Hardware Realization of Chaos-based Symmetric Video Encryption

Thesis by

Mohamad A. Ibrahim

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

King Abdullah University of Science and Technology
Thuwal, Kingdom of Saudi Arabia

June 2013
The thesis of Mohamad A. Ibrahim is approved by the examination committee.

Committee Chairperson [David Keyes]
Committee Co-Chair [Khaled Salama]
Committee Member [Mohamed-Slim Alouini]
Committee Member [Tareq AlNaffouri]
ACKNOWLEDGMENTS

This work would never been possible to see the lights without first and foremost help, mercy and blessings from Allah the exalted and most generous. To him I pray and from him I seek payback and Thawab for dedicating this work to the benefit of my Ummah and humanity in general.

I would like to secondly extend my deep gratitude thankfulness and love to my great family and particularly my parents; Radia and Abdulla for their ultimate support and continuous prayers and encouragements. Special thanks go as well to all my friends who cared to ask and send their implicit inspiration messages.

I also would like to express my sincere appreciations to my supervisor, Dr. Khaled Salama and Dr. David Keyes, who granted me with the full needed support during my thesis work. Thanks also to the group I was involved in specially Mohamed Zidan and Nif Abdulwahed who have always been there whenever I needed them.

Finally, the friend who preferred not to be mentioned here and I feel like stepping up on his shoulders and I always was by presenting this work under my name alone, to him special thanks. Also a dedication must go to all engineers and researchers who work hard to make this planet a better livable place.
This thesis reports original work on hardware realization of symmetric video encryption using chaos-based continuous systems as pseudo-random number generators. The thesis also presents some of the serious degradations caused by digitally implementing chaotic systems. Subsequently, some techniques to eliminate such defects, including the ultimately adopted scheme are listed and explained in detail. Moreover, the thesis describes original work on the design of an encryption system to encrypt MPEG-2 video streams. Information about the MPEG-2 standard that fits this design context is presented. Then, the security of the proposed system is exhaustively analyzed and the performance is compared with other reported systems, showing superiority in performance and security. The thesis focuses more on the hardware and the circuit aspect of the system’s design. The system is realized on Xilinx Vetrix-4 FPGA with hardware parameters and throughput performance surpassing conventional encryption systems.
# Table of Contents

Acknowledgments ............................................................................................................. 4
Abstract .............................................................................................................................. 5
List of Abbreviations .......................................................................................................... 8
List of Illustrations ............................................................................................................. 9
List of Tables ...................................................................................................................... 10
Introduction ....................................................................................................................... 11
I. Overview ....................................................................................................................... 11
II. Video Encryption .......................................................................................................... 12
III. Motivation and Contribution ...................................................................................... 14
IV. Thesis Organization .................................................................................................... 15
Chapter One ...................................................................................................................... 16
I. Some Issues of Video Encryption ............................................................................... 16
   a. Security ..................................................................................................................... 16
   b. Computational Complexity ..................................................................................... 18
   c. Invariance of Compression Ratio ......................................................................... 18
   d. Format Compliance ................................................................................................. 19
   e. Demand to Real-Time ............................................................................................. 19
   f. Scalability ................................................................................................................ 19
   g. Error Tolerance ........................................................................................................ 20
II. MPEG-2 Standard ....................................................................................................... 20
   a. The Layered Hierarchy of MPEG-2 ..................................................................... 21
   b. MPEG-2 Video Compression ............................................................................... 22
   c. MPEG-2 Bitstream Syntax .................................................................................... 25
Chapter Two ..................................................................................................................... 27
I. Chaos and Encryption ................................................................................................. 27
   a. Introduction to Chaos ............................................................................................. 27
   b. Chaos-based Encryption ....................................................................................... 29
II. Chaos in Video Encryption (Literature Survey) .......................................................... 31
a. Encryption of Raw (Uncompressed) Video Data ........................................ 31
b. Encryption during MPEG Video Compression ........................................ 32
c. Encryption Compressed Video Data ....................................................... 32

III. Digital Implementation of Chaos ............................................................. 33
   a. Hardware Realization ........................................................................... 33
   b. Defects in Digital Chaos ..................................................................... 34
   c. Post Processing of Chaos-based PRNG’s ............................................. 38

Chapter Three ................................................................................................. 42
I. Encryption Algorithm ................................................................................. 42
   a. Introduction ......................................................................................... 42
   b. The Proposed Video Encryption Algorithm .................................. 44

II. Hardware Realization ................................................................................. 47
   a. Parser Unit ....................................................................................... 48
   b. Feedback Unit .................................................................................. 49
   c. Hardware Realization Results .......................................................... 49

