2005

Improved watermarking scheme using decimal sequences

Naveed Shaik Ashfaq
Louisiana State University and Agricultural and Mechanical College, ashai1@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/1577
IMPROVED WATERMARKING SCHEME USING DECIMAL SEQUENCES

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
Shaik Ashfaq Naveed
Bachelor of Engineering, Osmania University, 2003
May 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 thesis advisory committee. I am grateful to Mr. Quinn Rainwater for all his support and encouragement throughout my stay at LSU. I also thank my friends Seema, Lavanya and Priya for their help and encouragement.

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 ................................................................................ ix

Chapter 1 Introduction ............................................................. 1
  1.1 Background ..................................................................... 2
  1.2 Steganography and Watermarking .................................... 3
  1.3 Applications of Watermarking .......................................... 5
  1.4 Outline of Thesis .......................................................... 7

Chapter 2 Literature Survey ...................................................... 9
  2.1 Watermarking Attacks ..................................................... 9
  2.2 Basics of Spread Spectrum Watermarking ......................... 9
  2.3 Watermarking Schemes .................................................. 11

Chapter 3 Random Sequences and Properties ............................ 13
  3.1 Pseudorandom Noise Sequences (PN Sequences) ............... 13
    3.1.1 Properties of PN Sequences ...................................... 14
  3.2 Elliptic Curve Pseudorandom Sequences ......................... 15
    3.2.1 Elliptic Curves over $F_{2^n}$ ..................................... 16
    3.2.2 Construction of Pseudorandom Sequences from Elliptic Curves... 17
  3.3 Logistic Map Pseudorandom Sequences ............................. 19
    3.3.1 Properties of Logistic Map Numbers .......................... 19
    3.3.2 Generation of Random Sequences from Logistic Maps ....... 20
  3.4 Decimal Sequences ........................................................ 22
    3.4.1 Properties of Decimal Sequences .............................. 22
    3.4.2 Generation of Decimal Sequences ............................. 26

Chapter 4 Decimal Sequence Image Watermarking ....................... 29
  4.1 Improvements to Traditional Watermarking Scheme ............ 31
  4.2 Gain Analysis of Decimal Sequence Watermarking Scheme .... 40
  4.3 Implementation of Improved Scheme in Audio Watermarking .. 42
  4.4 Multiple Watermark Embedding Using Decimal Sequences .. 45

Chapter 5 Analysis of Results .................................................. 48
  5.1 Analysis of Gain .......................................................... 48
  5.2 Image Watermarking in Presence of Noise ....................... 50
  5.3 Analysis of Multiple Watermark Embedding ...................... 52
  5.4 Variation of PSNR with Watermark Size ......................... 54
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Correlation and PSNR Values for Various Primes</td>
<td>49</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Watermark Embedding</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Watermark Detection</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Applications of Digital Watermarking</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Realization of Spread Spectrum Watermarking</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Linear Feedback Shift Register</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Autocorrelation Plot for PN Sequence for $n=6$</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Exponents of $\alpha$</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Autocorrelation Plot for Elliptic Curve with $n=13$</td>
<td>18</td>
</tr>
<tr>
<td>3.5</td>
<td>Butterfly Effect</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Autocorrelation Plot for CCS-PRBG</td>
<td>21</td>
</tr>
<tr>
<td>3.7</td>
<td>Cross Correlation Plot for Two CCS-PRBG</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>Autocorrelation Plot</td>
<td>25</td>
</tr>
<tr>
<td>3.9</td>
<td>Cross Correlation Plot for Primes 101 &amp; 181</td>
<td>26</td>
</tr>
<tr>
<td>3.10</td>
<td>Generation of Decimal Sequences</td>
<td>28</td>
</tr>
<tr>
<td>4.1</td>
<td>Decimal Sequence Watermarking Scheme</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Watermarked Image, PSNR=36.05 dB</td>
<td>34</td>
</tr>
<tr>
<td>4.3</td>
<td>Embedded Watermark</td>
<td>34</td>
</tr>
<tr>
<td>4.4</td>
<td>Recovered Watermark</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>Correlation Plot for Detection Statistic $r$</td>
<td>35</td>
</tr>
<tr>
<td>4.6</td>
<td>Watermarked Image, PSNR=39.22 dB</td>
<td>36</td>
</tr>
<tr>
<td>4.7</td>
<td>Embedded Watermark</td>
<td>36</td>
</tr>
<tr>
<td>4.8</td>
<td>Recovered Watermark</td>
<td>36</td>
</tr>
</tbody>
</table>
Figure 4.9 Correlation Plot for Detection Statistic $r$ ........................................36
Figure 4.10 Watermarked Image, PSNR=36.50 dB ........................................37
Figure 4.11 Embedded Watermark .................................................................37
Figure 4.12 Recovered Watermark .................................................................37
Figure 4.13 Correlation Plot for Detection Statistic $r$ ........................................38
Figure 4.14 Watermarked Image (381 X 381), PSNR=36.28 dB .......................38
Figure 4.15 Embedded Watermark .................................................................39
Figure 4.16 Recovered Watermark .................................................................39
Figure 4.17 Correlation Plot for Detection Statistic $r$ ........................................39
Figure 4.18 Gain Analysis Graph ................................................................40
Figure 4.19 Decimal Sequence Audio Watermarking Scheme .........................42
Figure 4.20 Input Audio File ........................................................................44
Figure 4.21 Watermarked Audio File ..............................................................44
Figure 4.22 Watermark ................................................................................44
Figure 4.23 Detected Watermark .................................................................44
Figure 4.24 Watermark 1 ...........................................................................45
Figure 4.25 Watermark 2 ...........................................................................45
Figure 4.26 Detection Statistic for Prime 2467 ...............................................46
Figure 4.27 Detection Statistic for Prime 8069 ...............................................46
Figure 4.28 Original Cover Image .................................................................47
Figure 4.29 Watermarked Cover Image .........................................................47
Figure 5.1 Different Watermarks Embedded .................................................49
Figure 5.2 Correlation vs. Variance of Gaussian Noise ....................................51
Figure 5.3 Watermarked Image with Gaussian Noise of Variance 0.06..............51
Figure 5.4 Detected Watermark from Noise Introduced Image.......................51
Figure 5.5 Watermarked Image.....................................................................52
Figure 5.6 Cover Image..................................................................................52
Figure 5.7 Embedded Watermarks.................................................................52
Figure 5.8 Noise Introduced Image.................................................................53
Figure 5.9 Recovered Watermarks.................................................................53
Figure 5.10 PSNR vs. Size of Watermark.......................................................55
Figure 5.11 PSNR vs. Size of Watermark for Pseudorandom Sequence...........55
Figure 5.12 PSNR vs. Size of Watermark Graph............................................56
Abstract

This thesis improves upon the work carried out by Mandhani on decimal sequence watermarking showing that the embedding and decoding algorithms used by Mandhani are not robust. We next specify improvements to these algorithms so that the peak signal to noise ratio (PSNR) is maximized and the distortion introduced in the image due to the embedding is minimized. By exploiting the cross correlation property of decimal sequences, the concept of embedding more than one watermark in the same cover image has been investigated. A comparison between the PSNRs for decimal sequence watermarking and other pseudorandom sequences has been made.
Chapter 1

Introduction

Cryptography and steganography (covert writing) have been used to add secrecy to communication all though history. Early methods of hiding information include invisible writing using invisible ink, writing text on wax-covered tablets and using null ciphers. Null ciphers were developed during the World War II, in which a secret was camouflaged in an innocent sounding message.

