2005

Augmented watermarking

Kiranmayi Penumarthi
Louisiana State University and Agricultural and Mechanical College, kpenum1@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses

Part of the Electrical and Computer Engineering Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_theses/2100

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master’s Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
AUGMENTED WATERMARKING

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

By
Kiranmayi Penumarthi
Bachelor of Technology, Jawaharlal Nehru Technological University
Hyderabad, India, 2002
December 2005
Acknowledgements

I would like to acknowledge certain people who have encouraged, supported and helped me complete my thesis at LSU.

I am extremely grateful to my advisor Dr. Subhash Kak for his guidance, patience and understanding all through this work. His suggestions, discussions and constant encouragement have helped me gain a deep insight in the field of Watermarking. I would like to thank Dr. Suresh Rai and Dr. Xue Bin Liang for sparing their time to be a part of my advisory committee. I am thankful to Mr. Bob Zinn for all his support and encouragement. I would also thank Ashfaq Naveed Shaik for all his help. I am grateful to my parents Mr. and Mrs. Ramachandra Rao and my sisters for the tremendous amount of inspiration they have given me.

I would like to thank all my friends here at LSU, who have helped me all through my stay at LSU and have made my stay a pleasant one.
# Table of Contents

Acknowledgements ........................................................................................................... ii  
List of Tables .................................................................................................................. v  
List of Figures .................................................................................................................. vi  
Abstract ........................................................................................................................ viii  

## Chapter 1 Introduction ................................................................................................ 1  
1.1 Digital Rights Management .................................................................................. 1  
   1.1.1 Copy and Copyright Protection ........................................................................ 3  
1.2 Steganography ........................................................................................................ 4  
1.3 Digital Watermarking ............................................................................................ 5  
1.4 Types of Watermarks ............................................................................................. 7  
   1.4.1 Visible Watermarks ......................................................................................... 7  
   1.4.2 Invisible Watermarks ..................................................................................... 8  
1.5 Watermarking Techniques .................................................................................... 9  
1.6 Watermarking Attacks .......................................................................................... 12  
1.7 Watermarking Process ......................................................................................... 13  
1.8 Outline of Thesis .................................................................................................. 13  

## Chapter 2 Spatial and Spectral Domain Watermarking ............................................ 15  
2.1 Pseudo Random Noise Sequences ....................................................................... 15  
   2.1.1 Properties of PN Sequences .......................................................................... 16  
   2.1.2 Watermarking using PN Sequences ............................................................... 17  
2.2 Decimal Sequences ............................................................................................... 21  
   2.2.1 Properties of Decimal Sequences ................................................................. 21  
   2.2.2 Generating the Decimal Sequences ............................................................... 24  
   2.2.3 Watermarking using Decimal Sequences ..................................................... 26  
2.3 Spectral Domain Watermarking .......................................................................... 28  
2.4 Invertible Watermarking ...................................................................................... 29  
   2.4.1 Invertible Spread Spectrum Watermarking ................................................... 29  
   2.4.2 Algorithm for Spread Spectrum Watermarking Inversion ......................... 30  

## Chapter 3 Augmented Watermarking ....................................................................... 31  
3.1 Introduction ............................................................................................................ 31  
3.2 Image Noisification Scheme ................................................................................ 31  
   3.2.1 Key for Noise Embedding .......................................................................... 32  
   3.2.2 Noise Embedding Method using the key ......................................................... 35  
3.3 Image Retrieval Scheme ...................................................................................... 35  
3.4 Noise Removal ...................................................................................................... 36  
3.5 Implementation in Different Domains .................................................................. 37  
   3.5.1 Image Noisification and Retrieval for PN Sequences ................................... 37  
   3.5.2 Image Noisification and Retrieval for Decimal Sequences ......................... 38
3.5.3 Image Noisification and Retrieval for DCT Watermarking .......... 39
3.6 Overlaid Augmented Watermarking ........................................... 41
3.7 Extension to Digital Rights Management ................................. 45
3.7.1 Protocol for Obtaining the Original Data ............................... 47

Chapter 4 Analysis of Results .......................................................... 48
  4.1 Analysis of Gain ........................................................................ 48
  4.2 Analysis of Augmented Watermarking on PN Sequences ............ 48
  4.3 Analysis of Augmented Watermarking on Decimal Sequences .... 50
  4.4 Analysis of Augmented Watermarking in the Frequency Domain ... 51
  4.5 Robustness to External Noise .................................................. 52

Chapter 5 Conclusions .................................................................. 55

Bibliography ................................................................................. 56

Vita ............................................................................................... 59
List of Tables

4.1 Correlation and PSNR values for various watermarks for PN sequences for lower gains…………………………………………………………………………………… 49
4.2 Correlation and PSNR values for various watermarks for PN sequences for higher gains……………………………………………………………………………………. 50
4.3 Correlation and PSNR values for various watermarks for PN sequences for different values of gain………………………………………………………………….50
4.4 Correlation and PSNR values for various watermarks for D-sequences………… 51
4.5 Correlation and PSNR values for various watermarks for DCT watermarking …… 51
List of Figures

Figure 1.1 DRM architecture ................................................................. 2
Figure 1.2 Branches of Steganography ...................................................... 5
Figure 1.3 Visible Watermarking ............................................................... 7
Figure 1.4 Invisible Watermarking ............................................................. 8
Figure 1.5 LSB encoding ................................................................. 9
Figure 1.6 Watermarking the Media .......................................................... 11
Figure 1.7 Common attacks that a Media would undergo ................................ 12
Figure 1.8 Watermark Embedding ........................................................... 13
Figure 1.9 Watermark Detection .............................................................. 13
Figure 2.1 Basic circuit for a PN generator .................................................. 15
Figure 2.2 Outputs for Different Stages of PN generator ................................. 15
Figure 2.3 Watemark Embedding ............................................................ 18
Figure 2.4 Watermark Recovery ............................................................... 19
Figure 2.5 Watermarking for PN sequences .............................................. 20
Figure 2.6 Generation of decimal sequence .............................................. 25
Figure 2.7 Decimal sequence watermarking scheme ................................... 27
Figure 2.8 Watermarking the PN sequences .............................................. 27
Figure 2.9 Watermarking in the frequency domain ..................................... 28
Figure 2.10 Image noisification and retrieval for DCT watermarking ............... 29
Figure 3.1 Image noisification scheme ..................................................... 31
Figure 3.2 Watermark ................................................................. 33
Figure 3.3 Digital sequence of the watermark ............................................. 33
Figure 3.4 Image retrieval scheme from the noisified image .......................... 36
Figure 3.5 Augmented watermarking for PN sequences ............................... 38
Figure 3.6 Augmented watermarking for decimal sequences ......................... 39
Figure 3.7 Frequency domain watermarking .............................................. 40
Figure 3.8 Augmented watermarking in spectral domain .............................. 41
Figure 3.9 Image noisification for overlaid augmented watermarking for N stages.... 42
Figure 3.10 Image retrieval for overlaid augmented watermarking for N stages .... 43
Figure 3.11 Overlaid augmented watermarking ............................................ 45
Figure 3.12 Watermarking for Digital Rights Management ............................ 46
Figure 4.1 Different embedded watermarks ............................................... 49
Figure 4.2 Variation of PSNR with the size of the watermark ......................... 49
Figure 4.3 Variation of PSNR with the size of the watermark for D-Sequences ..... 51
Figure 4.4 Variation of PSNR with the size of the watermark for DCT watermarking ................................................................. 52
Figure 4.5 Results after adding external noise ............................................. 53
Abstract:

This thesis provides an augmented watermarking technique wherein noise is based on the watermark added to the watermarked image so that only the end user who has the key for embedding the watermark can both remove the noise and watermark to get a final clear image. The recovery for different values of noise is observed. This system may be implemented as a basic digital rights management system by defining a regime of partial rights using overlaid watermarks, together with respectively added layers of noise, in which the rights of the users define the precision with which the signals may be viewed.
Chapter 1
Introduction

1.1 Digital Rights Management

The old adage that ‘seeing is believing’ is no longer true due to the pervasive and powerful signal processing capabilities that exist now. Digital distribution of multimedia information allows the introduction of flexible, cost-effective business models that are advantageous for commerce transactions. On the other hand, its digital nature also allows individuals to manipulate, duplicate or access media information beyond the terms and conditions agreed upon [2, 5].