III. System Analysis ........................................................................................ 52
   a. Key Analysis ..................................................................................... 53
   b. Histogram Analysis ........................................................................... 55
   c. Chi-square Analysis ($\chi^2$) .............................................................. 57
   d. Entropy Analysis .............................................................................. 58
   e. PSNR Analysis .................................................................................. 59
   f. Fractional Dimension (FD) Analysis ................................................. 60
   g. Error Robustness Analysis ................................................................. 62
   h. Scalability Analysis .......................................................................... 63
   i. Discussion ......................................................................................... 66

Conclusion ....................................................................................................... 70

References ..................................................................................................... 71
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>FD</td>
<td>Fractal Dimension</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>HDTV</td>
<td>High Definition TV</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square error</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Lyapunov Exponent</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Pictures Experts Group</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institutes of Standard and Technology</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Transportation Safety Committee</td>
</tr>
<tr>
<td>PRNG</td>
<td>Pseudo Random Number Generator</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Sig MSE</td>
</tr>
<tr>
<td>RSA</td>
<td>Ron Rivest, Adi Shamir and Leonard Adleman Algorithm</td>
</tr>
<tr>
<td>UACI</td>
<td>Unified Average Changing Intensity</td>
</tr>
<tr>
<td>USC-SIPI</td>
<td>University of Southern California – Signal and Image Processing Institute</td>
</tr>
</tbody>
</table>


LIST OF ILLUSTRATIONS

Figure 1: encryption/decryption process of a cryptosystem .................................................. 11
Figure 2: the hierarchical structure of mpeg-2 standard .................................................... 21
Figure 3: types of mpeg-2 frames ......................................................................................... 23
Figure 4: forward/backward prediction ................................................................................. 24
Figure 5: mpeg-2 system headers in its binary stream ......................................................... 25
Figure 6: recognizing frame type from mpeg-2 binary stream .............................................. 26
Figure 7: attractors’ projection snapshotted from the oscilloscope ....................................... 35
Figure 8: histogram of the output x, y and z ...................................................................... 37
Figure 9: auto-correlation coefficient of the system ............................................................. 37
Figure 10: histogram of the output after post processing ..................................................... 41
Figure 11: auto-correlation coefficient of the post processed output .................................... 41
Figure 12: encryption/decryption naïve process .................................................................. 43
Figure 13: encryption/decryption selective process ............................................................... 43
Figure 14: selective encryption ........................................................................................... 45
Figure 15: overall view of the encryption system ................................................................ 46
Figure 16: hardware design of the encryption system ......................................................... 47
Figure 17: simulation results of input video stream and output video stream from both encryptor and decryptor .................................................................................. 51
Figure 18: test of key’s sensitivity. (a) Plain video frame encrypted with k=1000, (b) The wrongly decrypted video with k=1001, (c) the wrongly decrypted video with 999 and (d) Video frame correctly decrypted with 1000 ............................................ 54
Figure 19: uncompressed digital video encryption ................................................................ 56
Figure 20: transmission error experiment setup .................................................................... 62
Figure 21: test of error robustness ......................................................................................... 63
Figure 22: mpeg-2 video encryption ....................................................................................... 64
Figure 23: selecting frame types for encryption .................................................................... 65
Figure 24: PSNR comparison between this system and other systems ............................... 68
Figure 25: FD Comparison between this system and other systems ................................. 68
10

LIST OF TABLES

Table 1: MPEG-2 main system headers................................................................. 25
Table 2: comparison between chaos systems and encryption systems .................. 30
Table 3: Von Neumann mapping technique for balancing the distribution............... 39
Table 4: synthesis report for different array sizes in the encryptor ....................... 50
Table 5: synthesis report for different array sizes in the decryptor....................... 50
Table 6: uncompressed digital video list............................................................... 52
Table 7: correlation coefficient between selected plain frames from the uncompressed digital video and their corresponding wrongly decrypted ones....... 55
Table 8: entropy values of for the original and encrypted selected frames from uncompressed digital video............................................................... 59
Table 9: PSNR of encrypted and unencrypted videos ........................................... 60
Table 10: fractal dimension results of encrypted and unencrypted videos ............. 61
Table 11: hardware comparison between the proposed system and other conventional ones......................................................................................... 67
Table 12: comparison between different chaos-based video encryption systems.
INTRODUCTION

I. Overview

Cryptology is the science concerned with the study of hiding specific information while being transmitted between two ends, i.e., sender and receiver, and can be divided into cryptography and cryptanalysis. Cryptography studies design of efficient (fast and secure) cryptosystems, including encryption algorithms, and cryptanalysis concerns with exploiting weaknesses a cryptosystem may have.

Generally speaking, cryptosystems are modeled to have encryption/decryption systems, which are sometimes called cipher/decipher systems. Encryption/decryption systems are based on an encryption/decryption functions which could be denoted as $E$ and $D$, respectively. An original message ($m$) for encryption is called plain text and the encrypted message ($c$) is called ciphered text. The encryption process could be defined as $E_{K_e}(m) = c$, where $K_e$ is the encryption key. Likewise, the decryption is defined function as $D_{K_d}(c) = m$, where $K_d$ is the decryption key. When $K_e = K_d$, the encryption algorithm is said to be symmetric and the need to have a separate channel for keys distribution is needed. In contrast, in non-symmetric encryption $K_e$ is called the public key and it is published for public but $K_d$ is called the private key and is kept hidden. In this case there is no need for a separate channel.

