

numblock= Br*Bc; % number of blocks
k=max(primes(numblock/2));


ABB=[];
for A= 1:numblock
   AB = mod((k*A), numblock)+1; %mapping the blocks
   ABB=[ABB AB];
end
B=[1:numblock];
mapA= [B;ABB];

mapB=[];mapBB=[];
for i= 1:numblock
   mapB(1,mapA(2,i))=mapA(2,i); %mapping the blocks
   mapB(2,mapA(2,i))=mapA(1,i);

end
new_image= cover_object;
x=1; y=1;
for i=1:numblock
    % numbering block
```

```
   block=cover_object(y:y+blocksize-1,x:x+blocksize-1);

   targetblock= mapA(2,i);
   rownum=round( ceil(targetblock/Br));
   colnum= round(targetblock-(rownum-1)*Br);
   xs=(rownum-1)*blocksize+1;
   ys=(colnum-1)*blocksize+1;


   if (x+blocksize) > Nc
    if y+blocksize < Mc
       x=1;
       y=y+blocksize;
   end
    else
       x=x+blocksize;

   end
       new_image(xs:xs+blocksize-1,ys:ys+blocksize-1)=block;
 end

%sub-block watermark generation
toral_image=uint8(new_image);
imwrite(toral_image,'toraltest3.bmp','bmp');


% display processing time
%elapsed_time=cputime-start_time,

% display psnr of watermarked image
%psnr=psnr(cover_object,watermarked_image),

% display watermarked image
figure
imshow(toral_image)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:                  Jasni Zain
%Project:       Spread Embedding for Tamper detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% save start time
start_time=cputime;

blocksize=8;                 % set the blocksize
percent= 50;               %set tamper percentage

% read in the cover object
file_name='ustest.bmp';
cover_object=(imread(file_name));
%cover_object=rgb2gray(cover_object);
cover_object=double(cover_object);
% determine size of cover image
Mc=size(cover_object,1);             %Height
Nc=size(cover_object,2);             %Width
areatam= Mc*Nc*percent;              % area of tamper
%ABB=[];
Br=floor(Nc/blocksize);  % Blocks per row
Bc= floor(Mc/blocksize); % Blocks per column

numblock= Br*Bc; % number of blocks
numtamblk= floor(numblock*percent/100);

factam=floor(100/percent);  %factor of tampered block
k=max(primes(numblock/2));
tampered_image=cover_object;
for i= 1:numtamblk
   targetblock= i;
   rownum=round( ceil(targetblock/Br));
   colnum= round(targetblock-(rownum-1)*Br);
   ys=(rownum-1)*blocksize+1;
   xs=(colnum-1)*blocksize+1;
   startblock=cover_object(ys:ys+blocksize-1,xs:xs+blocksize-1);%start of target block
      for ii=1:blocksize
         for jj= 1:blocksize
            startblock(ii,jj)=255;
         end
      end
      tampered_image(ys:ys+blocksize-1,xs:xs+blocksize-1)= startblock;
 end
 tampered_image=uint8(tampered_image);
```

```
imwrite(tampered_image,'tamper250.bmp','bmp');


% display watermarked image
figure
imshow(tampered_image)
title('Tamper detect')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:              Jasni Zain
%Project:      Embedding for Tamper detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% save start time
start_time=cputime;

blocksize=8;                % set the blocksize


% read in the cover object
file_name='ultrasound2.jpg';
cover_object=(imread(file_name));
cover_object=rgb2gray(cover_object);
cover_object=double(cover_object);
% determine size of cover image
Mc=size(cover_object,1);            %Height
Nc=size(cover_object,2);            %Width

Br=floor(Nc/blocksize);  % Blocks per row
Bc= floor(Mc/blocksize); % Blocks per column
ABB=[];
numblock= Br*Bc; % number of blocks
k=max(primes(numblock/2));

for A= 1:numblock
   AB = mod((k*A), numblock)+1; %mapping the blocks
   ABB=[ABB AB];
end
B=[1:numblock];
mapA= [B;ABB];
mapB=[];
for i= 1:numblock
   mapB(1,mapA(2,i))=mapA(2,i); %mapping the blocks
   mapB(2,mapA(2,i))=mapA(1,i);

end
mapB;