Multimedia data such as photos, video or audio clips, printed documents can carry hidden information or may have been manipulated so that one is not sure of the exact data. To deal with the problem of trustworthiness of data, authentication techniques are being developed to verify the information integrity, the alleged source of data, and the reality of data [22]. This distinguishes from other generic message authentication in its unique requirements of integrity.

Multimedia authentication techniques are usually designed based on two kinds of tools: digital signature or watermarking [19]. Digital signature is an 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 multimedia data 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. Given the objective for multimedia authentication to reject the crop-and-replacement process and accept content-preserving or
imperceptible manipulations, traditional digital signature or watermarking method cannot be directly applied to authentication [31]. Traditional digital signature does not allow even a single bit change in the data. On the other hand, traditional watermarking techniques are designed for surviving all kinds of manipulations that may miss a lot of content-altering manipulations. Therefore, there is a need for designing novel robust digital signature or semi-fragile watermarks for multimedia authentication.

Digital rights management (DRM) has been proposed to address these issues [30]. It involves linking specific user rights to media in order to control viewing, duplication, and access among other operations. Ideally, a DRM system balances information protection, usability, and cost to provide a beneficial environment for all parties; this includes expanded functionality, cost effectiveness and new marketing opportunities. Overall management is achieved through the interaction of effective economic models, representations of consumer ethics, legal policy, and technology.

Fig. 1.1 DRM architecture
At the technological level, DRM systems incorporate encryption, copy control, tagging, tracing, conditional access and media identification. The challenge is to engineer secure systems for an environment of dynamic applications and standards in which appropriate business models and consumer expectations are only now being identified [30]. The schematic is as shown in figure 1.1. DRM describes a range of techniques that use information about rights and rights holders to manage copyright material and the terms and conditions on which it is made available to users.

1.1.1 Copy and Copyright Protection

Copy protection attempts to find ways, which limits the access to copyrighted material and/or inhibit the copy process itself. Examples of copy protection include encrypted digital TV broadcast, access controls to copyrighted software through the use of license servers and technical copy protection mechanisms on the media. A recent example is the copy protection mechanism on DVDs. However, copy protection is very difficult to achieve in open systems, as recent incidents (like hacking the DVD encryption) show.

Copyright protection inserts copyright information into the digital object without the loss of quality. Whenever the copyright of a digital object is in question, this information is extracted to identify the rightful owner. It is also possible to encode the identity of the original buyer along with the identity of the copyright holder, which allows tracing of any unauthorized copies. The most prominent way of embedding information in multimedia data is the use of digital watermarking. Whereas copy protection seems to be difficult to implement, copyright protection protocols based on watermarking and strong cryptography are likely to be feasible.

Cryptography, the science of writing in secret codes addresses all of the elements
necessary for secure communication over an insecure channel, namely privacy, confidentiality, key exchange, authentication, and non-repudiation. A limitation of cryptography is that it does not always provide safe communication. Once the presence of encrypted data is detected, there are multiple ways for an attacker to gain access to the data. This lead to the wide usage of steganography, the art of hiding messages.

1.2 Steganography

While the goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party [8]. While most cryptography applications are used to encrypt information so that only the sender and recipient can understand it, steganography hides information that only the sender and recipient know it exist. The word ‘steganography’ can be defined as ‘covered writing’. It is the art of hiding a message, rather than encoding it. If a message is not suspected then it is rather difficult to begin to decode it. It includes a vast array of techniques for hiding messages in a variety of media.

There are two directions in steganography as shown in figure 1.2. One of them is used for protection against detection and the other for protection against removal. As the purpose of steganography is to achieve a covert communication between two parties whose existence is unknown to a possible attacker, a successful attack consists in detecting the existence of this communication (e.g., using statistical analysis of images with and without hidden information). Watermarking, as opposed to steganography, has an additional requirement of robustness against possible attacks.
1.3 Digital Watermarking

Digital watermarking embeds identifying information in an image, which is not always hidden, in such a manner that it cannot easily be removed. It can also contain device control code that prevents illegal recording. An application of watermarking is copyright control, in which an image owner seeks to prevent illegal copying of the image.

Watermarking has been considered to be a promising solution to protect the copyright of multimedia data through transcoding, because the embedded message is always included in the data. There is no evidence that watermarking techniques can achieve the ultimate goal to retrieve the right owner information from the received data after all kinds of content-preserving manipulations [4, 11]. Because of the fidelity constraint, watermarks can only be embedded in a limited space in the multimedia data. There is always a biased advantage for the attacker whose target is only to get rid of the watermarks by exploiting various manipulations in the finite watermarking embedding space. A more reasonable expectation of applying watermarking techniques for copyright protection may be to consider specific application scenarios. The fundamental
requirements of a watermark are

- Imperceptibility: In watermarking, we traditionally seek high fidelity, i.e. the watermarked work must look or sounds like the original. Whether or not this is a good goal is a different discussion.

- Robustness: It is more a property and not a requirement of watermarking. The watermark should be able to survive any reasonable processing inflicted on the carrier (carrier here refers to the content being watermarked).

- Security: The watermarked image should not reveal any clues of the presence of the watermark, with respect to unauthorized detection, or indefectibility or unsuspicious.

- Efficiency: Efficiency is the speed of the algorithm for inserting and detecting the watermark

- Capacity: It is the extent of watermark a cover image that watermark can hold.

Currently, watermarking is used for

- Copyright protection - to prevent third parties from copying or claiming the ownership of the digital media.

- Authentication – e.g. tamper detection and monitoring

- Owner identification

- Broadcast monitoring – to track the broadcast of a particular media file over a channel.

- Medical Applications – used in X-ray film references where they are marked with a unique ID of the patient.

- Fingerprinting - to convey information about the recipient of the digital media (rather than the owner) in order to track distributed copies of the media.
• Image authentication - to check the authenticity of the digital media.

1.4 Types of Watermarks

With images widely available on the web, watermarks could be used to provide authentication in terms of a secondary image which is overlaid on the primary image, and provides a means of protecting the image. This overlay may be visible or invisible.

1.4.1 Visible Watermarks

A visible watermark is a visible translucent image that is overlaid on the primary image. Visible watermarks change the signal altogether such that the watermarked signal is totally different from the actual signal, for example, adding an image as a watermark to another image. Consisting of the logo or seal of the organization allows the primary image to be viewed, but still marks it clearly as the property of the owning organization. The watermark doesn't totally obscure the primary image, but it does identify the owner and prevents the image from being used without that identification attached. It is important to overlay the watermark in a way which makes it difficult to remove, if the goal of indicating property rights is to be achieved.

The example in the figure 1.3 shows both a watermark and an image with the overlaid watermark.

![Fig.1.3 Visible Watermarking](https://example.com/image1.jpg)
1.4.2 Invisible Watermarks

An invisible watermark is an overlaid image which cannot be seen, but which can be detected algorithmically. Invisible watermarks do not change the signal to a perceptually great extent, i.e., there are only minor variations in the output signal. An example of an invisible watermark is when some bits are added to an image modifying only its least significant bits. Different applications of this technology call for two very different types of invisible watermarks:

- A watermark which is destroyed when the image is manipulated digitally in any way may be useful in proving authenticity of an image. If the watermark is still intact, then the image has not been ‘doctored’. If the watermark has been destroyed, then the image has been tampered with. Such a technology might be important, for example, in admitting digital images as evidence in court.

- An invisible watermark which is very resistant to destruction under any image manipulation might be useful in verifying ownership of an image suspected of misappropriation. Digital detection of the watermark would indicate the source of the image.

The example in the figure 1.4 shows the invisibly watermarked image.