![Figure 1: encryption/decryption process of a cryptosystem](image-url)
In most common cryptosystems, the encryption algorithm is assumed to be known to attackers who intend to crack the encrypted message \[1\]. Therefore, the security of a cryptosystem system depends primarily on \(K_0\).

Cryptanalysis, however, is the science concerned with studying and exploiting weaknesses of a cryptosystem. Based on the practical intelligence of the attacker, cryptanalysis designs some attacks trying to recover a plaintext out of its respective ciphered one. The resistance of a cryptosystem to such attacks is considered a metric to evaluate its security. Therefore, a cryptosystem that survives more of such attacks is generally said to be more secure.

II. Video Encryption

In recent years, the rapid development of the internet has catalyzed generation of large amount of digital video content. With the advancement in communication technology, the quantity and popularity of video content on the internet is increasing exponentially. According to \[2\], YouTube was the most accessible video web host from Google in 2012. As for the recent statistics \[3\], YouTube obtain more than 1 billion unique users’ visits each month watching over than 4 billion hours of videos. In addition to, 72 hours of video being uploaded to YouTube every minute. This huge booming in video content was eased by the availability of distribution channels as well as recording devices such as mobile phones and handheld mobile devices. Several video interactive services like pay-TV, video conferencing and YouTube streaming have found their way to end users through open channels like the World Wide Web (www). Nevertheless, most of
these channels are open and insecure which makes them unsuitable to distribute valuable video data [4]. This necessitates encrypting video content before sending it over these channels for the purposes of protecting it and making it less vulnerable to attacks.

Since 1990s, there have been several attempts to introducing encryption algorithms to investigate solutions to encrypt video data. However, many algorithms proposed to encrypt textual data such as Data Encryption Standard (DES) [5], Advanced Encryption Standard (AES) [6] and RSA [7], appear not to be practical to encrypt video data for the following reasons:

1- Video data are usually naturally bulky and large in size and therefore require more processing time for encryption. For example, an one-hour MPEG-1 video has size of approximately 0.5 GB which takes an AES algorithm half an hour to be encrypted running on a 2.6 GHz CPU [8]. This time overhead makes it impossible to satisfy the real-time requirement of video streaming.

2- Digital videos have high redundancy due to the fact that neighboring pixels in one frame as well as in consecutive frames are likely to have the same values. Encrypting video data with conventional encryption algorithms before compression removes this redundancy and hence decreases compression ratio.

3- Fully encrypting video data after compression, on the other hand, destroys video format and makes it undecodable at the receiver side. This will violate the format compliance requirement explained in chapter 2.

4- For some practical applications such as pay-per-view, there are no expensive attacks have been conducted to crack such encrypted videos. Adversaries are
not interested in paying high cost for such low value information. Military videos, however, contain very sensitive and high value information and thus require heavier encryption scheme. Conventional encryption algorithms fail to treat different video applications with different data cost and sensitivity.

III. Motivation and Contribution

From the previously mentioned points we can see that encrypting video data is not a mere application of existing encryption algorithms. Video encryption requires special design and deep analysis to satisfy typical video encryption requirements. Recently, there have been extensive efforts in trying to exploit chaos theory in encryption applications [9-16]. Chaos theory provides encryption systems with two main advantages; first, the chaotic signal looks like a noise for a non-authorized user who is ignorant about the source of generating it. Second, chaotic signal and its time evolution depend strictly on initial conditions and control parameters of the equation generating it. Strictly speaking, a slight change in the initial conditions or control parameters will result in a completely different chaotic signal, as we shall elaborately see in chapter 3. In addition, to the fact that generating chaos signal is practically cheap, so utilizing chaos encryption for video application has been of high interest up to now.

Video encryption using software platforms is not practical for standalone applications as the case with mobile phones due to the large area a microprocessor may consume. Using hardware platforms for such applications, on the other hand, is more efficient in terms of area, time and power and also provides simple yet secure
solution. To our best knowledge there has been no work conducted on hardware realization trying to exploit the advantages of chaos theory in video encryption.

**IV. Thesis Organization**

This thesis is a presentation of an original work in the area of digital video encryption utilizing chaos theory attempting to build secure and robust encryption system. Due to its wide practical applications, the MPEG-2 digital video standard was adopted for our design however; it could easily be generally customized to other MPEG standards. The thesis focuses more on the circuit design aspects and hardware realization of the encryption system in addition to presenting a complete security analysis of the output. Moreover, the results are compared with similar reported systems in the literature and overall performance superiority is demonstrated. The encryption system is written in VHDL and synthesized on Xilinx Virtex-IV FPGA.