As technology improved and security became an important factor, other methods of hiding information were developed. Microdot technology was invented in 1941. Developments in this technology have advanced and today information is being hidden on strands of DNA using genomic technology [1].

With the growth of the internet, steganography has found new applications. Since steganography is relatively insecure to hide information, watermarking which is a more robust method of hiding information came into existence.

The growth in the distribution of media files over the internet has caused serious concerns regarding piracy and unauthorized distribution. The music industry claims annual multibillion-dollar loss in its revenues due to piracy, this poses a challenge to the cryptographers to improve on the existing copyright protection schemes and devise more robust and secure technologies.

Digital watermarking is also used for authentication. In watermarking information (watermark) is hidden in the digital file; therefore it can be implemented in digital rights management (DRM), so that the digital media files are distributed only to those who are authorized to receive it.
Traditional schemes of data protection such as scrambling and encryption are not feasible for multimedia digital files because the content must be played back in its original format, posing a threat of re-recording and re-distribution. Watermarking provides a solution to this, since the authenticity of the digital file may be recovered by detecting the watermark at the player (receiver). The existence of this watermark within the multimedia files goes unnoticed except when passed through an appropriate detector. Common types of signals to be watermarked are still images, audio, and digital video.

1.1 Background

The watermark is hidden in the cover object [10] such that it is completely invisible and it is difficult to either alter the watermark or completely remove it. Current DRM techniques are being researched exhaustively to establish more secure protocols and methods to secure information and control their distribution. DRM is not a single technology and it is not even a single philosophy. It refers to a broad range of technologies and standards that are still in the development state.

Microsoft’s Media Player has incorporated the DRM architecture to restrict the unauthorized distribution of media files. The basic technology being used in this scheme by Microsoft does utilize the watermarking technique.

DRM was introduced to reduce piracy of digital media. One can share digital files by simply copying them from one digital storage media to another. When the digital files are copyrighted it becomes hard to control the distribution rights of the digital file. This is the reason why a control mechanism has to be established on the transfer of copyrighted files over the internet.
Many application softwares have incorporated the Rights Expression Language (REL). This allows a publisher to designate a complex set of usage controls in terms of restricting the distribution of the media file, like printing, copying, etc. REL is a formal language like mathematics or like a programming code; it is a language that can be executed as an algorithm. Other languages like Open Digital Rights Language (ODRL) [24] and eXtensible rights Markup Language (XrML) [25] have been developed for the DRM architectures. Watermarking can be exploited in enhancing the scope of the DRM architectures.

1.2 Steganography and Watermarking

While cryptography deals with securing the contents of the message, steganography deals with concealing the message. Steganography means covered (stego) writing (graphy) [4].

Consider that Alice wants to transmit a message \( m \) to Bob. Alice selects a cover object \( C \), a stego-key \( K \) is used to embed the message \( m \) in the cover object \( C \) to transform it to a stego-object \( S \). Alice then transmits the stego-object to Bob without raising any suspicion. An eavesdropper cannot detect the existence of the message \( m \) in the stego-object. In a perfect system, the cover object cannot be differentiated from the stego-object either by a human or by a statistical system. Bob can decode the message \( m \) using the stego-key \( K \) only. Steganography requires that the cover object must have sufficient redundancy, which can be replaced by the secret information [10].

While steganography deals with concealing the message, watermarking has an additional requirement of securely concealing the message, such that the message cannot be tampered with.
An ideal steganographic system can embed a large amount of information with no visible degradation to the cover object, but an ideal watermarking system would embed an amount of information that cannot be altered or removed without making the cover object entirely unusable. A watermarking system involves trade off between capacity and security. A watermarking system comprises of the following procedures

- Watermark embedding; and
- Watermark detection.

Both these procedures use a secret *key*. The *key* could be either a public key or a private key. The *key* is used to enforce the security, i.e. prevention of unauthorized parties from detecting or tampering with the watermark. The embedded watermark can be recovered only with the secret key. Figure 1.1 shows the embedding procedure. The watermark can be any data file (audio, video, binary data). The cover object and the watermarked data are almost similar statistically.

![Figure 1.1 Watermark Embedding.](image)
Spread spectrum watermarking is one of the most popular methods of watermarking. In this technique, the watermark bits are randomly scattered in the cover object. This not only ensures that the watermark is robust to attacks but also simplifies the detection algorithm using correlation analysis [19, 13].

Other methods of watermarking of images and audio files have been proposed [16, 12, 2, 14]. Periera et.al [3] propose a technique for watermarking images using the lapped orthogonal transform while Kundur et.al [16] propose using the discrete Haar transform to watermark digital files. The watermarking schemes of Kundur et.al [12] are robust to cropping, filtering and scaling.

1.3 Applications of Watermarking

Some of the important applications of watermarking are as follows:

- User Authentication: Watermarking the digital stream of data can provide a way to authenticate specific users and to identify owners. For example, consider a DVD with a movie. The distribution center sends the copies of the movie to different distributors or movie stores. The distribution center can identify which DVD was sent to which distribution center by incorporating a watermark in the DVD data. This ensures that
watermark data can be recovered even if the DVD was illegally copied. Hence a trace can be performed to see exactly which distribution center was responsible for copying the DVD.

- **Data Authentication:** Since data can always be manipulated while being transmitted, it is very important that it should be authenticated. For example, consider different images being sent over a channel. An eavesdropper can alter these images and retransmit them, but watermarking the images can help in authenticating them. Even if an eavesdropper manages to alter the data, we can verify it by checking for the embedded watermark in the images.

- **Piracy Protection:** Piracy protection deals with stopping the unauthorized copying of data files. By embedding a watermark and a copy protect bit in the data, we can disable the copy procedure in a copying machine. For this scheme to work, it is imperative that the copying machine must have the necessary hardware to detect the watermark and consequently disable the copy procedure.

- **Broadcast Monitoring:** Broadcast monitoring refers to the concept of tracking the broadcast of a particular media file over a channel. For example, in video broadcasting, different watermarks can be embedded at particular intervals of time. By detecting these watermarks, a reference of as to how many times the video was broadcasted or whether the complete video was broadcasted can be ascertained.

- **Medical Applications:** Watermarking can be used in many medical applications also. Since many medical procedures use photographs (X-ray film) for references, these photographs may be watermarked with a unique ID of the patient, providing a method for patient identification.
- **Covert Communication**: Watermarking can be used in communication systems in countries where strong cryptography is not allowed.

![Diagram showing applications of digital watermarking]

**Figure 1.3 Applications of Digital Watermarking [12].**

### 1.4 Outline of Thesis

This thesis improves upon the work conducted by Mandhani [15] on watermarking using decimal sequences. In chapter 2, we provide a survey of this and the related watermarking schemes. Chapter 3 deals with the properties of sequences like decimal sequences, logistic map sequences, elliptic curve sequences and pseudorandom sequences. In chapter 4, we propose the new improved spread spectrum watermarking system using decimal sequences for images. Similar method has been adopted by Malvar, *et.al* [14]. Adaptive gain correction is performed in the embedding procedure so that the peak signal to noise ratio (PSNR) can be maximized. Gain analysis was performed so as to estimate the optimum gain necessary for robust watermarking. Also discussed in this chapter is
multiple watermark embedding using decimal sequences. We also discuss certain aspects of audio watermarking. Chapter 5 deals with the results and discussions, and Chapter 6 presents the conclusions.
Chapter 2

Literature Survey

Watermarking alters the original data $I$ with the watermark data $W$ such that the original image and the watermark can be recovered later. Some factors related to watermarking are robustness, security, transparency, complexity and capacity and some of these parameters are mutually exclusive tradeoffs. Robustness is related to the reliability of watermark detection after it has been processed through various signal-processing operations. Security deals with the difficulty of removing the watermark. A scheme is considered secure if the knowledge of the embedding algorithm does not help in detecting the hidden data bits. Capacity relates to the amount of information that can be embedded in a given cover object.