![Invisible Watermarking](image.png)
1.5 Watermarking Techniques

There are various spatial and frequency domain techniques used for adding watermarks to and removing them from signals [13]. Purely spatial techniques are not robust to some attacks to the signal like cropping and zooming, whereas most frequency domain techniques and mixed-domain techniques are quite robust to such attacks.

LSB Embedding: LSB encoding is very simple and has been used for a variety of purposes. In this method the least significant bit of every component is replaced by the watermark information bit [23]. This method can store quite some information, but the amount of information that can be embedded is still limited and method is more susceptible to attacks. LSB encoding is shown in figure 1.5.

![Fig. 1.5 LSB encoding](image)

CDMA Spread Spectrum: Code Division Multiple Access (CDMA) is a transmission technique in which the frequency spectrum of a data-signal is spread using a code uncorrelated with that signal and unique to every addressee. It is used in spread spectrum systems to enable multiple-access.

Early experimentation with CDMA demonstrated exceptional robustness with relation to noise and high-level JPEG compression, with flawless recovery of the embedded watermark from the watermarked image. CDMA in the spatial domain has certain limitations.
The main drawback of CDMA is that its message capacity is more limited than similar correlation-based techniques. One reason for this is that watermark recovery drops off quickly at higher message sizes. Good results are obtainable using the small watermark. But the results with the normal-sized watermark were disappointing. Also, the processing time for spatial-domain CDMA watermarking increases exponentially with the increase in the size of the message. In CDMA systems all users transmit in the same bandwidth simultaneously. The systems that follow this concept are spread spectrum systems.

The mechanism for embedding one bit in original media is the most basic element in a data hiding system. Many embedding approaches have been proposed in the literature and there are many ways to classify them. For example, some schemes work with the multimedia signal samples while others work with transformed data. It is found it beneficial to study the existing embedding approaches under noise-free conditions (i.e., directly passing a watermarked media to a detector) and to examine whether knowledge of the original host media will enhance the detection performance, regardless of whether a detector uses such knowledge or not. Many existing embedding approaches would then fall in one of the following two categories.

In all the cases mentioned above, watermark should not be placed in certain regions of the image. Further, the watermark is embedded in the least significant bits of the image, i.e., the LSB embedding and prone to attacks. Cropping is also one of the most important factors that noisify the image. The problem is to find the significant regions in the watermark such that it is not detected and with certain alterations, noticeable. In such case, rather than limiting the watermark encoding to the least significant bits, it is advisable to spread it over a large range of image bits. The data is possibly encoded,
modulated, and/or scaled, is added to the host signal, as shown in figure 1.6.

Fig.1.6 Watermarking the Media

The addition can be performed in a specific domain or on specific features. Considering the embedding of only one bit, the difference between marked signal $I_1$ and the original host signal $I_0$ is a function of $b$, the bit to be embedded, i.e., $I_1 - I_0 = f(b)$. Although it is possible to detect $b$ directly from $I_1$, $I_0$ can be regarded as a major noise source in such detection. Therefore, the knowledge of $I_0$ will enhance detection performance by eliminating the interference. Spread spectrum watermarking is a representative of this category.

Spread spectrum embedding has been demonstrated with excellent robustness and invisibility when the original host media is available in detection. In non-coherent detection, the interference from host signal exists even when there is no subsequent processing or intentional attack. Spreading a watermark throughout the message bits of the image ensures a large measure of security against unintentional or intentional attacks. First of all, the location of the watermark is not obvious. Furthermore, the message bits should be selected in a fashion that ensures severe degradation of the original watermark following any attack on the watermark. Spread Spectrum watermarking is one of the most widely used watermarking methods. Here the data signal is multiplied by a Pseudo
Random Noise code. Due to the relative immunity of spread-spectrum modulation to noise, spread spectrum watermarking is widely preferred.

1.6 Watermarking Attacks

In the following figure, ‘transmission’ refers to the application of any encryption standard to data. While most of these are information lossless, many compression schemes can potentially degrade the quality through irretrievable loss of data. Lossy compression is an operation that usually eliminates perceptually non-salient components of an image. If one has to preserve a watermark in the face of such operation, the watermark must be placed in perceptually significant regions of the data [4].

After reception, an image may endure many common transformations that are broadly categorized as geometric and signal noisifications. The common attacks media would undergo are shown in figure 1.7. Geometric noisifications are specific to image and video including operations such as rotation, translation, scaling and cropping.

Fig.1.7 Common attacks that a Media would undergo

In the above transmission of the signal, the data sent to each user should be secretive and unique. This brings the CDMA systems into existence.
1.7 Watermarking Process

The digital watermark embedding and retrieval is as shown in the figure 1.8 and figure 1.9. In the embedding process, the watermark to be embedded is hidden in the cover object, may be an image, audio or video file and during extraction, watermark is retrieved and removed from the image to obtain the original image.

![Fig. 1.8 Watermark Embedding](image1)

![Fig. 1.9 Watermark Detection](image2)

The characteristics of a watermarking algorithm is normally tied to the application it is designed for.

1.8 Outline of Thesis

The thesis introduces and develops the idea of augmented watermark which introduces additional noise to the original image that is based on the hidden watermark. It considers several problems related to watermarking. It presents an algorithm to noisify the image, after watermarking, based on the watermark and finally retrieval of the original image. The algorithm used is robust and able to work effectively in the presence
of additional noise. This method is proposed as the foundation on which a DRM system may be devised.

Chapter 2 introduces watermarking with PN and decimal sequences and DCT watermarking. In Chapter 3 the algorithm is proposed for noisifying the watermarked image based on the watermark and retrieving the original image back. Chapter 4 analyses the results and compares the spatial and spectral domain image noisification. Chapter 5 gives conclusions and suggestions for future research.
Chapter 2
Spatial and Spectral Domain Watermarking

2.1 Pseudo Random Noise Sequences

A Pseudo-random Noise (PN) sequence is a sequence of binary numbers, e.g. ±1, which appears to be random but is in fact perfectly deterministic. A PN generator is typically made of N cascaded flip-flop circuits and a specially selected feedback arrangement as shown in figure 2.1.

![Fig.2.1 Basic circuit for a PN generator](image)

For example take the initial state to be 100. The obtained output sequence will be 0010111, periodic with period 7. The outputs for different stages of the PN generator are shown in figure 2.2.

<table>
<thead>
<tr>
<th>x₁</th>
<th>x₂</th>
<th>x₃</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 2.2 Outputs for different stages of PN generator
The flip-flop circuits when used in this way is called a shift register since each clock pulse applied to the flip-flops causes the contents of each flip-flop to be shifted to the right. The feedback connections provide the input to the left-most flip-flop. The period of the PN sequence is \(2^n-1\).

Starting with the register in state 001 as shown, the next 7 states are 100, 010, 101, 110, 111, 011, and then 001 again and the states continue to repeat. The output taken from the right-most flip-flop is 1001011 and then repeats. With the three stage shift register shown, the period is 7.

The maximum length of a PN sequence is determined by the length of the register and the configuration of the feedback network. An n bits register can take up to \(2^n\) different combinations of zeros and ones. Since the feedback network performs linear operations, if all the inputs (i.e. the content of the flip-flops) are zero, the output of the feedback network will also be zero. The maximum length of any PN sequence is \(2^n-1\).

This is a 3-stage PN generator running on the clock pulse. After every clock pulse, the contents of the registered are shifted as shown in the figure. The period of the generated sequence depends on the feedback connections [20]. The sequence generated by using the above LSFR with initial state \([0 \ 1 \ 0]\) is \([0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0]\).

2.1.1 Properties of PN Sequences

PN sequences having periods equal to \(2^n-1\) for an n-stage LSFR satisfy the following properties

- **Balance Property:** This property states that in the sequence generated the number of ones is equal to the number of zeros.

- **Run Property:** A run is nothing but a sequence containing a single type of a digit. In general, a sequence of length n will have exactly \(1/2n\).
• **Shift Property:** This property states that for any ML sequence and its cyclically shifted sequences, the agreements and disagreements among them will be approximately equal.