The rest of this thesis is organized as follows: Chapter One introduces chaos-based video encryption in addition to listing some of the flows in digital chaos, and proposes post processing techniques for solving such defects. Chapter Two lists some practical requirements a typical video encryption system should satisfy and explains in detail the MPEG-2 standard used to build our system. Chapter Three discusses the video encryption framework side-by-side to its digital implementation, followed by analyzing the security of the proposed encryption system. Finally, the conclusion summarizes the work presented in this thesis and provides some suggestions for the future work and open research.
This chapter discusses the requirements a video encryption system should entertain. The motivation behind introducing these requirements is that video data have special characteristics that make general encryption schemes not suitable for encrypting it. Moreover, this chapter describes MPGEG-2 standard with all relative theoretical information needed to describe the encryption system introduced next.

I. Some Issues of Video Encryption

Video data have special characteristics such as large data volumes, high redundancy, and real-time restrictions. These characteristics impose some requirements on video encryption systems. This section lists and describes some of these requirements:

a. Security

Obviously, security should be the primary requirement for any encryption system including video encryption systems. In general, a video is said to be secure, if the cost of revealing its security is no less than the value of the video content itself [17]. For example, some video calls lose its value after certain time duration. So if cracking this secured video takes longer than that specific time duration then the video is said to be secured. Thus the use encrypting using chaotic systems should
guarantee secure transmission of data and storage by having the following properties:

1. **Perceptual Security**: A video encryption system should make video content perceptually unrecognized. The more random visually the system gets, the more secure it is considered to be.

2. **Key Space**: This is defined as the number and size of the encryption key(s) used in a cryptographical system. As the number and size of encryption key(s) increases, the more secure the system is, as it makes it less vulnerable to brute-force attacks[18]. As in [19], a brute-force attack is to systematically exploits the whole range of key space until the correct one is found. In chaotic encryption systems the key represents the initial state of the random number generator's registers.

3. **Key Sensitivity**: The output of a video encryption system should be sensitive towards any change in the keys supplied to the system. This means varying any bit in the video encryption system should result in a completely different encrypted video output. In chaotic video encryption systems, this depends on the encryption algorithm itself, as well as the divergence of the output stream of the chaotic generator which is directly related to the system's maximum Lyapunov exponent (MLE). As suggested by [20], two metrics could be used to measure the key's sensitivity: (1) correlation between wrongly decrypted video output and original plain video input as the key changes, and (2) Measuring distortion in video content by calculating Peak Signal to Noise
Ratio, PSNR of the encrypted video. For a given $m \times n$ frame $I$ and its encrypted version $K$, PSNR is described as:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2_i}{MSE} \right) = 20 \cdot \log_{10} \left( \frac{MAX_i}{\sqrt{MSE}} \right)$$  \hspace{1cm} \text{Equation 1}$$

$$= 20 \cdot \log_{10}(MAX_i) - 10 \cdot \log_{10}(MSE)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$  \hspace{1cm} \text{Equation 2}$$

$MAX_i$ is the maximum possible value a pixel in a frame could take on.

b. Computational Complexity

A video encryption system should introduce minimal overhead in terms of bandwidth requirements and the processing power needed. As in [21], an efficient encryption system will help ensuring a better presentation at the host end.

c. Invariance of Compression Ratio

An encryption scheme should maintain the size of video datum and do not increase its size. This is helpful in the case of storage and transmission. For some practical applications, it allows the change of compression ration but overall aims to keeping the resulting video size minimal [22].
d. Format Compliance (also called syntax-awareness [23], transcodability [21] or transparency):

Due to its large size, video data is encoded and compressed before being transmitted. After encoding video data will have certain format depending on the encoding scheme used. At the receiver side, it is desirable that the video content is still decodable even without having the knowledge about the decryption key. This could be achieved by preserving the syntax of structure of the encrypted video stream. This feature is called format compliance and is suitable to serve several applications such as decoding, playing, bit-rate conversion and digital watermarking [24]. For online streaming applications, especially those running on wireless, this feature becomes so useful to eliminate data loss or receiving error-full data [25].

e. Demand to Real-Time

Some applications, such as video conferencing, demand real-time streaming. Therefore, encryption and decryption time should not introduce delay that could affect this requirement for such applications.

f. Scalability

Scalability provides the feature if multi levels of security for an encryption system for different applications with easily accessible settings. The nature of MPEG-2 videos makes it easy affordable to have this feature, due to the hierarchal self-structure of MPEG-2. The simple realization of scalability in MPEG-2 is to encrypt partial layers or partial data blocks in a selected layer. Scalability in this
context could be seen as a mechanism to control visual distortion of video encryption.

**g. Error Tolerance**

It is desirable to be able to decrypt and decode an encrypted video stream even when transmission error(s) is/are introduced. This feature is demanded in real-time applications especially those that run over wireless communications channels [26]. In [27], a selective algorithm scheme was introduced to help provide error tolerance to the proposed encryption system.