2.1 Watermarking Attacks

Any procedure that can decrease the performance of the watermarking scheme may be termed as an “attack”. Voloshynovskiy et.al [22] categorize attacks into four classes viz. removal, geometric, cryptographic and protocol. Removal attack removes the watermark without having any prior knowledge about the watermark, while geometric attacks deal with de-synchronization of the receiver so that watermark detection is distorted. Cryptographic schemes are those that tend to crack the watermarking scheme and protocol attacks exploit invertible watermarks to cause ownership ambiguity [17, 23].

2.2 Basics of Spread Spectrum Watermarking

A data signal to be watermarked may be modeled as a random vector. Let the elements of the data signal be denoted by $x \in R^N$, which will be assumed to be independent identically distributed (i.i.d) Gaussian random variables having a standard
deviation of $\sigma_x$ and mean zero. A watermark is defined as a direct spread spectrum sequence $w$, which is a random vector pseudo-randomly generated in $w \in \{\pm1\}^N$. Each element of $w$ is called a “chip”. Watermark chips have to be generated such that they are mutually independent with respect to the original data signal. The watermarked signal is $y = x + \delta w$, where $\delta$ is the watermark gain. It is implicit that the signal variance $\sigma_y^2$ directly impacts the security of the scheme; greater the value of the variance the more securely information can be hidden in the signal. On the other hand, greater the value of $\delta$, the more reliable the detection, less secure and potential visibility/audibility of the watermark. A basic model of this scheme is given in Figure 2.1.

![Figure 2.1 Realization of Spread Spectrum Watermarking.](image)

The watermark bit $b \in \{-1, 1\}$ is multiplied by the pseudorandom sequence, and added to the cover object $x$. Watermark extraction is achieved by cross correlating with the pseudorandom sequence.

The detection criteria is established using correlation analysis, that is the watermark $w$ is detected by correlating the given signal $y$ with $w$. 

10
\[ C(y, w) = y \cdot w = E[y \cdot w] + N\left(0, \frac{\sigma_x}{\sqrt{N}}\right) \]

If the signal \( y \) has been marked and no malicious attacks or other signal modifications are performed, then \( E[y \cdot w] = \delta \), else \( E[y \cdot w] = 0 \). The threshold for detection is \( \tau \). The watermark is present if \( C(y, w) > \tau \). Under the conditions that \( x \) and \( w \) are i.i.d. signals, such a detection scheme is optimal. We can thus define the probability of false alarm as follows

\[
P_{FA} = \Pr[C(y, w) \geq \tau \mid (y = x)] = \frac{1}{2} \text{erfc}\left(\frac{\tau \sqrt{N}}{\sigma_x \sqrt{2}}\right)
\]

Similarly the probability of false negative detection can be written as

\[
P_{ND} = \Pr[C(y, w) \leq \tau \mid (y = x + w)] = \frac{1}{2} \text{erfc}\left(\frac{(E[y \cdot w] - \tau) \sqrt{N}}{\sigma_x \sqrt{2}}\right)
\]

The straightforward application of the above method is neither highly robust nor reliable. Improvements have been suggested to this scheme by Kirovski et.al [11]. Rather than using the spatial domain for watermarking, a more stable and reliable transformation is performed on the cover data file. The transformation coefficients are then used for watermarking. This scheme provides greater robustness to attacks and does not cause audible/perceptual changes to the cover data file. Kirovski et.al [11] examine the various techniques that can be used for improving the robustness to attacks.

### 2.3 Watermarking Schemes

Traditional watermarking schemes consisted of visible watermarking [15]. Applications now demand that the watermark being embedded be highly robust to attacks. Techniques of hiding information in images include the use of discrete cosine transform
DCT) [18], discrete Fourier transform (DFT) [2] and wavelet transform. Periera et.al [16] have proposed a new method of watermarking using the lapped orthogonal transform (LOT), which offers advantages over DCT in terms of not producing the blocking artifacts when the gain is increased. But LOT is not secure to cropping, rotation and scaling attacks.

Cryptographers believe that spread spectrum (SS) method of watermarking can incorporate a high degree of robustness because the pseudo-random sequences being used in SS watermarking are very difficult to generate without the prior knowledge of the initial state of the random number generator. This secures decoding or removal of the watermark and also provides resistance to cropping. The major drawback of the SS watermarking scheme is that it requires a high gain value $\delta$, which sometimes tends to alter the cover data file considerably such that it is noticeable. To overcome this problem, Malvar, et.al [14] proposed the improved spread spectrum (ISS) technique. In this technique an adaptive gain mechanism has been established which enhances the performance by modulating the energy of the inserted watermark to compensate for the signal interference. A description of this is discussed in chapter 4. This thesis utilizes the ISS technique using the decimal sequences to enhance the performance of the embedding procedure and improve the overall performance of the watermarking scheme.
Chapter 3

Random Sequences and Properties

Many random sequences that have been generated using computers and statistical models appear to be random in nature. These sequences may be generated if we have the knowledge of the generating algorithm and the initial state of the random sequence generator.

3.1 Pseudorandom Noise Sequences (PN Sequences)

The generation of PN sequences may be viewed as representing the outcome of tossing a fair unbiased coin. The occurrence of head would denote +1 and a tail would denote a –1. Such a sequence would be perfectly random and would satisfy the conditions that qualify a sequence to be random.

A specific class of PN sequences may be generated using a simple Linear Feedback Shift Register (LFSR). Figure 3.1 shows a PN sequence generator that generates a maximal length sequence. A PN sequence is termed as a maximal length (ML) sequence if for a $n$-stage LFSR the period of the PN sequence equals exactly $2^n-1$.

![Figure 3.1 Linear Feedback Shift Register.](image)

$R_1 = R_2 \oplus R_3$, where $R_i$, $i=1, 2, 3$ refers to the register outputs.
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 sequence generated depends on the feedback connections. The sequence generated by using the above LSFR with initial state [0 1 0] is [0 1 0 1 1 1 0].

3.1.1 Properties of PN Sequences

PN sequences having periods equal to $2^n - 1$ for a $n$-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 can be termed as a sequence containing a single type of a digit. For a ML sequence, the number of runs of length 1 will be exactly one half, one quarter of length 2, one eighth of run 3, etc. In general, a sequence of length $n$ will have exactly $1/2^n$.

- **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

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

where

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

$R_{ss}(k)=-1/(2^n-1)$ ∀ $k \neq n$; and $R_{ss}(k)=1$ if $k=n$
- **Cross-correlation Property**: The cross-correlation property provides a measure of resemblance between two different sequences. Let $a = \{a_0, a_1, \ldots, a_{N-1}\}$ and $b = \{b_0, b_1, \ldots, b_{N-1}\}$ denote two different pseudorandom sequences. The cross-correlation of these two sequences is defined as follows

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

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

### 3.2 Elliptic Curve Pseudorandom Sequences

Elliptic curve pseudorandom sequences are relatively new when compared to the other pseudorandom sequences. The properties of these sequences can find use in cryptographic and watermarking applications. Elliptic curve pseudorandom sequences (ECPS) are generated in a finite field.
The detailed description of the generation of ECPS has been provided by Gong, *et al* [3].