• **Autocorrelation Property:** The autocorrelation of ML sequence is single peaked. The auto correlation of any sequence $S$ can be defined as follows

$$R_{ss}(k) = \frac{1}{N} \left( \sum_{0}^{N-1} S_n * S_{n-k} \right)$$

Where $R_{ss}$ is auto correlation of $S$. $S_{n-k}$ is the cyclic shift by $k$.

$$R_{ss}(k) = -1/(2n-1) \text{ where } k \neq N; \text{ and } RSS(k) = 1 \text{ if } k = n$$

• **Cross-correlation Property:** The cross-correlation property provides a measure of resemblance between two different sequences. Let $\{ 1 \ 1 \ 0 \ = \ N \ a \ a \ a \ a \}$ and denote two different pseudorandom sequences. The cross-correlation of these two sequences is defined as follows

$$R_k(a,b) = \frac{1}{N} \sum_{n=0}^{N-1} a_n * b_{n-k}$$

The two sequences are said to be orthogonal if the cross-correlation between them is equal to zero.

2.1.2 Watermarking using PN Sequences

As PN sequences have good correlation properties, noise like characteristics and resistance to interference, they are used for watermarking. In the watermark, each data bit is represented by a sequence of bits [20]. Of these, a significant portion of bits remain unchanged or without loss, without losing the watermark thoroughly. This confirms the survival of the watermark due to redundancy. Other reasons for using PN sequences as a good watermarking tool are
• PN generator produces periodic random sequences.
• PN sequences are generated by an algorithm that uses an initial seed.
• The generated PN sequence generated is not statically random, but it passes many tests of randomness.
• If the algorithm and seed are not known, it is impractical to predict the sequence.

The method of watermarking using PN sequences is to embed a PN sequences into the data where every PN sequence represents one bit of watermarking information. To extract the watermark, the sequence of marked bits is correlated with known PN sequence. To robustly embed one bit of watermark information with this method the PN sequence length should be much greater than the square of the maximum data values. The embedding and recovery procedures are as follows. The PN sequence generated is used for embedding the data in the cover image. This helps us exploit the correlation properties of the PN sequences

Embedding:

Consider watermark image \( a(x,y) \) as the information bearing data signal and PN sequence \( b(x,y) \) as the spreading signal. The desired modulation is achieved by applying both the watermark image and the PN sequence to a product modulator. The resultant signal \( W(x,y) \) is a pseudorandom noise pattern that is added to the cover image \( I(x,y) \) to produce the resultant watermarked image \( I_w(x,y) \).

Fig. 2.3 Watermark Embedding
Hence

\[ I_{w}(x, y) = k_{1} \times W(x, y) + I(x, y) \]

\[ = a(x, y) \times b(x, y) + I(x, y) \]

Recovery:

To recover the original watermark \( a(x, y) \), the watermarked image \( I_{w}(x, y) \) is multiplied at the receiver again with a pseudonoise sequence which is an exact replica of that used for embedding the data.

The unwanted noise signal can be filtered out during the process of correlation by setting the threshold as mean of correlation, \( \tau \).

\[
\text{Correlator Decision} = \begin{cases} 
0 & \text{if } \tau > \text{mean} \\
1 & \text{if } \tau < \text{mean}
\end{cases}
\]

\[ \text{Threshold } (\tau) = \text{Mean } (\text{Correlation}) \]

The cover image used for the watermarking is a 512 × 512, 8 bit gray scale, bitmap image. The watermark used is a monochrome image. Key and the gain are fixed before the generation of PN sequences. The watermark is then converted to a string of zeroes and ones. A PN sequence of size equal to the original cover image is generated for each of the pixel in the watermark vector. If the pixel in the watermark vector is zero then the PN sequence with appropriate gain is added to the cover image else zeroes are added.

For retrieval of the watermark the PN sequences are generated with the same key as used during the embedding process. The correlation is calculated between the
generated PN sequence matrix and the watermarked image for each of the pixels in the
watermark string and if it exceeds a particular threshold then the watermark is said to be
detected.

The robustness of the watermarked image increases as the gain k increases. But
with the increase in the gain k, there is a reduction in the quality of the final watermarked
image. Therefore, there is a tradeoff between the robustness and the quality of the image.
PN sequences can be added to the cover image either by applying a random shift or
circular shift. The watermarked image and the retrieved watermark are as follows:

Original Image:          Embedded Watermark

Watermarked Image        Recovered Watermark

Fig 2.5 Watermarking for PN sequences
2.2 Decimal Sequences

Decimal sequences are generated when a number is represented in a decimal form in a given base $r$. These sequences may terminate, repeat or be aperiodic. A certain class of decimal sequences of the form $1/q$, $q$ being a prime number exhibit the property wherein the digits spaced half a period apart add up to exactly $r-1$, $r$ being the base in which the number is expressed. Properties of decimal sequences have established an upper bound to the autocorrelation function. The properties of decimal sequences have been presented by Kak [10, 11] and some of the important properties from this are presented here.

2.2.1 Properties of Decimal Sequences

Any positive number as a decimal in the base $r$ can be represented as

$$A_1A_2........A_{s+1}a_1a_2........$$

where, $0 \leq A_i < r$, $0 \leq a_i < r$, not all $A$ and $a$ are zero, and an infinity of the $a_i$ are less than (r-1). There exists a one to one correspondence between the numbers and the decimals, and

$$x = A_1r^s + A_2r^{s-1} + ... + A_{s+1} + \frac{a_1}{r} + \frac{a_1}{r^2} + ...$$

Decimal sequences can be used for rational and irrational numbers to generate pseudorandom noise sequences [20]. Following are some theorems on decimal sequences:

Theorem 1: If $q$ is prime and $r$ is a primitive root of $q$, then the decimal sequence for $1/q$ is termed as maximal length decimal sequence in the base $r$.

The string of their first $q-1$ digits often represents maximal length sequences. It can be clearly observed that for every prime $q$, there exists $\phi(q-1)$ maximal length sequences in different scales.
Theorem 2: A maximal length decimal sequence \( \{1/q\} \), when multiplied by \( p, p<q \), is a cyclic permutation of itself.

Proof: The remainders 1, 2, …, \( q-1 \) obtained during the division of 1 by \( q \) map into the coefficients 0, 1, …, \( r-1 \). Since \( p/q \) starts off with a remainder \( rp \) (modulo \( q \)) instead of \( r \) (modulo \( q \)), there would be a corresponding shift of the decimal sequence.

Example: Consider \( x = \{1/7\} \). The corresponding decimal sequence for \( x \) in base 0 is maximal length because \( 10^2 \neq 1 \) (modulo 7), \( 10^3 \neq 1 \) (modulo 7). But \( 10^6 \equiv 1 \) (modulo 7).

The decimal sequence is 1 4 2 8 5 7, which corresponds to the remainder sequence 3 2 6 4 5 1. This 3, 3^2, 3^3, 3^4, 3^5, 3^6 all computed modulo 7 yield the successive digits of the sequence. Now if \( x = \{3/7\} \), the remainder sequence starts with 30 \( \equiv 2 \) (modulo 7) and in fact is 2 6 4 5 1 3, and therefore the decimal sequence for 3/7 would be 4 2 8 5 7 1. This suggests that the structure of the remainder sequence must also show in the decimal sequence.

Theorem 3: If the decimal sequence in base \( r \) of \( p/q; (p, q) = 1, p<q \), and \( (r, p) = 1 \) is shifted to the left in a cyclic manner \( l \) times, the resulting sequence corresponds to the number \( p'/q \), \( (p', q) = 1, p' < q \) where \( p' \equiv r' \times p \) (modulo \( q \)).