**II. MPEG-2 Standard**

MPEG (Moving Picture Experts Group) compression standards[28] compress Video data to form small bits that are suitable for transmission and storage. The main principle behind MPEG high compression rate is that stores only the changes from one frame to another, instead of storing each entire frame. The video information is encoded using a technique called Discrete Cosine Transform (DCT). Since some of the video data are removed, MPEG is classified as a type of lossy compression. But generally speaking the diminishment of data is still imperceptible to the human eye and this makes it powerful and practical. MPEG has standardized several compression standards but the one adopted here is MPEG-2 due to its popularity in the research community and its wide range of applications. MPEG-2 standard is described next [29].
a. The Layered Hierarchy of MPEG-2

The MPEG-2 standard is explained fully in [29]. It is basically an extension to the MPEG-1 standard but it describes the generic coding of moving pictures and associated audio information. The resolutions available by MPEG-2 is of $720 \times 480$ and $1280 \times 720$ at 60 fps (frames per second), with audio integration. This is sufficient for most TV standards, including HDTV and NTSC. MPEG-2 is used by DVD-ROMs. MPEG-2 can compress a 2-hour video into gigabytes. Compared to what decompressing an MPEG-2 video stream requires, compressing videos in MPEG-2 format requires significantly more processing power. In this section we provide a brief overview of the related information to this project on MPEG-2 standard. For a more profound and detailed explanation of the mathematical concepts refer to [30].

![Figure 2: the hierarchical structure of mpeg-2 standard](image)
MPEG-2 standard organizes video data into a hierarchical scheme. At the very bottom we have blocks which contain pixel data, discrete cosine (DC) coefficients and motion vector. Macroblocks consist of several blocks and the encoding/compression process mainly takes place at these two layers. Blocks and macroblocks are grouped into bigger structures called slices and these in turn group up to form the main frame or picture as could be seen in Figure 2. For synchronization purposes pictures could optionally be grouped into a bigger structures called group of pictures. Finally at the top layer of the hierarchy there exists the full sequence which contains everything in addition to the starting and ending headers marking of the entire video stream.

**b. MPEG-2 Video Compression**

As stated earlier, most of the encoding/compression process happens at the block/macroblock layer which is fully contained in the slices layer. The main principle behind compression is to get rid of the redundant data in the video stream. There exist three types of redundancies in video stream which are listed below:

*Spatial Redundancy*

In MPEG-2, as many other compression schemes, reduces spatial redundancy by exploiting the fact that neighboring pixels in a frame have similar values. And to some extent a pixel value could be predicted using its neighbors. MPEG-2 utilizes two main concepts to get rid of spatial redundancy, i.e., Quantization and the two-dimensional DCT (discrete cosine transformation). DCT is done at the block level,
where it compacts redundancy to a few DC coefficients with no information loss because this process is completely reversible. Quantization is to remove information in order to obtain zero coefficients as much so it won't be transmitted. Quantization in this context is the lossy part of MPEG-2 standard because it is an irreversible process.

![Figure 3: types of mpeg-2 frames](image)

**Temporal Redundancy**

The nature of video data makes redundancy exists among neighbor pixels across consecutive series of frames this is referred to as temporal redundancy. To reduce temporal redundancy, MPEG-2 uses original I-frame in addition to motion vectors to encode only the differences between uncompressed frames and the original I-frame. This process, illustrated in Figure 3, and is called motion estimation. It does better job on the part of the frame that does not change from a certain frame to the ones coming afterwards. Motion estimation is conducted at the macroblock level and hence all motion vectors within the same macroblock possess the same value.

This concept leads to categorizing frames in MPEG-2 video stream to I, P and B frames. I-frames are independent and left un-encoded. However P and B-frames depend on I-frame or another P-frame. During the motion estimation process
described earlier, B-frames can go into both directions while P-frame can go only in the forward direction. This is illustrated clearly in Figure 4:

**Psychovisual Redundancy**

The human eye has finite response to infinite details it actually exists in a scene. For example it has limited sensing ability to edges near objects or around it [30]. In addition, the human eye is more sensitive to luminance than it is to real colors. So, when encoding luminance (Y) and chrominance (CbCr, or UV) values which were transformed from red, green and blue color values of the original picture, a special treatment has to be given to luminance values because it primarily affects a human eye.

*Figure 4: forward/backward prediction*
c. MPEG-2 Bitstream Syntax

By looking at an encoded MPEG-2 video bitstream, we will find unique start code words that are used to tag specific parts of the bitstream, i.e., sequence, group of pictures, frames and slices. These code words do not happen in the bitstream otherwise and used to recognize the mentioned specific parts of stream and sometimes for synchronization purposes. Macroblocks have headers but they are not prepended with such special start code words. This is why macroblocks cannot be recognized by just looking at the bitstream except during the decoding process.