% generate shell of watermarked image
watermarked_image=cover_object;
x=1;
y=1;
```

```
% process the image in blocks

for i = 1:numblock

   % numbering block
   block=cover_object(y:y+blocksize-1,x:x+blocksize-1);
   cover= RemoveLSB(block,1); % reset the LSB to 0
   AvgB=round(sum(sum(cover))/(blocksize*blocksize)); % average of block
   targetblock= mapB(2,i);
   rownum=round( ceil(targetblock/Br));
   colnum= round(targetblock-(rownum-1)*Br);
   ys=(rownum-1)*blocksize+1;
   xs=(colnum-1)*blocksize+1;
   startblock=cover_object(ys:ys+blocksize-1,xs:xs+blocksize-1);%start of target block
   cover1=RemoveLSB(startblock,1);

   % prepare sub-block
   x1=1;
   y1=1;
   for sub= 1:4
   subblok= cover(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1);
   AvgBs=round(sum(sum(subblok))/(blocksize/2)*(blocksize/2));
   if AvgBs >= AvgB
      v=1;
   else v=0;
   end
   %parity
   par=0;
   embedbit=[];

    for b=8:-1:2
      bit=bitget(AvgBs,b);
         if bit==1
         par=par+1;
         end
      end

   if rem (par,2)==0 %even
      p= 1;
   else p=0;
   end
   substart=cover1(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1);
   AvgBc=round((sum(sum(substart)))/((blocksize/2)*(blocksize/2)));
      for b=8:-1:2
      bit=bitget(AvgBc,b);
      embedbit=[embedbit bit];
      end
```

```
   embedbit=[v p embedbit]; %the watermark (v,p,r)
   %embed in 2x2 subblock
   n=1;
   for ii=y1:y1+(blocksize/2)-2
      for jj=x1:x1+(blocksize/2)-2
         block(ii,jj)=bitset(block(ii,jj),1,embedbit(n));
         n=n+1;
      end
   end

  % block(x1:x1+(blocksize/2)-1, y1:y1+(blocksize/2)-1)=subblok;

   if (x1+(blocksize/2))> blocksize
      if y1 +(blocksize/2) < blocksize
         x1=1;
         y1= y1+(blocksize/2);
      end
      else
         x1=x1+(blocksize/2);

   end
end
 watermarked_image(y:y+blocksize-1, x:x+blocksize-1)=block;

 if (x+blocksize) > Nc
    if y+blocksize < Mc
      x=1;
      y=y+blocksize;
   end
    else
      x=x+blocksize;

   end
 end

difference= cover_object- watermarked_image;
imshow(difference,[-1 1])
%sub-block watermark generation
watermarked_image_int=uint8(watermarked_image);
imwrite(watermarked_image_int,'ustest.bmp','bmp');


% display processing time
%elapsed_time=cputime-start_time,
```

```
% display psnr of watermarked image
psnr=psnr(cover_object,watermarked_image),

% display watermarked image

imview(watermarked_image_int)
title('Watermarked Image')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:          Jasni Zain
%Project:       Level-2 detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% save start time
start_time=cputime;

blocksize=8;            % set the blocksize


% read in the cover object
file_name='tamper250.bmp';
cover_object=(imread(file_name));

%cover_object=rgb2gray(cover_object);
cover_object=double(cover_object);
% determine size of cover image
Mc=size(cover_object,1);        %Height
Nc=size(cover_object,2);        %Width
Br=floor(Nc/blocksize);  % Blocks per row
Bc= floor(Mc/blocksize); % Blocks per column

ABB=[];
numblock= Br*Bc; % number of blocks
k=max(primes(numblock/2));
for A= 1:numblock
   AB = mod((k*A), numblock)+1; %mapping the blocks
   ABB=[ABB AB];
end
B=[1:numblock];
mapA= [B;ABB];


mapB=[];
for i= 1:numblock
   mapB(1,mapA(2,i))=mapA(2,i); %mapping the blocks
   mapB(2,mapA(2,i))=mapA(1,i);

end
mapB;

% determine maximum message size based on cover object, and blocksize
max_message=Mc*Nc/(blocksize^2);
```

```
% generate shell of watermarked image
recover_image=cover_object;
x=1;
y=1;