3.2.1 Elliptic Curves over $F_{2^n}$

Consider an elliptic curve $E$ over $F_{2^n}$, the curve $E$ can be represented in the standard form as follows:

\[ y^2 + y = x^3 + c_4 x + c_6 \quad c_j \in F_{2^n} \text{ if } E \text{ is supersingular, and} \]
\[ y^2 + xy = x^3 + c_2 x + c_6, \quad c_j \in F_{2^n} \text{ if } E \text{ is non-supersingular.} \]

If we consider a point $P(x, y), x, y \in F_{2^n}$, that lies on the above mentioned curve, together with a point $O$ called the “point at infinity”, then we can construct an Abelian group $(E, +, O)$ where the identity element is $O$.

Let $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ be two different points in $E$ and both $P, Q \neq O$.

For $2P=P+P=(x_3, y_3)$,
\[ x_3 = x_1^4 + c_4^4; \quad y_3 = (x_1^2 + c_4)(x_1 + x_3) + y_1 + 1 \]

Let $P+Q=(x_3, y_3)$, if $x_1 = x_2$, then $P + Q = O$. Otherwise,
\[ x_3 = \lambda^2 + x_1 + x_2 \]
\[ y_3 = \lambda(x_1 + x_3) + y_1 + 1 \]

where
\[ \lambda = \frac{(y_1 + y_2)}{(x_1 + x_2)} \]
3.2.2 Construction of Pseudorandom Sequences from Elliptic Curves

Let $E$ be an elliptic curve as defined. Let $|E|$ denote the number of points of $E$. Let $P = (x_i, y_i)$ be a point of $E$. Let $\zeta = (P, 2P, 3P, iP, vP)$ where $iP = (x_i, y_i)$ and $P$ has the order of $v+1$.

Let

$$a_i = Tr(x_i) \quad \text{and} \quad b_i = Tr(y_i), i = 1, 2, \ldots, v,$$

$$S_0 = (a_1, a_2, \ldots, a_v) \quad \text{and} \quad S_1 = (b_1, b_2, \ldots, b_v)$$

where $Tr(x)$ is the trace function on $x$ [15].

We define $S = (S_0, S_1)^T$, a $(2,v)$ interleaved sequence such that the elements of $S$ are given by

$$s_{2i-1} = a_i \quad \text{and} \quad s_{2i} = b_i, i = 1, 2, \ldots, v$$

This interleaved sequence $S$ is called an elliptic curve pseudorandom sequence of type 1. Other methods of generation of ECPS have been discussed by Gong, et.al [3].

The period of such a sequence generated is equal to $2v$.

The various properties like the statistical properties, period of ECPS, linear span, etc. have been discussed by Gong.

For example let $n=5$,

- A finite field $F_{2^5}$ is generated using the primitive polynomial $f(x) = x^5 + x^3 + 1$, let $\alpha$ be a primitive root of $f(x)$. We represent the elements in $F_{2^5}$ as a power of $\alpha$.

  For the zero element, $0 = \alpha^\infty$.

- Choose a curve $E$: $y^2 + y = x^3$.

- Choose $P = (\alpha, \alpha^{23})$ with order 33.
• Compute $iP=(x_i, y_i)$, $i=1, 2, 3\ldots 32$. The exponents of $\alpha$ for each of the points are as shown below.

| (1, 23) | (4, 13) | (18, 7) | (16, 27) | (13, 5) |
| (10, 2) | (26, 6) | (2, 22) | (5, 14) | (21, 12) |
| (\infty, 0) | (9, 19) | (22, 17) | (11, 9) | (20, 25) |
| (8, 29) | (8, 26) | (20, 4) | (11, 24) | (22, 18) |
| (9, 8) | (\infty, \infty) | (21, 20) | (5, 1) | (2, 15) |
| (26, 10) | (10, 28) | (13, 3) | (16, 21) | (18, 16) |
| (4, 30) | (1, 11) |

Figure 3.3 Exponents of $\alpha$ [3].

• $\text{Tr}(x_i)=00101110110111110011110111011000$

• $\text{Tr}(y_i)=0110100110110110100100100101001$

• Interleaved $(a_i, b_i)=0001110011101001111100111000110111000111001100011$

The period is 64.

Figure 3.4 Autocorrelation Plot for Elliptic Curve with $n=13$. 
3.3 Logistic Map Pseudorandom Sequences

The logistic map is a simple example of a discrete dynamical system that names a complete family of iterative functions, given by

$$f_n = cf_{n-1}(1 - f_{n-1}); \quad c \in R \mid 0 \leq c \leq 4, n \in Z$$

Different values of $c$ would yield different dynamic behavior of the system. A class of logistic map is defined when $f_n \in (0, 1)$. The logistic map numbers are generated by iteration of the above function. Such numbers exhibit chaotic nature and are highly deterministic. That is, if we iterate the function $n$-times the final value of the function is solely determined by the initial state $f_0$ and the constant $c$.

3.3.1 Properties of Logistic Map Numbers

For the logistic map function as defined above, it is difficult to predict the future values of the function. Edward Lorenz first analyzed this effect in 1963. The butterfly effect has usually been stated as “the propensity of a system to be sensitive to initial conditions”. The values of the function vary largely even for small changes in the initial conditions [26]. This is illustrated in the Fig 3.5.

In figure 3.5, the values of two logistic maps with initial conditions of $f_0=0.95$, $0.949999999$ and $c=3.75$ tend to exhibit greater changes for iteration values greater than 30. This is the region where the logistic maps tend to start exhibiting chaos. For a very small variation in the initial conditions, the generated sequences are completely different. This property wherein minute changes in the input intensify with time and later exhibit total chaos is termed as “Butterfly Effect”
Figure 3.5 Butterfly Effect [26].

The behavior of the logistic maps not only depends on the initial value $f_0$ but also on the constant $c$. For numerous values of $c$, the value of $f_0$ doesn’t really matter, since the function converges eventually. For values of $c \in (0, 3)$ approximately, the map converges to various points depending upon the initial condition $f_0$. It converges first to 2, then to 4, 8, 16, etc., until it reaches the chaotic behavior.

3.3.2 Generation of Random Sequences from Logistic Maps

The algorithm to generate logistic map random sequences is as follows:

Consider two different one-dimensional chaotic maps $F_1(f_1(0), c_1)$ and $F_2(f_2(0), c_2)$ where $f$ and $c$ have the usual meaning. We define a pseudorandom bit sequence $k(i) = g(f_1(i), f_2(i))$, where

$$g(x, y) = \begin{cases} 
1, & x > y \\
no output, & x = y \\
0, & x < y 
\end{cases}$$
This realization can be used to generate “Couple of Chaotic Systems based Pseudo-Random Bit Generator” or simply CCS-PRBG. [20].

Figure 3.6 Autocorrelation Plot for CCS-PRBG.

Figure 3.7 Cross Correlation Plot for Two CCS-PRBGs.
3.4 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 [7] and some of the important properties from this are presented here.

3.4.1 Properties of Decimal Sequences

We can express any positive number as a decimal in the base \( r \) as

\[
A_1A_2........A_{x+1}a_1a_2........
\]

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

\[
x = A_1 r^r + A_2 r^{r-1} + ... + A_{x+1} \frac{a_i}{r} + \frac{a_1}{r^2} + ...
\]

We can use decimal sequences for rational and irrational numbers to generate pseudorandom noise 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 is evident 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 \( 10^2 \equiv 1 \) (modulo 7), \( 10^3 \equiv 1 \) (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 \( \frac{p'}{q}, \ (p', q) = 1, p' < q \) where \( p' \equiv r' \times p \) (modulo \( q \)).