Start code words always start with the 24 bits (0x000001) and then appended with different 8 bits for each different part of the bitstream respectively. Table 1 lists those parts of the bitstream with their corresponding start code words. Figure 5 illustrates how the MPEG-2 bitstream structure looks like in general:

<table>
<thead>
<tr>
<th>Part of The Bitstream</th>
<th>Acronym</th>
<th>Start Code Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Start</td>
<td>Sq.S</td>
<td>0x000001B3</td>
</tr>
<tr>
<td>Sequence End</td>
<td>Sq.E</td>
<td>0x000001B7</td>
</tr>
<tr>
<td>Group of Pictures</td>
<td>GoP</td>
<td>0x000001B8</td>
</tr>
<tr>
<td>Frames</td>
<td>FC</td>
<td>0x00000100</td>
</tr>
<tr>
<td>Slice Min</td>
<td>FS</td>
<td>0x00000101</td>
</tr>
<tr>
<td>Slice Max</td>
<td>FS</td>
<td>0x000001AF</td>
</tr>
<tr>
<td>User Data</td>
<td>-</td>
<td>0x000001B2</td>
</tr>
<tr>
<td>Extension</td>
<td>-</td>
<td>0x000001B5</td>
</tr>
</tbody>
</table>

Figure 5: mpeg-2 system headers in its binary stream
Figure 6 shows bytes right after frame start code word which are specifically used to recognize the frame type:

<table>
<thead>
<tr>
<th>Temporal sequence Number</th>
<th>Frame type</th>
<th>VBV delay</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I=1, P=2, B=3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6: recognizing frame type from mpeg-2 binary stream**
CHAPTER TWO

This chapter gives a brief and general introduction about chaos theory as an abstract frame to dynamical systems and other systems. Moreover, it elaborates on the discussion of the applications of chaos systems in cryptography. Afterwards, the adopted chaos system in this thesis is presented and explained thoroughly. Finally, we discuss the defects introduced by digital systems to chaos output and some post processing techniques to overcome those defections.

I. Chaos and Encryption

a. Introduction to Chaos

Chaos in general is a deterministic nonlinear system that follows a random behavioral output that dynamically changes with the input. Mathematically, a dynamical system can be fully defined by a set of equations characterizing its output in addition to the all possible states this system may naturally have [31]. As stated earlier in this thesis, chaos systems are useful for several practical applications due to their deterministic feature since their behavioral output can fully be determined and described using specific equations [32]. Nevertheless, the behavioral output of the system strongly depends on the initial conditions which control how the system is exponentially going to diverge. This adds the feature of unpredictability to chaos
systems as well. In general, according to [33] chaos systems can be characterized based on:

1. Sensitivity to initial conditions. This could be measured by maximum Lyapunov exponent (MLE). MLE mathematically approximates the long-term divergence in the output of a chaotic system by the equation:

   \[ \delta S(t) = e^{\lambda t} S(\delta t) \]  

   The more positive MLE value goes the more sensitive a chaos system is considered.

2. Ergodicity. How well the phase space is covered by the dynamical orbits generated by the system.

3. Short-term predictability. Given a state to be known, short-term predictability is a measure of how well the next state could be predicted.

4. Continuity of broad-band power system.

In mathematics, chaotic dynamical systems that operates based on a chaotic map which generate an evolutionary chaotic behavior can be parameterized and characterized as continues or discrete dynamical systems [34]. Following is a description of mathematical equations that used to model the systems respectively:

\[ \frac{dx}{dt} = f(x,p), x \in X \subseteq \mathbb{R}, p \in P \subseteq \mathbb{R} \]  

Equation 4

\[ x_{n+1} = f(x_n,p), x_n \in X \subseteq \mathbb{R}, p \in P \subseteq \mathbb{R}, n = 0,1,2, ... \]  

Equation 5

Equation 4, describes continues dynamical systems where \( f \) is the function expressing the behavioral output of the system. \( X \) is the set of all possible states a
system may have and \( P \) is the set of control parameters. The curve that describes the orbit corresponds the evolution of the system is referred to as \( \varnothing(t,x) \). Likewise, Equation 5 describes discrete-time dynamical chaotic systems where \( X_n \) expresses all the states this system may have and \( n \) represents iteration index of the trajectory it follows and consists of the sequence \( x_1, x_2, x_3, \ldots \).