% process the image in blocks
%for block=1:(round(Mc/blocksize)* round(Nc/blocksize))

%sumblock=sum(sum( cover_object(x:x+blocksize-1,y:y+blocksize-1)));
%wm= round(sumblock/(blocksize*blocksize));
tamperblock=0;
for i = 1:numblock

   % numbering block
   block=cover_object(y:y+blocksize-1,x:x+blocksize-1);
   cover= RemoveLSB(block,1); % reset the LSB to 0
   AvgB=round(sum(sum(cover))/(blocksize*blocksize)); % average of block

   targetblock= mapA(2,i);
   rownum=round( ceil(targetblock/Br));
   colnum= round(targetblock-(rownum-1)*Br);
   ys=(rownum-1)*blocksize+1;
   xs=(colnum-1)*blocksize+1;
   startblock=cover_object(ys:ys+blocksize-1,xs:xs+blocksize-1);
   cover1=RemoveLSB(startblock,1);
  % AvgBc=round(sum(sum(cover1))/(blocksize*blocksize));

   % prepare sub-block
   x1=1;
   y1=1;
   for sub= 1:4
      subblok= block(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1);
      bitv= bitget(subblok(1,1),1);   % getting v from sub block
      bitp= bitget(subblok(1,2),1);   % getting p from sub block
      subblok=RemoveLSB(subblok,1);
      AvgBs=round(sum(sum(subblok))/(blocksize/2)*(blocksize/2));
      if     AvgBs >= AvgB
            v1=1;
      else    v1=0;
      end
      %parity
      par=0;
      embedbit=[];

      for b=8:-1:2
         bit=bitget(AvgBs,b);
         embedbit=[embedbit bit];
```

```
    if  bit==1
        par=par+1;
    end
end

if   rem (par,2)==0 %even
     p1= 1;
else p1=0;
end

if and(p1==bitp, v1==bitv)
    for ii= 1:(blocksize/2)
        for jj=1:(blocksize/2)
            subblok(ii,jj)=subblok(ii,jj);

        end
    end

    block(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1)=subblok;
    if (x1+(blocksize/2))> blocksize
        if y1 +(blocksize/2) < blocksize
        x1=1;
        y1= y1+(blocksize/2);
        end
    else
    x1=x1+(blocksize/2);

    end
else
%find recovery block

x1=1;y1=1;
 for l=1:4

    substart=startblock(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1);
    n=1;
    targetbit=0;
    for ii=y1:y1+(blocksize/2)-2
        for jj=x1:x1+(blocksize/2)-2

            bit=bitget(startblock(ii,jj),1);
            targetbit=bitset(targetbit,8-(n-3), bit);

           n=n+1;
        end
    end
```

```
        targetbit=bitset(targetbit,10, 0);
        targetbit=bitset(targetbit,9, 0);

          for ii= 1:(blocksize/2)
             for jj=1:(blocksize/2)
                subblok(ii,jj)=targetbit;
             end
          end
          block(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1)=subblok;
           tamperblock=tamperblock+1;
          if (x1+(blocksize/2))> blocksize
             if y1 +(blocksize/2) < blocksize
             x1=1;
             y1= y1+(blocksize/2);
             end
          else
          x1=x1+(blocksize/2);

          end
       end

       end



    end
    recover_image(y:y+blocksize-1, x:x+blocksize-1)=block;

       if (x+blocksize) > Nc
          if y+blocksize < Mc
             x=1;
             y=y+blocksize;
          end
       else
       x=x+blocksize;

       end
  end


%sub-block watermark generation
recover_image_int=uint8(recover_image);
imwrite(recover_image_int,'recovered250.bmp','bmp');

tamperblock
% display processing time
%elapsed_time=cputime-start_time,
```

```
% display psnr of watermarked image
%psnr=psnr(cover_object,watermarked_image),

% display watermarked image
figure
imshow(recover_image,[0 255])
title('Tamper detect')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Name:          Jasni Zain
%Project:       Image recovery
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% save start time
start_time=cputime;

blocksize=8;           % set the blocksize


% read in the cover object
file_name='ustamp.bmp';
filename2='sig4.bmp';
sig_file=(imread(filename2));
sig_file=double(sig_file);
cover_object=(imread(file_name));