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

\[
a_i + a_{(i+1)_{\text{mod} q-1}} = r-1
\]

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 \( a_i + a_{8i} = r - 1 = 9 \)

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 \( a_i + a_{9i} = r - 1 = 1 \)

**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, we may state 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 \( a_n \in \{0, 1, 2, ..., r-1\} \). 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 \((r-1)\) were mapped symmetrically about zero by the transformation \(a'_i = 2a_i - (r-1)\). A straightforward calculation shows that

\[
R_x(\tau) = \begin{cases} 
\frac{(r^2 - 1)}{3} & \tau = 0; \\
0 & \text{otherwise.}
\end{cases}
\]

Figure 3.8 Autocorrelation Plot.

- **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 \(\{x\} = a_1a_2...a_k\) and \(\{y\} = b_1b_2...b_k\). The period of the product sequence \(a_i b_{i+\tau}\) is \(N=\text{LCM} (k_1, k_2)\), where \(\text{LCM}\) 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.

![Cross Correlation Plot for Primes 101 & 181](image)

Figure 3.9 Cross Correlation Plot for Primes 101 & 181.

### 3.4.2 Generation of Decimal Sequences

Using feedback shift registers that allow carry we can generate decimal sequences. We can also generate these sequences using a computational device by using the following equations [7]:

$$a^i = [i/(r^i \mod q)] \mod r$$

$$q \mod r \equiv -k \equiv -1/l$$

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 \textit{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.

\textit{Theorem}: Let $1/(tr-1)$ define the decimal sequence $a_1a_2a_3\ldots a_k$, where $r$ is the radix.

Consider another sequence $u_1u_2u_3\ldots u_k$, where, for all $i$, $u_i < t$, then

$$ru_i + a_i = u_{i+1} + ta_{i+1}$$

\textit{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

$$a_{k-1}(tr-1) + t = m_{l-1}r$$

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

$$u_ku_{k-1}u_{k-2}\ldots 1$$

$$a_k a_{k-1}a_{k-2}\ldots 1$$

Equals

$$0 0 \ldots 1$$

$$1 \ t \ [t^2] \mod r \ldots 0$$

The circuit for the generation of decimal sequences is given in figure 3.10. It consists of $n$ stages of shift registers. The $C$s represent the carries that are added to the immediate preceding stages. 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.

![Diagram](image)

Figure 3.10 Generation Of Decimal Sequence.
Chapter 4

Decimal Sequence Image Watermarking

The decimal sequence spread-spectrum watermarking scheme is shown in figure 4.1. 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.

\[ \beta = \langle x, u \rangle = \frac{1}{N} \sum_{i=0}^{N-1} x_i u_i \quad \text{and} \quad \|x\| = \langle x, x \rangle \quad (4.1) \]

where \( N \) is the length of the vectors \( x, s, u, n \) and \( y \). We assume that we are embedding one bit of information in a vector \( s \) of \( N \) transform coefficients. That is, the bit rate is \( 1/N \) bits/sample. The embedding is performed according to the following equation,

\[ s = x + bu \quad (4.2) \]
Distortion $D$ associated with the embedded signal is defined as $\|s - x\|$.  

$$ D = \|s - x\| = \|bu\| = \|n\| = \sigma_u^2 \quad (4.3) $$

We assume that the channel is being modeled as an additive white Gaussian noise (AWGN). Thus,

$$ y = s + n \quad (4.4) $$

Detection is performed based on the detection statistic $r$

$$ r = \frac{\langle y, u \rangle}{\langle u, u \rangle} = \frac{\langle bu + x + n, u \rangle}{\sigma_u^2} = b + \bar{x} + \bar{n} \quad (4.5) $$

and the estimated bit

$$ \beta = \text{sign}(r) \quad (4.6) $$

where

$$ \bar{x} = \frac{\langle x, u \rangle}{\|u\|} \quad \& \quad \bar{n} = \frac{\langle n, u \rangle}{\|u\|} \quad (4.7) $$

We assume the simple statistical model for the signal $x$ and noise $n$. We assume these two to be Gaussian random processes. Therefore,

$$ x_i \approx N(0, \sigma_x^2), \quad n_i \approx N(0, \sigma_n^2) $$

Thus the detection statistic $r$ is also Gaussian, i.e.

$$ r \approx N(m_r, \sigma_r^2), \quad m_r = E[r] = b\sigma_r^2 = \frac{\sigma_x^2 + \sigma_n^2}{N\sigma_u^2} \quad (4.8) $$

The above statistic is a very good approximation for spread spectrum watermarking.

The probability of error for the above detection statistic $r$ is as follows

$$ p = \Pr(\beta < 0 | b = 1) = \frac{1}{2} \text{erfc} \left( \frac{m_r}{\sigma_r\sqrt{2}} \right) $$
\[ \frac{1}{2} \text{erfc} \left( \frac{\sigma_x^2 N}{\sqrt{2(\sigma_x^2 + \sigma_n^2)}} \right) \]  

(4.9)

where \( \text{erfc}(\cdot) \) is defined as the complementary error function. The same error probability is achieved for \( b = -1 \).

4.1 Improvements to Traditional Watermarking Scheme

In this technique of watermarking we utilize the knowledge of the signal \( x \) to modulate the energy of the inserted watermark so that the interference due to \( x \) can be compensated for [14]. We define a function \( g(x, b) \) which helps in varying the amplitude of the chip sequence \( u \). Thus we have

\[ s = x + g(x, b) \cdot u \]  

(4.10)

For simplicity we assume that \( g(x, b) \) is a linear function. We can thus write

\[ g(x, b) = \mu b - \lambda \bar{x} \]  

(4.11)

thus,

\[ s = x + \left( \mu b - \lambda \bar{x} \right) \cdot u \]  

(4.12)

and the detection statistic would thus be

\[ r = \frac{\langle s, u \rangle}{\|u\|} = \mu b + (1 - \lambda) \bar{x} + n \]  

(4.13)

The closer we make \( \lambda \) to 1, the greater the influence of \( x \) is removed from \( r \). The detector still remains the same, i.e., the detected bit is \( \text{sign}(r) \).

The expected distortion \( D \) is

\[ E[D] = E[\|s - x\|] \]
\[
E\left[ \mu b - \lambda \chi^2 \sigma_u^2 \right] = \left( \mu^2 + \frac{\lambda^2 \sigma_x^2}{N\sigma_u^2} \right)\sigma_u^2 \tag{4.14}
\]

In order to have the same distortion as the traditional distortion we force \( D \) to be equal to that of the traditional distortion (equation 4.3). We have

\[
\left( \mu^2 + \frac{\lambda^2 \sigma_x^2}{N\sigma_u^2} \right)\sigma_u^2 = \sigma_u^2 \tag{4.15}
\]
or

\[
\mu = \sqrt{\frac{N\sigma_u^2 - \lambda^2 \sigma_x^2}{N\sigma_u^2}} \tag{4.16}
\]

We can choose a particular distortion value by equating Equation 4.15 to be equal to a fraction of the traditional distortion. We compute the mean and variance of the sufficient statistic. They are

\[
m_r = \mu b, \quad \text{and} \quad \sigma_r^2 = \frac{\sigma_n^2 + (1 - \lambda)^2 \sigma_x^2}{N\sigma_u^2} \tag{4.17}
\]

The error probability can be calculated as follows:

\[
p = \frac{1}{2} \text{erfc} \left( \frac{m_r}{\sigma_r \sqrt{2}} \right) \tag{4.19}
\]

\[
= \frac{1}{2} \text{erfc} \left( \frac{1}{\sqrt{2}} \sqrt{\frac{N\sigma_u^2 - \lambda^2}{\sigma_n^2 + (1 - \lambda)^2}} \right) \tag{4.20}
\]