The strange attractor notion in chaos theory was first discovered and introduced by Lorenz in 1963 [35]. Afterwards, it was a rich and hot area for many researchers to be deployed and utilized in wide range of practical and theoretical applications. It was used in digital communications [36], cryptography [37], electrical circuits [38], fluid mechanics [39], random number generation [40] and sometimes for chemical reactions [41] and economics [42].

b. Chaos-based Encryption

The two main features of chaotic systems outlined earlier, i.e., the deterministic nature due to sensitivity to initial conditions and unpredictability due to the noise-like output make chaotic systems an attractive to be deployed for security applications. For example, due to high demand of security in communication systems, the first chaos system was utilized in communication was implemented in [33]. By providing chaos systems with controllable initial conditions, communication systems were equipped with the desired security through having several modulation schemes. Some of these systems deploy the concept of additive masking [43], shift keying [44] and message embedding [45]. Due to randomness chaotic systems provide, they have recently been heavily used
as pseudo random number generators embedded in contemporary encryption systems [40]. There are several similarities and differences between chaos systems in general and encryption systems. These are explained elaborately in [16, 46, 47] and summarized in Table 2. Both chaos systems and encryption systems have strong dependence on initial conditions which is called the encryption key in the case of encryption systems. The encryption process evolves through rounds in order to go through the desired operations of diffusion and confusion. On the other hand, iterative evolution of chaotic systems is responsible to spread the output over the entire phase space. Encryption systems can only be defined over finite set of integer while chaotic systems can take any set of real numbers. Furthermore, the notion of performance evaluation and security assessment is yet not clearly defined for chaos systems.

Table 2: comparison between chaos systems and encryption systems

<table>
<thead>
<tr>
<th>Chaos Systems</th>
<th>Encryption Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters/Initial Conditions</td>
<td>Encryption Key</td>
</tr>
<tr>
<td>Iterations</td>
<td>Rounds</td>
</tr>
<tr>
<td>Any set of Real numbers</td>
<td>Finite set of Integers</td>
</tr>
<tr>
<td>Sensitivity to initial conditions</td>
<td>Diffusion with a small change in plaintext</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Unpredictability</td>
<td>unpredictability</td>
</tr>
<tr>
<td>-</td>
<td>Security &amp; Performance</td>
</tr>
</tbody>
</table>
II. Chaos in Video Encryption (Literature Survey)

For the previously mentioned benefits of chaos systems and its applications in encryption, chaos-based video encryption has been a topic of research for the past two decades. With respect to the operations of video compression and encryption and their relation, chaos-based video encryption can be classified into three types: encryption of raw video data (before compression), encryption during the video compression and encryption after video encryption.

a. Encryption of Raw (Uncompressed) Video Data

This is to encrypt video streams before compression as frames, pixels or bytes [48-51].

Chaotic video encryption scheme (CVES) [48], is video encryption algorithm was proposed by Li et al. which utilizes multiple chaotic maps. CVES, encrypts video stream as frame by frame by XOR'ing the data stream first and then substituted by a pseudo random s-box. CVES has shown fairly good security strength against several attacks and it generates fully perceptually distorted videos. The detailed analysis of CVES also showed that CVES could be easily realized on hardware. Moreover, CVES could be extended to be applicable for real-time video applications.

Lorenz chaotic system (LCS) [49] was introduced by Kezia, et al., using a very high dimensional Loranz chaotic system for encrypting video data as frame by frame bases. They also deployed the concept of multi-key by integrating a logistic map for generating unique keys for each frame. Furthermore, LCS has moderate security
strength and it was proven to be robust against transmission errors and fits real-time video applications.

b. Encryption during MPEG Video Compression

Here the encryption process is integrated within the compression process meaning realizing encryption before moving in the entropy encoding process [52-54].

Chaos-based selective video encryption scheme (CSVE) [53], was proposed by Lian, et al. It encrypts the video partially using a 2-d coupled map lattice. The encryption happens during the compression process, i.e., after color space transformation, to be more specific. It operates at the block level that depends on a scheme called block partitioning. The security of CSVE highly depends on the chaotic sequence generated using the lattice map.

c. Encryption Compressed Video Data

Encrypting compressed video data meaning establishing the encryption process after completely finishing with compressing digital video stream.

Lian, et al., [55] built a chaotic stream encryptor adding a random feedback to the system and it was based on the discrete piecewise linear chaotic map. It encrypts the video partially considering only intra-macroblocks and motion vectors. The scheme’s security depends mostly on the chaotic random number generator and it is shown to be secure in perception and satisfies both of format compliance and error robustness requirements.
Multiple chaotic system (MCS) is an encryption algorithm designed for MPEG-2 and proposed by Qian, et al. [56]. MCS uses three chaotic maps to perform partial encryption with confusion and block permutation, where it works at a block level. The three chaotic maps are namely and respectively mention Logistic Maps [57], 2-D Baker Map [58], and 4-D hyperchaotic maps [59]. MCS encrypts only DC coefficients by conducting the XOR'ing operation meanwhile compression processing is taking place. Afterwards, the confusion and block permutation is carried out on the compressed MPEG-2 video stream.