Theorem 4: For a maximum length sequence \( \left( \frac{1}{q} \right) = a_1a_2...a_k, k = q-1 \), in base \( r \):

\[ a_i + a_{(i+k-1)} = r - 1 \]

Example: let \( x = \{1/17\} \) in base 10

The Decimal sequence for \( x \) is 0 5 8 8 2 3 5 2 9 4 1 1 7 6 4 7

Note that \( a_i + a_{8+i} = r - 1 = 9 \)
Similarly if \( x = \{1/19\} \) in base 2. The decimal sequence for \( x \) is \( 0 0 0 0 1 1 0 1 0 1 1 1 0 0 1 0 1 \). Note that \( a_i + a_{q+i} = r - 1 = 1 \)

Theorem 5: The hamming distance \( d_j \) between the binary maximal length sequence \( \{1/q\} \) and its \( j \)'th cyclic shift satisfies

\[
d_j \geq k/m, \ j \neq 0, \ j < k
\]

Where \( 2^m > q, \ k = q-1 \).

From this theorem, it may be stated that at least one of the \( m \) consecutive digits is going to be different. Hence the hamming distance between each set of \( m \) digits is one. Thus if \( k \) such groups are considered, then the distance is \( k \), and since the sequence considered is \( m \) times over, the distance is \( k/m \).

- **Autocorrelation Property**

For a symmetric binary decimal sequence, the autocorrelation \( R_x(j) \leq 1 - 2/m, \ j \neq 0, \ j < k \). Thus a lower bound exists on the distance between a sequence and its cyclic shifts. For a normal number, the autocorrelation function is defined as

\[
R_x(\tau) = E(a_n, a_{n+\tau})
\]

where the \( n \)'th digit of the sequence \( a_n \in \{0, 1, 2, \ldots, r-1\} \). Since each of the digits occur with a frequency \( 1/r \), \( R_x(0) = E(a_n^2) = (r-1)(2r-1)/6 \). Also for such a number, the successive sequence of digits are independent and therefore

\[
R_x(\tau) = E(a_n, a_{n+\tau}) = E(a_n)E(a_{n+\tau}) = (r-1)^2 / 4
\]

The autocorrelation function is two valued if the digits from zero to \((r-1)\) are mapped symmetrically about zero by the transformation \( a_i = 2a_i - (r-1) \). A straightforward calculation shows that
\[
R_x(\tau) = \begin{cases} 
\frac{(r^2 - 1)}{3} & \tau = 0; \\
0 & \text{otherwise.}
\end{cases}
\]

- **Cross-Correlation Property**

Let \( R_{xy}(\tau) = \frac{1}{N} \sum_{i=1}^{N} a_i b_{i+\tau} \) represent the cross-correlation function of two maximal length sequences \( \{x\} = a_1 a_2 ... a_k \) and \( \{y\} = b_1 b_2 ... b_k \). The period of the product sequence \( a_i b_{i+\tau} \) is \( N = \text{LCM} (k_1, k_2) \), where LCM is the least common multiple.

Theorem 6: The cross-correlation function of two maximal length sequences in the symmetric form is identically equal to zero if the ratio \( k_1/k_2 \) of their periods reduces to an irreducible fraction \( n_1/n_2 \) where either \( n_1 \) or \( n_2 \) is an even number.

**2.2.2 Generating the Decimal Sequences**

Decimal sequences can be generated using feedback shift registers that allow carry. These sequences can also be generated by using a computational device by using the following equations [12]:

\[
a' = \lfloor l(r^i \mod q) \rfloor \mod r
\]

\[
q \mod r \equiv -k \equiv -1/l
\]

The hardware similar to the one used in generation of maximal length PN sequences can be used for the generation of decimal sequence. The algorithm used for the generation is called the Tirtha algorithm, which may be used whenever the prime number \( q \) is given in terms of radix \( r \) as \( q = tr - 1 \), where \( t \) is an integer.

Theorem: Let \( 1/(tr - 1) \) define the decimal sequence \( a_1 a_2 a_3 ... a_k \), where \( r \) is the radix. Consider another sequence \( u_1 u_2 u_3 ... u_k \), where, for all \( i \), \( u_1 < t \), then
\[ru_i + a_i = u_{i+1} + ta_{i+1}\]

Proof: Since the sequence repeats itself, \(a_k = 1\) and \(u_k = 0\). The remainder in the long division of 1 by \((tr-1)\) is therefore \(t\). The quotient \(a_{k-1}\) is given by

\[a_{k-1}(tr-1) + t = m_{i-1}r\]

This makes \(a_{k-1} = t\), extending the argument to the \(a\) and \(u\) sequences, when written in inverse as

\[u_ku_{k-1}u_{k-2}....1\]

\[a_k a_{k-1} a_{k-2} .... 1\]

which is equal to \(0 0 .... 1\)

\[1 \quad t \quad [t^2] \mod r \quad .... \quad 0\]

The circuit for the generation of decimal sequences is given in fig 2.6. It consists of \(n\) stages of shift registers. The carries that are added to the immediate preceding stages are represented by Cs.

![Fig. 2.6 Generation of decimal sequence.](image)
When the carry is generated by the extreme left stage, it is introduced into this stage at the very next clock pulse. The sequence generated will be in the inverse order. The same principle can be used to generate binary decimal sequences. The number of stages needed for the generation of binary decimal sequence for type $1/q$ is $\log_2 q$. The algorithm also works for the non-binary sequences of the type $1/(tr-1)$ when the given fraction is multiplied by an appropriate integer so that the standard form can be used.

2.2.3 Watermarking using Decimal Sequences

PN sequence based watermarking produces noise due to high autocorrelation values as the period of the PN sequences is too large compared to the size of the cover image. As d-sequences have zero cross correlation for some prime numbers, superior performance is obtained if different d-sequences are used in the watermark [20]. Using the decimal sequences also has the flexibility of trying out various prime numbers until satisfactory results of embedding and recovery is obtained. The function for generating the decimal sequences is

$$dseq = \left[r^i \mod q\right] \mod r$$

where $r$ is the radix and $q$ is the base of the prime number.

Decimal sequences are watermarked in a similar manner as that of the PN sequences by spread spectrum watermarking. The d-sequences are added to the cover image either by a circular shift or a random shift. The decimal sequence spread-spectrum watermarking scheme is shown in figure 2.7. The prime $q$ drives the decimal sequence (d-sequence) generator, produces the chip sequence $u$, which has zero mean and whose elements are equal of $-\sigma_u$ or $+\sigma_u$. The chip sequence $u$ is either added or subtracted from the signal $x$ depending on the value of the watermark bit $b$, which takes values $\{+1, -1\}$. The signal $s$ is the watermarked signal and $n$ is the noise introduced into the system.
Fig 2.7 Decimal sequence watermarking scheme

The watermarked image and the retrieved watermark are as follows:

![Original Image](image1.png) ![Embedded Watermark](image2.png)

Original Image  Embedded Watermark

![Watermarked Image](image3.png) ![Recovered Watermark](image4.png)

Watermarked Image  Recovered Watermark

Fig 2.8 Watermarking the PN sequences
2.3 Spectral Domain Watermarking

The general procedure for frequency domain watermarking is shown in figure 2.9. Upon applying a frequency transformation to the data, a perceptual mask is computer that highlights perceptually significant regions in the spectrum that can support the watermark without affecting perceptual fidelity. The watermark signal is then inserted into these regions.

![Fig.2.9 Watermarking in the frequency domain](image)

In principle, any frequency domain transform can be used. The transform used here is the discrete cosine transform (DCT). In order to place a watermark of length n into an \( N \times N \) image, \( X(i,j) \), the \( N \times N \) DCT of the image is computed and the watermark \( w(k) \) is embedded into that using a key. Then the IDCT of the image is computed to give the watermarked image, \( \hat{X}(i,j) \).

![Original Image](image)  ![Embedded Watermark](image)
The watermarked image and the retrieved watermark are as shown in Fig 2.10.

2.4 Invertible Watermarking

Invertible watermarking has been introduced for the purpose of image authentication to establish their integrity. Another application of invertible watermarking is multilevel access to watermarked images. Both applications make are used in precision-critical images (e.g. military, satellite, medical, quality control, reverse engineering images) whose copyright should be protected to some extent. While most watermarking schemes introduce some small amount of non-invertible noisification in the image, invertible watermarking methods are such that, if the watermarked contents are deemed authentic, the noisification due to watermarking can be removed to obtain the original contents.