The maximum value of error probability \( p \) can be found by equating \( \frac{\partial p}{\partial \lambda} = 0 \), we get the optimum value of \( \lambda \).
According to equation 4.14, we can reduce the distortion in the improved scheme by selectively choosing the values of $\lambda$ and $\mu$. Let us assume that we want a distortion level 10 times lesser than that of the traditional level. Thus using equation 4.15 and equating it to one tenth of the distortion of the traditional scheme we get

$$\left(\mu^2 + \frac{\lambda^2 \sigma_x^2}{N\sigma_u^2}\right)\sigma_u^2 = \frac{\sigma_u^2}{10}$$  \hspace{1cm} (4.22)

or

$$\mu = \sqrt{\frac{N\sigma_u^2 - 10\lambda^2 \sigma_x^2}{10N\sigma_u^2}}$$  \hspace{1cm} (4.23)

For a simplistic model let us initially consider that there is no channel noise associated with the scheme, i.e. $\sigma_n^2 = 0$, thus equation 4.21 becomes

$$\lambda_{\text{opt}} = \frac{1}{2} \left\{ 1 + \frac{N\sigma_u^2}{\sigma_x^2} \right\} - \sqrt{\left( 1 - \frac{N\sigma_u^2}{\sigma_x^2} \right)^2} = 1$$

now substituting this value of $\lambda_{\text{opt}}$ and taking $\sigma_u^2 = 1$ in equation 4.23 we get

$$\mu = \sqrt{\frac{N - 10\sigma_x^2}{10N}}$$  \hspace{1cm} (4.24)

We utilize the above equations to watermark grayscale images. The cover image is a grayscale image of Lena, the value of $\sigma_x^2$ is nearly equal to 2045 with a standard deviation of 45.22, also $N=512*512$ (size of image) thus substituting these values in equation 4.24 we get
\[ \mu \approx \sqrt{\frac{1}{10}} = 0.316 \]

we choose this value of \( \mu \) and \( \lambda \) for our embedding algorithm. From equation 4.7 we have \( \overline{x} = 0.0299 \) for the decimal sequence generated by the prime \( q = 8069 \). Thus the equation 4.12 becomes

\[ s = x + (0.316 \cdot b - 0.0299) \cdot u \] (4.25)

Figure 4.2 Watermarked Image, PSNR=36.05 dB.

Figure 4.3 Embedded Watermark.

Figure 4.4 Recovered Watermark.
Now we chose another prime $q=2467$ to compare the results. From equation 4.7 we have $\bar{x}=0.0367$.

Thus equation 4.25 changes to

$$s = x + (0.316^* b - 0.0367) \cdot u$$

(4.26)

We observe from equations 4.25 and 4.26, that the variations in the second term are very small, which means that the effect of the cover image in the embedding procedure is almost negligible. The Peak Signal to Noise Ratio (PSNR) for this prime was found to be 39.22 dB.
Figure 4.6 Watermarked Image, PSNR = 39.22 dB.

Figure 4.7 Embedded Watermark.

Figure 4.8 Recovered Watermark.

Figure 4.9 Correlation Plot for Detection Statistic $r$. 

36
Now we chose the same prime \( q = 2467 \) but a different cover image to compare the results.

From equation 4.7 we have \( \tilde{x} = -0.0909 \).

Thus equation 4.25 changes to

\[
\begin{align*}
\bar{s} &= x + (0.316 \cdot b + 0.0909) \cdot u \\
\end{align*}
\]

(4.27)

The Peak Signal to Noise Ratio (PSNR) for this prime was found to be 36.50 dB.

Figure 4.10 Watermarked Image, PSNR=36.50 dB.

Figure 4.11 Embedded Watermark.

Figure 4.12 Recovered Watermark.
Now we chose the same prime $q=2467$ but a different cover image to compare the results.

From equation 4.7 we have $\bar{x} = -0.0783$ Thus equation 4.25 changes to

$$ s = x + (0.316 \cdot b + 0.0783) \cdot u $$

(4.28)

The Peak Signal to Noise Ratio (PSNR) for this prime was found to be 36.28 dB.
From Figures 4.13 and 4.17 we notice that the detection statistic plot had changed, this is because of the change in size of the image. The cover image used in Figure 4.13 had dimensions 512 X 512, whereas the one in Figure 4.17 has 381 X 381. We infer that we can embed the same amount of information in two different sized images with an increase in distortion $D$ for the image with lesser dimensions.
4.2 Gain Analysis of Decimal Sequence Watermarking Scheme

Simulations were performed to analyze the effects of varying the parameter $\mu$ in the embedding algorithm. We refer this to be the gain. By optimizing the value of $\mu$ we can find the most effective scheme for embedding the watermark in the cover object. We performed simulations by varying $\mu$ from 0.1 to 1 in steps of 0.05. Primes were chosen at random for this simulation and the results are shown in figure 4.18. We observe that for most primes the optimal gain value lies close to 0.3, which indicates that with our proposed scheme using decimal sequences as the random sequences we can reduce the distortion $D$ almost 10 fold when compared to the traditional watermarking scheme using pseudorandom sequences as the spreading sequences. Also for certain primes perfect detection is possible for gain values as low as 0.175. These primes can be used to further reduce the distortion.

![Gain Analysis Graph](image)

Figure 4.18 Gain Analysis Graph.
Certain primes have low gains associated with them. Such primes are very important to us because the decimal sequences generated by these primes have good statistical properties. Primes, whose decimal sequence has period \( p \) equal to \( (q-1)/n \), where \( n \) is even, exhibit good statistical properties. The autocorrelation of such sequences is lower when compared to that of primes whose periods are \( (q-1)/n \), where \( n \) is odd. Whenever \( n \) is odd, the first \( (p-1)/2 \) digits of the decimal sequence will be exactly the complement of the subsequent digits. The first \( (p-1)/2 \) digits are sufficient to generate the last \( (p-1)/2 \) digits. This is the reason why such primes have higher gain values.

For any prime the optimal value of the gain should be greater than 0.32. This is the reason why in section 4.1 we have chosen the distortion \( D \) to be about 10 times lesser than that of the traditional watermarking scheme, so that the optimal gain then lies close to 0.32. Those primes which have gains lesser than 0.3 may be used to embed more information in the cover object because the distortion associated with these primes is lesser than that of primes whose gains are larger than 0.3. Thus more information can be hidden using primes whose gains are less than 0.3 for the same distortion value \( D \). For example for the prime \( q=5647 \), the period of the decimal sequence generated is 2823 and the optimal gain associated with this is about 0.2.

Thus from equations 4.22-4.24 we get that

\[
\mu \approx 0.2 = \frac{1}{\sqrt{x}}
\]

Solving the above equation for \( x \), we get \( x=25 \). The distortion can be reduced by a factor of 25, when compared to the traditional method.
4.3 Implementation of Improved Scheme in Audio Watermarking

Watermarking of audio signals requires criteria different from that of images. In most audio watermarking schemes the actual audio file is transformed to a domain in which spread spectrum watermarking can be carried out. This transform should be such that any alterations to the transformed data must not cause audible changes in the watermarked audio data. In this section we basically use the wavelet transform and the Bi-orthogonal window.

Wavelet-transformed domain is chosen for audio watermarking since the human auditory system (HAS) can not differentiate when we alter the frequencies above 2 KHz since our HAS acts a band pass filter with a number of overlapping filters. Above 2 KHz, the HAS focuses on the temporal envelope of the signal rather than its actual structure. We use the decimal sequences as the spreading sequences because these sequences have low gains. We exploit this property of the decimals sequences to find a robust way of watermarking audio files. The encoder for watermarking audio files is shown in Figure 4.19.