%cover_object=rgb2gray(cover_object);
cover_object=double(cover_object);
% determine size of cover image
Mc=size(cover_object,1);          %Height
Nc=size(cover_object,2);          %Width
Br=floor(Nc/blocksize);  % Blocks per row
Bc= floor(Mc/blocksize); % Blocks per column

ABB=[];
numblock= Br*Bc; % number of blocks
k=max(primes(numblock/2));
for A= 1:numblock
   AB = mod((k*A), numblock)+1; %mapping the blocks
   ABB=[ABB AB];
end
B=[1:numblock];
mapA= [B;ABB];


mapB=[];
for i= 1:numblock
   mapB(1,mapA(2,i))=mapA(2,i); %mapping the blocks
   mapB(2,mapA(2,i))=mapA(1,i);

end
mapB;
```

```
% determine maximum message size based on cover object, and blocksize
max_message=Mc*Nc/(blocksize^2);

% generate shell of watermarked image
recover_image=cover_object;
x=1;
y=1;

% process the image in blocks
%for block=1:(round(Mc/blocksize)* round(Nc/blocksize))

%sumblock=sum(sum( cover_object(x:x+blocksize-1,y:y+blocksize-1)));
%wm= round(sumblock/(blocksize*blocksize));
tamperblock=0;
for i = 1:numblock

   % numbering block
   block=cover_object(y:y+blocksize-1,x:x+blocksize-1);

   cover= RemoveLSB(block,1); % reset the LSB to 0
   AvgB=round(sum(sum(cover))/(blocksize*blocksize)); % average of block

   targetblock= mapA(2,i);
   rownum=round( ceil(targetblock/Br));
   colnum= round(targetblock-(rownum-1)*Br);
   ys=(rownum-1)*blocksize+1;
   xs=(colnum-1)*blocksize+1;
   startblock=cover_object(ys:ys+blocksize-1,xs:xs+blocksize-1);
   cover1=RemoveLSB(startblock,1);
   AvgBc=round(sum(sum(cover1))/(blocksize*blocksize));

   % prepare sub-block
   x1=1;
   y1=1;
   for sub= 1:4
      subblok= block(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1);
      bitv= bitget(subblok(1,1),1);   % getting v from sub block
      bitp= bitget(subblok(1,2),1);   % getting p from sub block
      subblok=RemoveLSB(subblok,1);
      AvgBs=round(sum(sum(subblok))/(blocksize/2)*(blocksize/2));
      if     AvgBs >= AvgB
            v1=1;
      else    v1=0;
      end
      %parity
      par=0;
      embedbit=[];
```

```
    for b=8:-1:2
        bit=bitget(AvgBs,b);
        embedbit=[embedbit bit];
        if  bit==1
            par=par+1;
        end
    end

    if   rem (par,2)==0 %even
         p1= 1;
    else p1=0;
    end

    if and(p1==bitp, v1==bitv)
        for ii= 1:(blocksize/2)
            for jj=1:(blocksize/2)
                subblok(ii,jj)=subblok(ii,jj);

            end
        end
    else
    %find recovery block
      block=sig_file(y:y+blocksize-1,x:x+blocksize-1);
   end

     %block(y1:y1+(blocksize/2)-1, x1:x1+(blocksize/2)-1)=subblok;

    if (x1+(blocksize/2))> blocksize
        if y1 +(blocksize/2) < blocksize
            x1=1;
            y1= y1+(blocksize/2);
        end
    else
        x1=x1+(blocksize/2);

    end
end
recover_image(y:y+blocksize-1, x:x+blocksize-1)=block;

    if (x+blocksize) > Nc
        if y+blocksize < Mc
            x=1;
            y=y+blocksize;
        end
    else
    x=x+blocksize;
```