2.4.1 Invertible Spread Spectrum Watermarking

In chapter 1, it is explained how the spread-spectrum technique is used to obtain an oblivious watermarking method in the spatial domain. Oblivious watermarking does not require the original image to recover the watermark embedded in the watermarked image.
Hence invertible watermarking technique [6] is used. Applications of Invertible spread spectrum watermarking are as follows.

- Authentication is the main application of reversible data hiding
- It can be used in some special applications, such as law enforcement and medical fields, where original cover media is required for legal reasons.
- It can be used in military, remote sensing and high energy physics experiments, where high accuracy is required, or data acquisition is expensive.
- Embedding data into cover media while keeping the media reversible opens a new door for linking some data with original media. (e.g. Medical data system).

To make the above watermarking scheme totally invertible, the following three conditions must be met:

1. The seeds used to generate the pseudo noise signal must be known. Recreating the pseudo-noise signal is needed to recover the embedded bits.
2. The locally adjustable amplitude factor used at each sample of the watermarked signal during the embedding phase must be known. This requirement can be easily met by using a constant value $\alpha$ for all samples.

2.4.2 Algorithm for Spread Spectrum Watermarking Inversion

1. All the embedded bits are recovered. The embedded watermark is $a(x,y) \in \{0,1\}$ and the spreading signal is $b(x,y)$.

2. These bits are removed from the watermarked image as

$$I(x,y) = I_w(x,y) - k_1 \times W(x,y)$$

$$= I_w(x,y) - a(x,y) \times b(x,y)$$
Chapter 3

Augmented Watermarking

3.1 Introduction

This chapter introduces the method of augmented watermarking where noise is added to the watermarked image giving appropriate rights to the authorized user to view the original image. The property that the watermark will not change after retrieving from the watermarked image even in the presence of noise is used in noisifying the image and retrieving it back. Initially the image is watermarked and then noise is added, based on the watermark. As the watermark is recovered perfectly, the noise which is based on the watermark can be removed from the watermarked image. Now the watermark is removed to obtain the original image.

3.2 Image Noisification Scheme

The image noisification scheme is as shown in the figure below.
The image taken is a 512 × 512 Lena. Initially the image is watermarked with the required watermark. The watermarking scheme used here is the spread spectrum watermarking. Noise is then added to the watermarked image so that the picture quality is far reduced. This noise is based on the watermark, that is, the key with which the noise is added to the watermarked image is retrieved from the watermark itself.

Watermark \( a(x,y) \) is embedded into embedding into the cover image is done according to the equation with the PN or Decimal sequence \( b(x,y) \) as the spreading signal. Let the key for generating the PN sequence be \( key1 \). The resultant signal \( W(x,y) \) is a pseudorandom noise pattern that is added to the cover image \( I(x,y) \) to produce the resultant watermarked image \( I_w(x,y) \).

Hence

\[
I_w(x,y) = k_1 \times W(x,y) + I(x,y) \\
= a(x,y) \times b(x,y) + I(x,y)
\]

where

- \( I_w(x,y) \) denotes the watermarked image.
- \( I(x,y) \) denotes the actual cover image.
- \( W(x, y) \) denotes a pseudorandom noise pattern that is added to the image.
- \( k_1 \) denotes the gain factor with which the sequence is embedded

**3.2.1 Key for Noise Embedding**

The watermark is embedded into the image using the spread spectrum watermarking. Let the key for embedding the noise sequence into the watermarked image be \( key2 \). Now based on the watermark, the noise is added to the image. For this the watermark is resized and split into bits of ‘t’. Each ‘t’ bit string is converted into a
decimal number and all the elements are stored in an array which make the key. Using the key, noise is added to the watermark.

Consider the watermark to be an m × n image. The elements in the image are read as a m × n matrix. For example, consider the following 9 × 12 image to be the watermark.

![Watermark Image](image)

Fig 3.2 Watermark

The image is read in the from of a matrix as

![Matrix Image](image)

Fig 3.3 Digital sequence of the watermark

The image is read in the from of a matrix as

Now these elements are split into bits of ‘t’ each. The number sequence so that the total number of elements are divisible by ‘t’. For that, the number sequence is appended by 0 s, if needed, to reach the limit that the total number are divisible by ‘t’.
If the value of ‘t’ is taken to be 5, then the above sequence can be split as

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

Here the number of elements in equation is 9 × 12 = 108. But 108 is not divisible by 5. Hence to make the number divisible by 5, 0s are appended to the above sequence to make up to a number which is divisible by 5.

\[
\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

Each subsequence is converted to a decimal number and all the numbers are stored in an array, which is the key for embedding the noise.
The obtained array is used as the key for embedding noise into the image.
\[ key_2 = 31\ 31\ 16\ 14\ 31\ 13\ 30\ 27\ 29\ 31\ 31\ 19\ 30\ 27\ 13\ 22\ 27\ 13\ 31\ 7\ 31\ 28 \]

### 3.2.2 Noise Embedding Method Using the Key

The same spread spectrum embedding is used here for noisifying the image but with a high gain. The key for adding the noise is dependent on the watermark itself. Noise is added to the watermarked image as

\[
I_{ww}(x, y) = k_{11} \times W(x, y) + I_w(x, y)
\]

\[= k_{11} \times W(x, y) + (k \times W(x, y) + I(x, y))\]

\[= k_{11} \times W(x, y) + (a(x, y) \times b(x, y) + I(x, y))\]

where,

- \(I_w(x, y)\) denotes the watermarked image.
- \(a(x,y)\) denotes embedded watermark
- \(b(x,y)\) denotes the spreading signal.
- \(W(x, y)\) denotes a pseudorandom noise pattern that is added to the image.
- \(k_i\) denotes the gain factor with which the watermark is embedded into the image
- \(I_{ww}(x, y)\) denotes the image after adding noise.
- \(k_{11}\) denotes the gain factor with which the noise is embedded into the watermarked image

### 3.3 Image Retrieval Scheme

At the receiver end, the end user is provided with the key, \(k_1\) which is used for watermarking the image. Using the key, the receiver calculates the watermark. The key for adding the noise is obtained from the watermark using the same algorithm by which it is calculated at the sender’s end for adding the noise. Then using the calculated key \(k_{11}\), the noise is removed from the image. This leaves the receiver with the watermarked image. Using the key which is used for embedding, the watermark is removed from the
image to obtain the original image.

Fig. 3.4 Image retrieval scheme from the noisified image

3.4 Noise Removal

Noise is removed from the final noisified image by performing the reverse operation of the reverse noise addition algorithm. These bits are removed from the watermarked image as

\[ I_w(x, y) = I_{ww}(x, y) - k_{11} \times W(x, y) \]

\[ = I_{ww}(x, y) - a(x, y) \times b(x, y) \]

After removing the noise, watermark is removed from the image using the inverse spread spectrum technique i.e the reverse procedure for embedding is followed.

\[ I(x, y) = I_w(x, y) - k_1 \times W(x, y) \]

\[ = I_w(x, y) - a(x, y) \times b(x, y) \]
3.5 Implementation in Different Domains

3.5.1 Image Noisification and Retrieval for PN Sequences

The key for embedding the noise, key2 is generated from the watermark. A PN sequence which is generated using key2 with high gain is embedded as noise into the watermarked image.

Original Cover Image

Embedded Watermark

Image after watermarking

Image after adding noise
The images for the Augmented Watermarking for PN sequences is as shown in figure 3.5. Here the used gain for embedding watermark into the image, \( k_1 = 2 \) and gain for adding noise to the image, \( k_{11} = 50 \).