![Wavelet Transform Diagram](image-url)

**Figure 4.19 Decimal Sequence Audio Watermarking Scheme.**
We watermarked an audio file using the above scheme and simulation results indicate that the appropriate value of the variance of the decimal sequence $\sigma_u^2 = 1.6 \times 10^{-5}$, provided excellent detection of the embedded data. The audio file used was a .wav file sampled at 12000 bits per second. The values taken by the various samples of the wave file lie between –1 and 1. A variance of $1.6 \times 10^{-5}$ is very low to cause audible fluctuations in the audio file and hence there is very less distortion introduced in the cover data file after the watermark was embedded.

An audio file as shown in Figure 4.20 was chosen for watermarking. The value of the variance was set to $1.6 \times 10^{-5}$, the prime $q=2467$ was used to generate the decimal sequence. After the embedding procedure, the correlation between the original signal and the watermarked signal was found to be 0.995. Figure 4.21 shows the watermarked audio file and figure 4.22 shows the watermark embedded. We chose a black and white image as our watermark. The decoded watermark is shown in figure 4.23. We have perfect detection, which indicates that our proposed method may provide a certain degree of spread spectrum robustness. If noise is considered to be a major factor then we may appropriately choose the values of the variance of the decimal sequence and the prime $q$ and use equations derived in section 4.1 to improve the noise immunity of the watermarked audio file.

Thus it is possible to utilize our proposed scheme for watermarking audio files also. Audio watermarking, requires criteria different from that of image watermarking. Kirovski, et.al [11] have proposed the use of modulated complex lapped transform to watermark audio files.
Figure 4.20 Input Audio File.

Figure 4.21 Watermarked Audio File.

Figure 4.22 Watermark.

Figure 4.23 Detected Watermark.
4.4 Multiple Watermark Embedding Using Decimal Sequences

In this section we deal with the multiple watermark embedding using the decimal sequences. As discussed in previous sections decimal sequences can be used with very low gain values, this property hints that we can perhaps embed multiple watermarks in the same cover object without causing considerable distortion in the cover image. The cross correlation properties of decimal sequences [7] which states that the cross correlation function of two maximum length decimal sequences in the symmetric form is identically equal to zero if the ratio \( k_1 / k_2 \) of their periods reduces to a irreducible fraction \( n_1 / n_2 \), where either \( n_1 \) or \( n_2 \) is an even number. By choosing the prime \( q_1 \) and \( q_2 \) we can generate decimal sequences whose cross-correlation is equal to zero. We used the primes 2467 and 8069 with periods 2466 and 4019 respectively. \( \bar{x} \) for \( q=2467 \) was found to be 0.0367 and that for \( q=8069 \) was –0.0299. First we embedded the watermark shown in figure 4.24, the PSNR was found to be 37.78 dB. Next we embedded the second watermark as shown in figure 4.25, the PSNR reduced to 34.98.

![Figure 4.24 Watermark 1.](image1) ![Figure 4.25 Watermark 2.](image2)

The detection for both the primes 2467 and 8069 was perfect. This indicates to us that we can embed more than one watermark in the same cover object with some deterioration in the PSNR. The detection statistics are shown in figures 4.26 and 4.27 for the primes 2467 and 8069 respectively.
We experimented with various combinations of primes to verify the robustness of the scheme and found that in all combinations the detection of the watermarks was perfect. Thus we have not only improved the previous decimal sequence watermarking scheme but
have also provided a way of embedding more than one watermark in the same cover object. The original cover object and the watermarked object are shown in figures 4.28 and 4.29 respectively.

Figure 4.28 Original Cover Image.           Figure 4.29 Watermarked Cover Image.
Chapter 5

Analysis of Results

5.1 Analysis of Gain

In chapter 4 we adopted the improved spread spectrum watermarking scheme for watermarking the images. The equations and results obtained in chapter 4 indicate to us that since the length $N$ is considerably large the effect of the signal energy on the watermarking scheme is almost negligible. From equations 4.25 - 4.28, the value of $x$ was found to have very low fluctuations; hence we can neglect this change. We preformed simulations for various images and various primes and chose the values of $\mu = 0.316$ and $\lambda = 1$ for our simulations.

As observed from the table, we were able to recover the watermarks perfectly for different cover images and also for different watermarks. We note that the correlation value for the prime 1109 is not equal to 1, the reason being, the period of the decimal sequence generated using the prime 1109 is 1108, which is small and hence as discussed by Mandhani [15], the perfect recovery of the watermark is not possible for certain primes with low periods. Mandhani in his thesis hasn’t considered the Peak Signal to Noise Ratio (PSNR) after the watermark has been embedded. This is a very important criterion in determining the imperceptibility of the watermark. Kundur et.al [12] state that the PSNR is one of the most important criterions in determining the invisibility of the watermark in the embedded image. Mandhani, had to use a gain factor of 3 for watermarking. We have been able to reduce this gain factor by a very large factor.
Table 5.1 Correlation and PSNR Values for Various Primes.

<table>
<thead>
<tr>
<th>Image</th>
<th>Watermark Image</th>
<th>PSNR (dB)</th>
<th>Correlation Value</th>
<th>Prime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>1</td>
<td>38.4220</td>
<td>1.000</td>
<td>2467</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>36.3912</td>
<td>1.000</td>
<td>8069</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>36.1023</td>
<td>1.000</td>
<td>5647</td>
</tr>
<tr>
<td>Lena</td>
<td>1</td>
<td>38.1792</td>
<td>1.000</td>
<td>2999</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>36.3004</td>
<td>1.000</td>
<td>2423</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>36.4236</td>
<td>1.000</td>
<td>1109</td>
</tr>
<tr>
<td>Lena</td>
<td>1</td>
<td>38.3528</td>
<td>1.000</td>
<td>10459</td>
</tr>
<tr>
<td>Lena</td>
<td>2</td>
<td>36.4151</td>
<td>1.000</td>
<td>14713</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>36.2188</td>
<td>1.000</td>
<td>17837</td>
</tr>
<tr>
<td>Building</td>
<td>1</td>
<td>38.7069</td>
<td>1.000</td>
<td>2467</td>
</tr>
<tr>
<td>Building</td>
<td>2</td>
<td>36.6781</td>
<td>1.000</td>
<td>8069</td>
</tr>
<tr>
<td>Building</td>
<td>3</td>
<td>36.3910</td>
<td>1.000</td>
<td>5647</td>
</tr>
<tr>
<td>Building</td>
<td>1</td>
<td>38.4669</td>
<td>1.000</td>
<td>2999</td>
</tr>
<tr>
<td>Building</td>
<td>2</td>
<td>36.5885</td>
<td>1.000</td>
<td>2423</td>
</tr>
<tr>
<td>Building</td>
<td>3</td>
<td>36.7105</td>
<td>0.9630</td>
<td>1109</td>
</tr>
<tr>
<td>Building</td>
<td>1</td>
<td>38.6374</td>
<td>1.000</td>
<td>10459</td>
</tr>
<tr>
<td>Building</td>
<td>2</td>
<td>36.7010</td>
<td>1.000</td>
<td>14713</td>
</tr>
<tr>
<td>Building</td>
<td>3</td>
<td>36.5069</td>
<td>1.000</td>
<td>17837</td>
</tr>
</tbody>
</table>

Figure 5.1 Different Watermarks Embedded.
As shown in Figure 4.18 the most optimal value of gain can be as low as 0.175 and for all values of gain over 0.4 the recovery of the watermark was always perfect. This fall in gain was possible because of the watermarking algorithm, which is superior to Mandhani’s. Our algorithm provides the flexibility of considering the effects of noise also. Also the distortion introduced in the cover image can be adjusted appropriately using these values of $\mu$ and $\lambda$. The values of $\mu$ and $\lambda$ can be appropriately chosen so that the effect of the cover image can be reduced in the embedding algorithm. Our watermarking scheme reduces the distortion by a factor of 10 when compared with traditional watermarking schemes.