```
      end
 end


%sub-block watermark generation
recover_image_int=uint8(recover_image);
imwrite(recover_image_int,'us3.bmp','bmp');

tamperblock
% display processing time
%elapsed_time=cputime-start_time,

% display psnr of watermarked image
%psnr=psnr(cover_object,watermarked_image),

% display watermarked image
figure
imshow(recover_image,[0 255])
title('Tamper detect')
```

Appendix C – Recovered Images


PLEASE WAIT
Taroni rene
LVO 71.L.001
1 /6/99
P 5
B:7.0 / OPT 112mm / G 77
CFM:3.6 PRF 1.3kHz /FLT M/G 115
1 :00:35 P
SCC 9
Spread 10%
D1
D2
D3
D4
D5
Clear

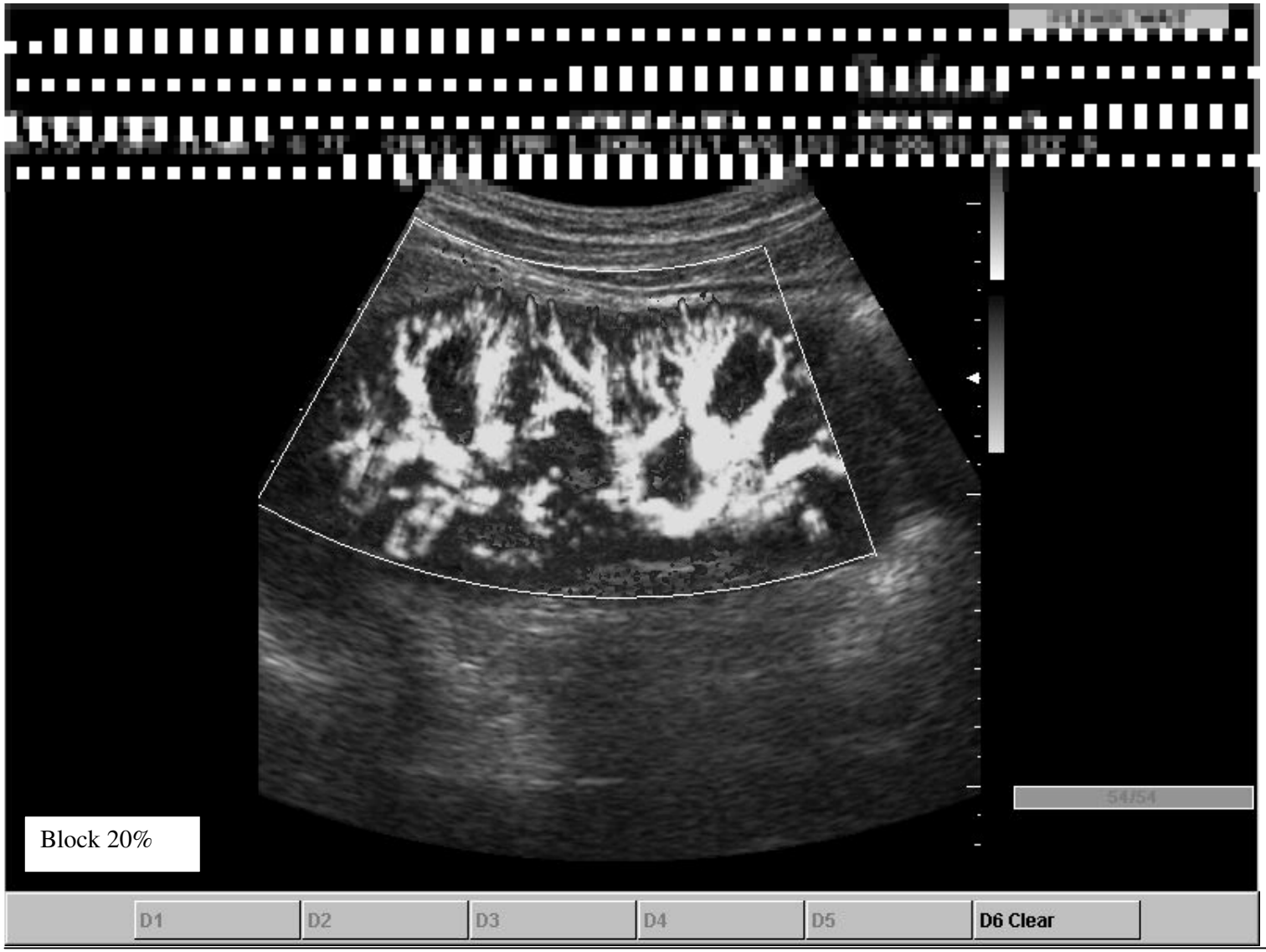