### 3.5.2 Image Noisification and Retrieval for Decimal Sequences

Image noisification for decimal sequences is implemented both with PN sequences and decimal sequences as noise signals. The figure 3.6 shows the augmented watermarking having the decimal sequence with a high gain as noise. Here the used gain for embedding watermark into the image, \( k_1 = 2 \) and gain for adding noise to the image, \( k_{11} = 50 \).
3.5.3 Image Noisification and Retrieval for DCT Watermarking

In this process, instead of watermarking to the entire image, watermark is embedded into certain portion of the image. After embedding the watermark into the cover image, noise is added to the watermarked portion and then attached again to the left portion of the cover image. Inverse transform is applied to obtain the watermarked image. The final user who has the key, which is used to embed noise, will remove the watermark and thus the key, which is used to embed noise into the image. The general frequency domain watermarking is shown in figure 3.7.
Consider the case of the DCT domain. On applying the discrete cosine transform to the original image to be $I(x,y)$,

$$I_{dc}(x,y) = DCT(I(x,y))$$

The image after watermarking is

$$I_{w}(x,y) = I_{dc}(x,y) + k_{11} \times W(x,y)$$

Noise is added to the watermark at this stage. This noise is retrieved from the watermark as shown in the previous methods.

$$I_{ww}(x,y) = I(x,y) + k_{11} \times W(x,y)$$

At the receiver end, initially noise is removed from the noisy image and then the watermark. The successive equations are

$$I_{w}(x,y) = I_{ww}(x,y) - k_{11} \times W(x,y)$$

$$I_{dc}(x,y) = I_{w}(x,y) - k_{1} \times W(x,y)$$
Here the used gain for embedding watermark into the image, $k_1 = 2$ and gain for adding noise to the image, $k_{11} = 50$.

### 3.6 Overlaid Augmented Watermarking

The method of augmented watermarking can become a foundation for the design of a DRM system. We may consider multiple watermarks overlaid one over the other so that the clarity of the image seen by the end user depends on the digital rights of a person. Embedding different watermarks:
Initially the original image is watermarked, then it is noisified using the first watermark. This noisified image is watermarked and then noisified with the second watermark. This process continues as layered or overlaid watermarking for N watermarks as shown in figure 3.9.

![Image](image.png)

**Fig. 3.9 Image noisification for overlaid augmented watermarking for N Stages**

Consider the example of an image overlaid augmented watermarked with three images $a_1(x,y)$, $a_2(x,y)$ and $a_3(x,y)$. Let the spreading sequences be $b_1(x,y)$, $b_2(x,y)$ and $b_3(x,y)$. Assume that if a user A has partial rights, he can only remove one of the watermarks obtaining an image that is clearer compared to that of the final watermarked image but much noisified one when compared to that of the original image.

The multiple watermarking schemes can be portrayed as below. Let $k_1$, $k_2$ and $k_3$ be the gains with which the three watermarks are embedded and $k_{11}$, $k_{22}$ and $k_{33}$.

\[
I_{w_1}(x,y) = k_1 \times W_1(x,y) + I(x,y)
\]

\[
I_{ww_1}(x,y) = k_{11} \times W_1(x,y) + I_{w_1}(x,y)
\]

\[
I_{w_2}(x,y) = k_2 \times W_2(x,y) + I_{ww_1}(x,y)
\]

\[
I_{ww_2}(x,y) = k_{22} \times W_2(x,y) + I_{w_2}(x,y)
\]

\[
I_{w_3}(x,y) = k_3 \times W_3(x,y) + I_{ww_2}(x,y)
\]

\[
I_{ww_3}(x,y) = k_{33} \times W_3(x,y) + I_{w_3}(x,y)
\]

$I(x,y)$ denotes the original image

$I_{w_1}(x,y)$ denotes the image embedding the first watermark

$I_{ww_1}(x,y)$ denotes the image after noisifying it with the first watermark

$I_{w_2}(x,y)$ denotes the image embedding the second watermark
$I_{ww2}(x,y)$ denotes the image after noisifying it with the second watermark

$I_{w3}(x,y)$ denotes the image embedding the third watermark

$I_{ww3}(x,y)$ denotes the image after noisifying it with the third watermark

Retrieving the Original Image:

The final image which is watermarked and noisified by N watermarks is sent to the end user. Depending upon the rights of the user, permissions are giving to remove the watermarks and the noise induced based on them. Initially the top most i.e., the last embedded watermark is removed as can be mentioned as the reverse procedure of embedding the watermarks as shown in figure 3.10.

\[
egin{align*}
I_{w3}(x,y) &= I_{ww3}(x,y) - k_{33} \times W_3(x,y) \\
I_{ww2}(x,y) &= I_{w3}(x,y) - k_3 \times W_3(x,y) \\
I_{w2}(x,y) &= I_{ww2}(x,y) - k_{22} \times W_2(x,y) \\
I_{ww1}(x,y) &= I_{w2}(x,y) - k_2 \times W_2(x,y) \\
I_{w1}(x,y) &= I_{ww1}(x,y) - k_{11} \times W_1(x,y) \\
I(x,y) &= I_{w1}(x,y) - k_1 \times W_1(x,y)
\end{align*}
\]

The watermark which is embedded last is recovered first as it is on the top layer.

Proceeding this way, the watermark embedded first is recovered last. The results for the overlaid watermarking system for a 512 × 512 for two watermarks are shown in figure 3.11. The values of gain used here are $k_{i}=2$ and $k_{ij}=20$. 

![Fig. 3.10 Image retrieval for overlaid augmented watermarking for N stages](image-url)
Original Image

Image after Noisifying with the First Watermark

Embedded First Watermark

Image after Noisifying with the Second Watermark

Embedded Second Watermark
Some differences between Overlaid Augmented and Multiple Watermarking:

- At each stage of recovery, particular watermarks are recovered but not all the watermarks as it depends on the key distribution. This paves the way for the extension of the overlaid augmented watermarking to digital rights management.

- Noise is added after embedding each watermark in overlaid watermarking whereas it is added at the final stage in multiple watermarking.

3.7 Extension to Digital Rights Management

Some applications of the augmented watermarking method could be the authorization for Video (since it is a stream of images) and Web. Augmented watermarking technique will have to be used in conjunction with an appropriate protocol.
which is initiated each time the data is updated. Figure 3.12 shows the extension of augmented watermarking to digital rights management.

This is more like public key cryptography where in there is a public key known to a specific set of users. The original cover image \( I(x,y) \) is watermarked using the watermark, \( W(x,y) \) the image, using \( key1 \) to obtain the watermarked image \( I_w(x,y) \). The substring algorithm is run on the watermark to obtain the key for embedding the noise i.e., \( key2 \).

\[
key2 = \text{substring}(W(x,y))
\]

Noise is added to the watermarked image \( I_w(x,y) \) with the help of \( key2 \) to obtain the noisy image, \( I_{ww}(x,y) \). Data set is sent to all the users in the network. The Certification Authority verifies the authorization of the user. Once the authorization is established, the users are provided with the \( key1 \) that is used for watermarking.

The protocol for dewatermarking is initiated thus the \( key2 \) for removing the noise is calculated. With this \( key2 \), the noise is removed using inverse noise removal algorithm.
This leaves the user with the watermarked data. With key1, the user removes the watermark using the inverse watermarking algorithm. These are run by each individual separately to get the original set of data.

3.7.1 Protocol for Obtaining the Original Data

A protocol for watermarking for the Digital Rights Management irrespective of the type of domain and method of watermarking is presented below. The user is provided with the key for embedding the watermark, key1. The steps for denoising and dewatermarking are as follows.

1. Watermark, \( W(x,y) \) is recovered using key1 from the noisy image \( I_{ww}(x,y) \) as the addition of noise doesn’t affect the recovered watermark.
2. Run the substring algorithm the watermark to obtain the key for embedding the noise i.e. key, which is calculated by performing operations on the watermark.

\[
key2 = \text{substring}(W(x,y))
\]

3. Remove the noise from the watermarked image \( I_{ww}(x,y) \) with the help of key2 to obtain the watermarked image without noise \( I_w(x,y) \) using the noise removal algorithm.
4. Now remove the watermark from \( I_w(x,y) \) using the inverse watermarking algorithm to obtain the original image \( I(x,y) \).
Chapter 4
Analysis of Results

4.1 Analysis of Gain

In chapter 3 we described our augmented watermarking scheme. As the length N of the image is considerably large the effect of the signal energy on the watermarking scheme is almost negligible. It is observed that the value of gain for embedding the watermark, $k_1$ and the value of gain for adding the noise, $k_{11}$, should be in the ratio of the length of the substring used to calculate the noise key. As the main aim of this thesis is to noisify the image, gains in the range 0.2 to 2 are used. For simulations where heavy noise is added to noisify the image to the maximum extent, gains in the range 2 to 20 are used. The watermark should be small for the spread spectrum watermarking.