5.2 Image Watermarking in Presence of Noise

We assume that the noise introduced in the digital data file is Gaussian in nature. We considered the Lena image of size 512 X 512 and introduced Gaussian noise with zero mean to the watermarked image to verify to what extent our proposed scheme can withstand noise. The noise in the image was introduced having variance in the range of 0.001 to 0.5. The correlation between the original watermark and the recovered watermark is shown in Figure 5.2. The noise was introduced in the watermarked image by using the MATLAB function `imnoise`. The graph shown in Figure 5.2 indicates that for Gaussian noise with a variance of upto 0.06 the watermark detection was perfect. When the variance was increased further to 0.2 we had almost perfect detection of the watermark. The graph indicates the noise immunity of the proposed watermarking scheme. Figure 5.3 shows the watermarked image with the Gaussian noise introduced. The variance of the Gaussian noise was 0.06. We could detect the watermark perfectly from this noise introduced image. The detected watermark is shown in figure 5.4.
Figure 5.2 Correlation vs. Variance of Gaussian Noise.

Figure 5.3 Watermarked Image with Gaussian Noise of Variance 0.06.

Figure 5.4 Detected Watermark from Noise Introduced Image.
The results obtained from the graph in figure 5.2 indicate that our proposed watermarking scheme may be immune to Additive Gaussian Noise considerably.

5.3 Analysis of Multiple Watermark Embedding

Our watermarking scheme causes less distortion to the image, hence we can embed more than one watermark in the same cover object. As discussed in section 4.4 we embedded two different watermarks in the same cover object and were also able to retrieve both the watermarks. In order to verify the robustness of our scheme to noise, we introduced noise in the watermarked cover image. We were able to retrieve both the watermarks even in the presence of noise. We chose the building image as our cover image. Primes 10459 and 2467 were chosen for this purpose.

![Figure 5.5 Watermarked Image.](image1)
![Figure 5.6 Cover Image.](image2)

The embedded watermarks were,

![Figure 5.7 Embedded Watermarks.](image3)
The PSNR was found to be 34.89 after both the watermarks were embedded in the cover image.

We introduced Gaussian noise in the cover image and fixed the variance at 0.05. The noise introduced cover image is as shown in figure 5.8.

![Noise Introduced Image](image)

Figure 5.8 Noise Introduced Image.

Both the watermarks were detected from this image were retrieved perfectly. The detected watermarks are as shown below.

![Recovered Watermarks](image)

Figure 5.9 Recovered Watermarks.
Multiple watermarking is possible with decimal sequences because of the cross-correlation property. The cross correlation between two decimal sequences is near to zero for most of the primes. This means that the two watermarks being embedded in the cover image are mutually independent of each other. The first watermark embedded in the cover image does not effect the embedding of the second watermark. This provides greater security to the cover image in terms of authenticating it and also as a means of securely transmitting covert information.

5.4 Variation of PSNR with Watermark Size

The peak signal to noise ratio [12] is a very good measure of the amount of distortion introduced in the cover image due to the watermark embedding procedure. In order to determine this distortion we found out the PSNR for different sizes of watermarks. Figure 5.10 shows the variation of the PSNR with watermark size. The graph is almost exponential in nature. From the graph we may infer that for watermark sizes lying between 200 and 600 the fall in PSNR is about 5 db. This is the best range of watermark sizes that we can choose for a 512 X 512 cover image. The prime 2467 was used for decimal sequence watermarking, while the pseudorandom sequence was generated using the \textit{rand} function in MATLAB. The PSNR of pseudorandom sequences vs. the size of watermark is shown in figure 5.11.

From the Figure 5.12, we observe that the PSNR due to the decimal sequence watermarking is almost 4 dB less than that of pseudorandom sequence (generated using \textit{rand} function in MATLAB) watermarking. This means that we can embed almost twice the amount of information using decimal sequence watermarking over pseudorandom sequence watermarking.
Figure 5.10 PSNR vs. Size of Watermark.

Figure 5.11 PSNR vs. Size of Watermark for Pseudorandom Sequence Watermarking.
Observe the point (190, 32.01) on the pseudorandom sequence curve and the point (390, 32.58) on the decimal sequence curve, both these points have almost the same PSNR, but the size of the watermarks is almost twice for decimal sequence curve. The difference in PSNRs mainly arises due to the fact that in pseudorandom sequence watermarking the correlation at the output is never complete because the period of pseudorandom sequence is larger than the size of the image (512 x 512). In order to counter the effect of this we have to increase the gain of the watermarking scheme considerably. This is the reason why there is a fall in PSNR for pseudorandom sequence watermarking. Thus using decimal sequence watermarking and our proposed scheme we can embed watermarks of larger size.
Chapter 6

Conclusions

This thesis improves the work by Mandhani [15] on watermarking using decimal sequences. By appropriately selecting the primes and the gain associated with the embedding procedure we can reduce the distortion introduced in the cover image. Our proposed scheme has the following advantages over the earlier watermarking scheme:

- The noise immunity of the watermarked image is increased by appropriately choosing the prime and the values of $\mu$ and $\lambda$.
- We have reduced the distortion introduced in the image. Mandhani had used values of gain in the range of 2-4 for his embedding algorithm, while in our method we have been able to use gains as low as 0.175.
- Multiple watermarks may be embedded using different decimal sequences by exploiting the cross correlation property.
- Results indicate that decimal sequence watermarking can be used advantageously for audio files. This is because of the low gains associated with it.
- Using decimal sequences as compared to pseudorandom sequences we were able to embed almost twice the data in a cover image for a fixed distortion value $D$. This is due to the fact that in pseudorandom sequence based systems, complete correlation is not possible because of the large periods of the pseudorandom sequences. Hence to compensate for this, the gain has to be increased considerably.

This thesis is limited to watermarking of grayscale images in the spatial domain. The watermark is a black and white image. Further research can be carried out for color image
watermarking, video watermarking and also audio watermarking. If one were to embed a
watermark with more than two symbols, advanced algorithms for embedding and detection
can be devised. We may use different decimal sequences to embed different symbols of a
watermark. Further research may lead to watermarking grayscale images with grayscale
watermarks.
Bibliography


24. [www.odrl.com](http://www.odrl.com)

25. [www.xrml.com](http://www.xrml.com)

Vita

Shaik Ashfaq Naveed was born in Andhra Pradesh, on 14\textsuperscript{th} June 1982, India. He earned his primary and secondary education from Bharatiya Vidya Bhavan’s Public School in Hyderabad, Andhra Pradesh. After finishing his high school, he took a very competitive entrance examination for engineering known as EAMCET. After qualifying this examination he got admission to, department of Electronics and Communication Engineering, Osmania University. He received his Bachelor of Engineering in Electronics and Communication Engineering from Osmania University, Hyderabad, India, in spring 2003. After his graduation, he came to United States of America to pursue master’s degree. He then joined the graduate program at Louisiana State University, Baton Rouge, in August 2003. He is a candidate for the degree of Master of Science in Electrical Engineering to be awarded at the commencement of spring, 2005.