PLEASE WAIT
Technos
Spread 20%
D1
D2
D3
D4
D5
D6 Clear

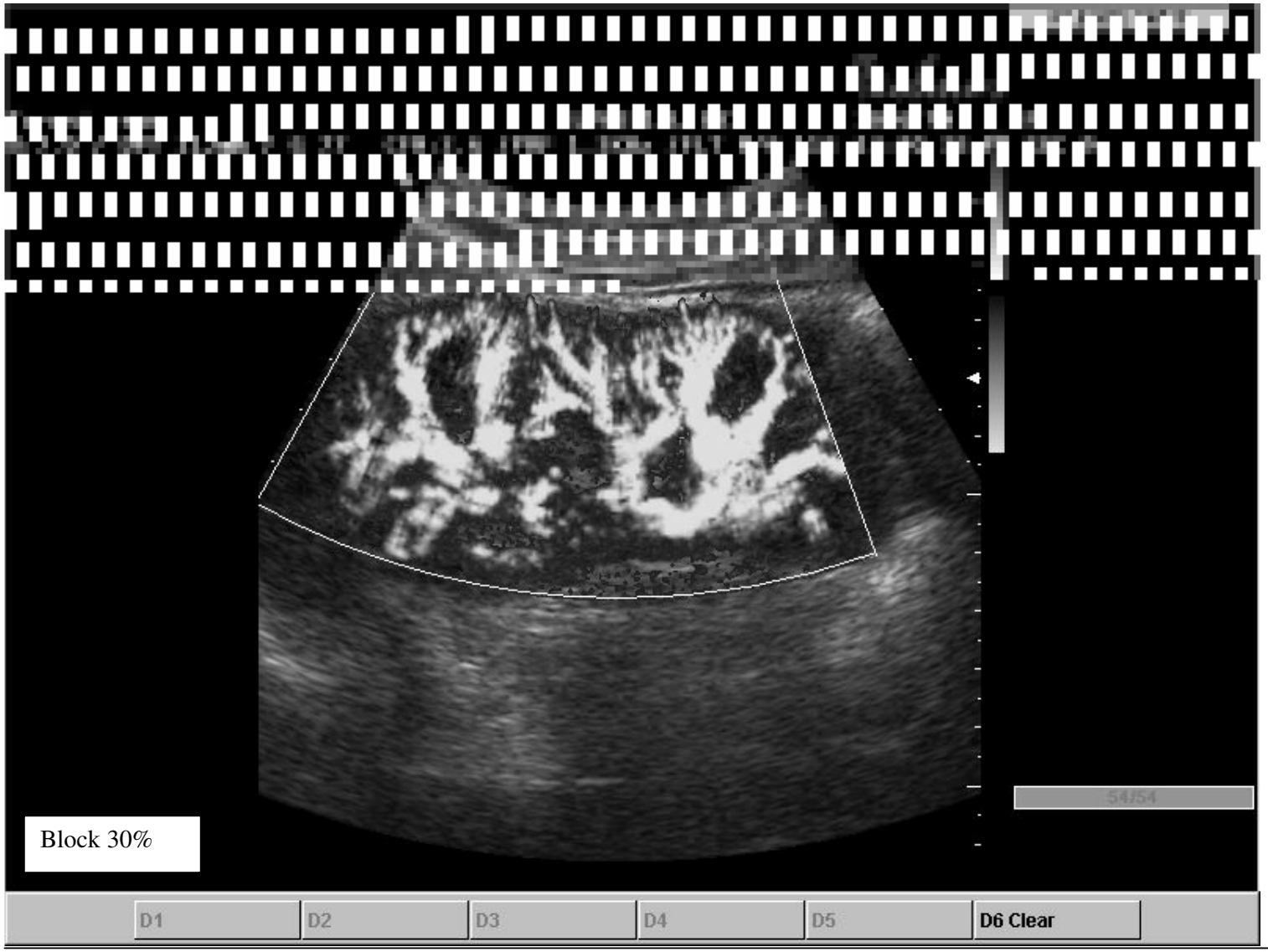

PLEASE WAIT
Technos
LV0571.L.001
10/6/99
Spread 30%
D1
D2
D3
D4
D5
D6 Clear

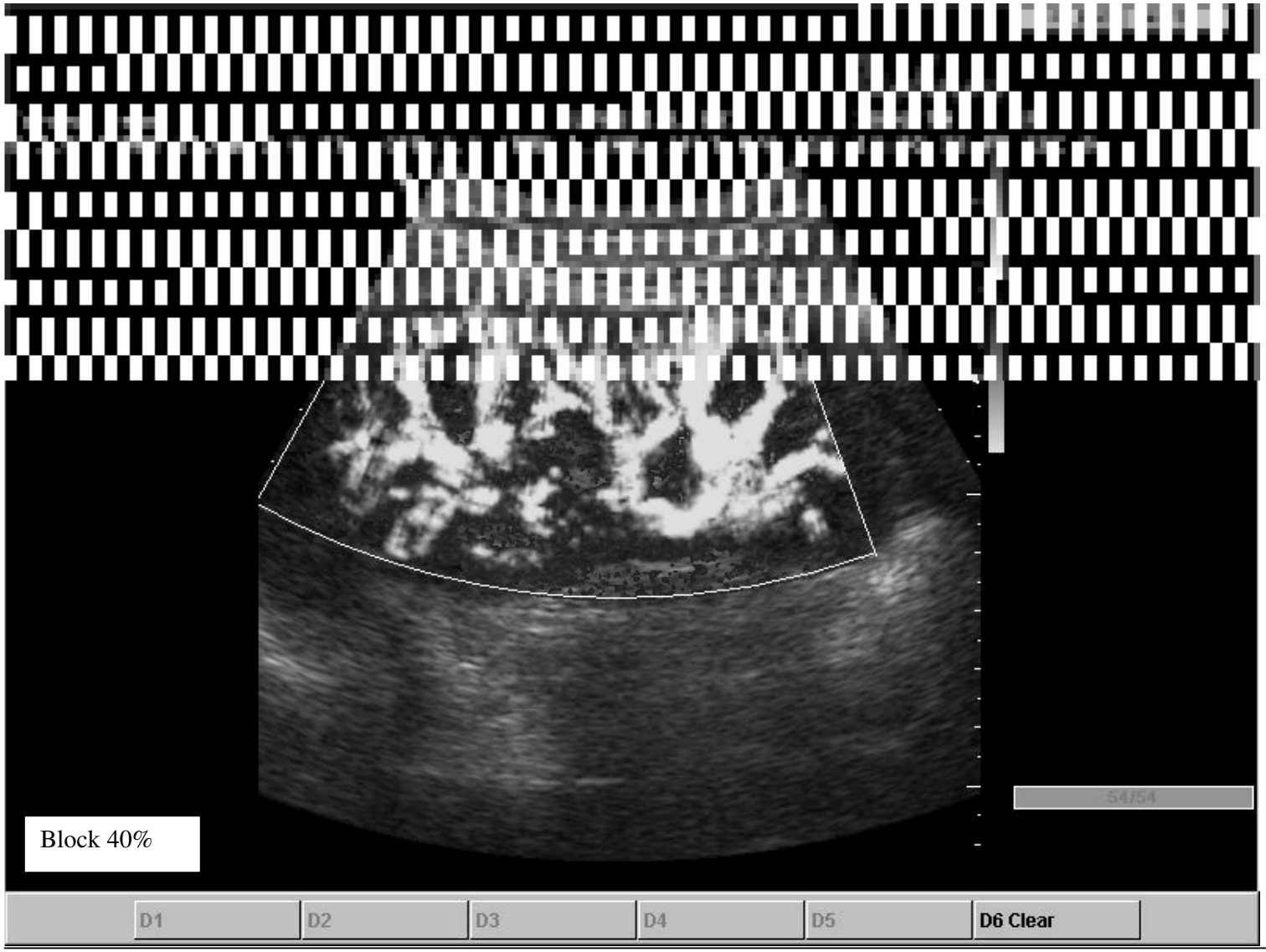

Spread 40%
D1
D2
D3
D4
D5
D6 Clear

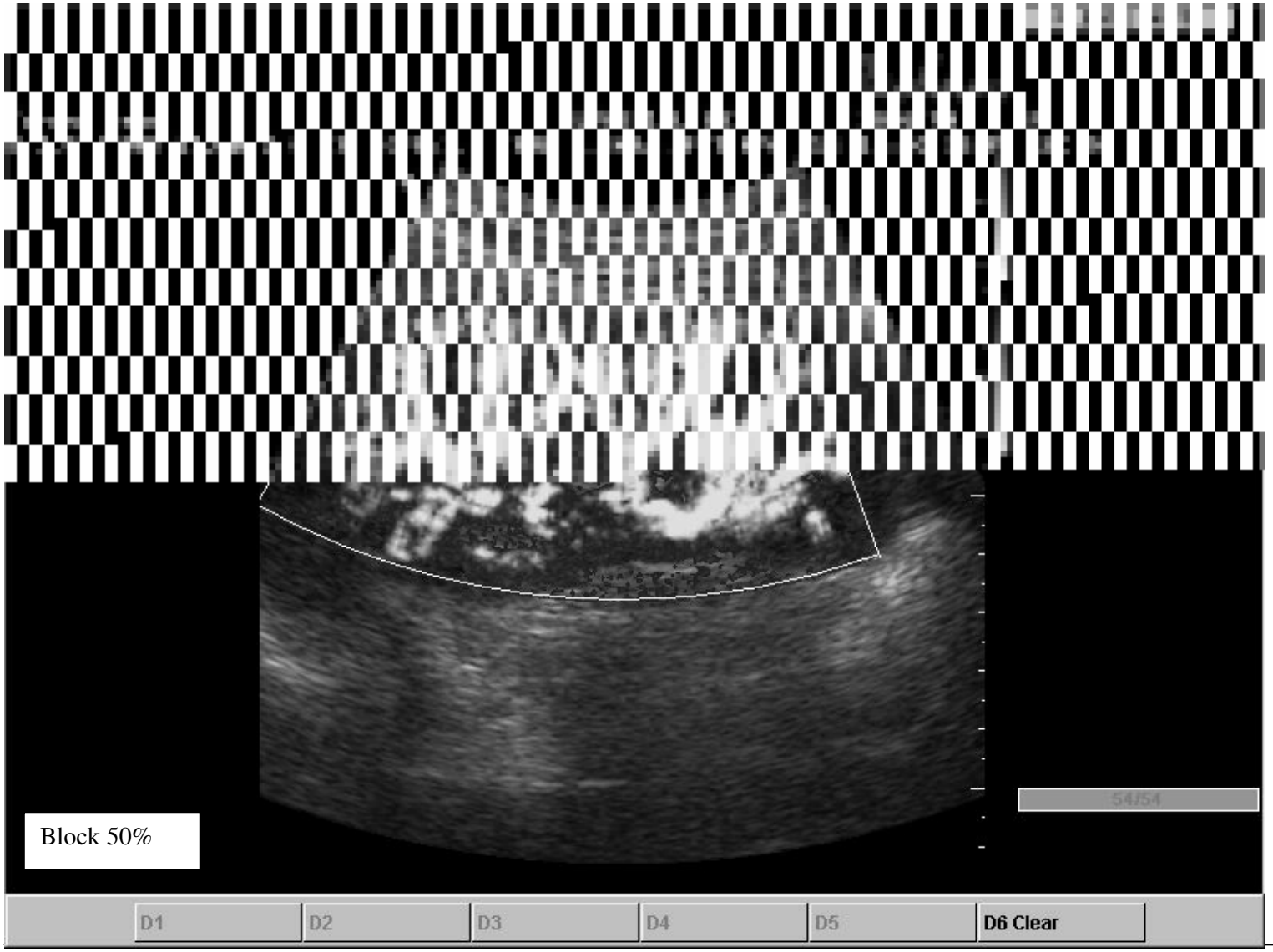

Spread 50%


Taroni rene
LV0571.L.001
10/6/99
P 5
B:7.0 / DPT 112mm / G 77
CFM:3.6 /PRF 1.3KHz /FLT M/G 115
12:00:35 PM SCC 9
Block 10%
D1
D2
D3
D4
D5
D6 Clear


Block 20%
D1
D2
D3
D4
D5
D6 Clear


Block 30%
D1
D2
D3
D4
D5
D6 Clear


Block 40%
D1
D2
D3
D4
D5
D6 Clear


Block 50%
54/54
D1
D2
D3
D4
D5
D6 Clear