The peak signal to noise ratio (PSNR) is one of the most important criteria in determining the extent of noisification. The correlation values for the original and the recovered images are different for the values of different gains where $k_1$ is the gain for watermarking and $k_{11}$ is the gain for embedding the noise.

4.2 Analysis of Augmented Watermarking on PN Sequences

Considering the values of PSNR between the original image and the noisified image and correlation value between the original image and the final recovered image and keeping the gain values constant for a specific image, the values are obtained as follows for PN Sequences. The noise used here is the PN Sequence embedded with a high gain. Table 4.1 shows that even at high noisification for gain values of 20 and 50, correlation values for almost all the images are nearly 1.0 which shows that we were able
to recover perfect image back. The values are noted for different watermarks as shown in figure 4.1.

Table 4.1 Correlation and PSNR values for various watermarks for PN sequences for lower gains

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark Image</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>$k_1$, $k_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 1</td>
<td>1</td>
<td>146.3432</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 2</td>
<td>2</td>
<td>64.9325</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 3</td>
<td>3</td>
<td>28.5901</td>
<td>0.9930</td>
<td>2,5</td>
</tr>
</tbody>
</table>

Watermark 1 (13 x 7).  Watermark 2 (17 x 15).  Watermark 3 (26 x 22).

Figure 4.1 Different embedded watermarks

At constant gain values of $k_1 = 20$ and $k_{11} = 50$, the variation of the size of the watermark to the PSNR of original image and the noisified image

Fig.4.2 Variation of PSNR with the size of the watermark

For high values of gain, where $k_1 = 20$ and $k_{11} = 50$, the values are as shown in Table 4.2.
Table 4.2 Correlation and PSNR values for various watermarks for PN sequences for higher gains

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark Image</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>k1, k11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>1</td>
<td>1.6879</td>
<td>1.000</td>
<td>20,50</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>0.6177</td>
<td>1.000</td>
<td>20,50</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>0.3298</td>
<td>0.9630</td>
<td>20,50</td>
</tr>
</tbody>
</table>

Considering another image, Baboon, the values obtained are almost the same which can be observed from Table 4.3.

For high values of gain, where $k_1 = 20$ and $k_{11} = 50$, the values are as shown in Table 4.2.

Table 4.3 Correlation and PSNR values for various watermarks for PN sequences for different values of gain

<table>
<thead>
<tr>
<th>Baboon</th>
<th>Watermark Image</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>k1, k11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>1</td>
<td>151.8969</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Baboon</td>
<td>2</td>
<td>53.5520</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Baboon</td>
<td>3</td>
<td>28.5926</td>
<td>0.9764</td>
<td>2,5</td>
</tr>
<tr>
<td>Baboon</td>
<td>1</td>
<td>1.5190</td>
<td>1.000</td>
<td>20,50</td>
</tr>
<tr>
<td>Baboon</td>
<td>2</td>
<td>0.5355</td>
<td>1.000</td>
<td>20,50</td>
</tr>
<tr>
<td>Baboon</td>
<td>3</td>
<td>0.2860</td>
<td>0.9764</td>
<td>20,50</td>
</tr>
</tbody>
</table>

4.3 Analysis of Augmented Watermarking on Decimal Sequences

For the decimal sequence image noisification, the PSNR and the correlation values are as follows. The noise used here is the decimal sequence with high gain.
Table 4.4 Correlation and PSNR values for various watermarks for D-sequences

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>$k_1, k_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 1</td>
<td>1</td>
<td>57.1748</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 2</td>
<td>2</td>
<td>4.6927</td>
<td>1.000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 3</td>
<td>3</td>
<td>3.0265</td>
<td>0.9616</td>
<td>2,5</td>
</tr>
</tbody>
</table>

At constant gain values of $k_1 = 20$ and $k_{11} = 50$, the variation of the size of the watermark to the PSNR of original image and the noisified image.

Fig. 4.3 Variation of PSNR with the size of the watermark for D-sequences

4.4 Analysis of Augmented Watermarking in the Frequency Domain

For the frequency domain image noisification, the PSNR and the correlation values are as follows. The noise used here is the decimal sequence with high gain.

4.5 Correlation and PSNR values for various watermarks for DCT watermarking

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark Image</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>$k_1, k_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 1</td>
<td>1</td>
<td>168.7804</td>
<td>1.0000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 2</td>
<td>2</td>
<td>61.7646</td>
<td>1.0000</td>
<td>2,5</td>
</tr>
<tr>
<td>Lena 3</td>
<td>3</td>
<td>32.9775</td>
<td>0.9616</td>
<td>2,5</td>
</tr>
</tbody>
</table>
At constant gain values of $k_1 = 20$ and $k_{11} = 50$, the variation of the size of the watermark to the PSNR of original image and the noisified image.

![Graph showing variation of PSNR with the size of the watermark for DCT watermarking](image1)

**Fig. 4.4** Variation of PSNR with the size of the watermark for DCT watermarking

### 4.5 Robustness to External Noise

Here external noise is added to the augmented watermarked image. Gaussian and salt & pepper noise samples were considered. The image after adding the external noise is as shown below.

![Original Image](image2) ![Embedded Watermark](image3)
Augmented watermarked Image

Recovered Watermark

Augmented watermarked Image after adding Gaussian noise

Recovered Watermark

Retrieved Image

Fig. 4.5 Results after adding external noise
The image after removing the added noise as well as watermark added by the proposed scheme leaves us with the image with external noise. This external noise may be reduced by using standard techniques of filtering and smoothing.
Chapter 5
Conclusions

This thesis provides an augmented watermarking scheme for signal protection in a transmission channel wherein the sent signal is noisified intentionally so that only the authorized party can have access to original signal with full precision whereas other parties can receive only degraded versions of the signal. This system may be implemented as a basic digital rights management system by defining a regime of partial rights using overlaid watermarks, together with respectively added layers of noise, in which the rights of the users define the precision with which the signals may be viewed.

Based on the analysis of results, our main conclusions are as follows.

• The decryption depends on the proper choosing of the gains for embedding the image and the noise i.e. $k_1$ and $k_{II}$.

• The image is recovered almost perfectly even in the presence of high noise where the values of gain for embedding the watermark, $k_I$ and the gain for embedding the noise, $k_{II}$ are 20 and 100.

• This watermarking scheme may be effectively implemented can be implemented both in the spatial and spectral domains.

• This watermarking scheme is obtained correctly even in the presence of external noise, so long as the variance of this noise is less than 0.6.

Future extensions of this method should take up the following problem for study:

• Use of augmented watermarking in compression schemes.

• Application of augmented watermarking to audio.
Bibliography


31. http://www.dlib.org/dlib/june01/iannella/06iannella.html
Vita

Kiranmayi Penumarthi was born in Andhra Pradesh, on 30\textsuperscript{th} August 1981, India. She finished her high school education with distinction from Bala Bhanu Vidyalayam in Srikakulam in 1996. She joined Sarada Institute of Science, Technology and Management, affiliated to Jawaharlal Nehru Technological University, Hyderabad, in 1998 for a degree in Bachelor of Technology in the Department of Electronics and Communications Engineering. She graduated with distinction in April 2002. After her graduation, she came to United States to pursue master’s degree and joined the graduate program at Louisiana State University, Baton Rouge, in January 2004. She is a candidate for the degree of Master of Science in Electrical Engineering to be awarded at the commencement of Fall 2005.