III. Digital Implementation of Chaos

a. Hardware Realization

There have been many analog chaotic systems introduced in the literature. these analog systems include discrete chaotic systems like the Henon map [60], the Logistic maps [61, 62], and continuous chaotic systems such as Jerk [63], Chen [64] and Lorenz [65]. These analog systems range from being built using purely discrete elements [66] to completely being built using MOS technology [67, 68]. These discrete elements are used to store and calculate the system states and compare them against the theoretical values. Having those analog components in the system would make it sensitive to process variations, operating conditions and temperature. Furthermore, setting the initial conditions using these analog devices can be extremely difficult due to current leakage which may cause synchronization problems between the two chaotic oscillators setting at the two different ends. In
addition, analog implementations generally consume large on-chip area and when using them for random number generation they lose a lot of their properties such as portability, efficiency and repeatability [69].

On the other hand, digital designs come into picture to overcome these limitations they provide area and power efficiency, portability and repeatability [70]. Using registers in digital systems we can store the system states with fixed point or floating point representation and hence we can guarantee synchronization between two chaotic systems. To digitally implement continues chaotic systems [71], they need to be first be discretized and then they need to be numerically solved by mathematical techniques like: Euler, Runge-Kutta and mid-point [63, 72, 73].

b. Defects in Digital Chaos

Period of length, unpredictability, and some other statistical properties are used to evaluate the performance of PRNGs. Due to quantization errors and finite representation, digital chaos-based PRNG suffers from loss of Ergodicity and short orbits [74]. To extensively study the defects in digital chaotic based PRNG systems, we are going to consider the differential based chaotic generator introduced in [66] and digitally implemented in [72]. The system’s general equations are given as:

\[-\ddot{X} = \ddot{X} + B\dot{X} + X\]  \hspace{1cm} \text{Equation 6}

And the nonlinear element of the system is defined as
\[ B(\dot{X}) = \begin{cases} \alpha, & \dot{X} \geq 1 \\ 0, & \dot{X} < 1 \end{cases} \quad \text{Equation 7} \]

Euler numerical solution gave a good MLE positive value of 0.377 using the following set of equations:

\[
\begin{align*}
X_{t+h} &= X_t + hY_t \\
Y_{t+h} &= Y_t + hZ_t \\
Z_{t+h} &= Z_t - h(Z_t + Y_tB(Y_t) + X_t) 
\end{align*}
\quad \text{Equation 8, 9, 10} \]

where \( t \) is the time and \( h \) is the time step.

The attractor phase plots of the digital generator output \((X-Y, X-Z, Y-Z)\) are captured from the oscilloscope and presented in Figure 7, in addition to the time series of the generated output \( X \).

![Attractors' projection](image)

**Figure 7**: attractors’ projection snapshotted from the oscilloscope
The above attractors display good phase space boundedness; however it is still insufficient to draw a conclusion about the system’s randomness. Digital limitations such as finite fixed point representation results in periodic trajectories that the output will follow and hence it is classified as a pseudo random [74]. The software introduced by [75] provides a mean to quantitatively asses how chaotic a system is by calculating its MLE.

There is also another defect caused by digital systems, namely short-term predictability. Due to the fact that the most significant bits are the primary contributors on shaping the attractors, they are more responsible for the short-term predictability in chaotic systems. This is because the most significant bits have lower transition rates than the least significant ones. Consequently, this causes the distribution of pseudo random numbers generated to look unevenly distributed across its output bits. Therefore, this makes the system less chaotic, as a uniform histogram is among the features PRNGs should possess [46]. To study this on our system, an output of 1,000,000 iterations consisting of series random numbers was collected. This output is considered essentially an approximation of the probability density function of the output. Figure 8, shows the histogram of the plain output \( \{X, Y, Z\} \) as could be seen the histogram of the output is not uniformly distributed across the entire range. This statistical defect could help adversaries grasp information about the PRNG in case it is used in cryptographical purposes and hence causing the overall system to be insecure.
Autocorrelation was calculated on the collected output as a quantitative mean to examine short term predictability. Figure 9, depicts the high correlation between the system’s output which depicts the short term predictability of the system.
c. Post Processing of Chaos-based PRNG’s

Post-processing techniques are used for improving the properties and cure some statistical defects of the generated random sequence. On the other hand such techniques reduce the throughput of the generator [76]. But according to [77], hardware RNG already expresses a high throughput, but with several statistical defects, and using post-processing techniques can enhance the statistical properties of the output without much losses. One of the main improvements that post-processing introduces is reducing the bias of the generated sequences [78]. There are many post-processing techniques proposed in the literature; some of them are mentioned in the following subsections:

1. Bit Skipping

The bit skipping technique was originally introduced in [79]. In this technique every n-th bit from the original binary is used forming a new sequence. This strategy reduces the redundancy but does not reduce difference in the probabilities of the two binary symbols [78].

2. Bit Counting

Bit counting technique is introduced in [78]. In this technique every group of p-bits are summed up in a modulo-2 addition (XOR) to form a new output bit. For the original sequence is $X_1, X_2, \ldots$ and the proposed sequence is $Y_0, Y_1, Y_2, \ldots$. The process is defined as:

$$Y_{ip} = X_{ip} \oplus X_{ip+1} \oplus \ldots \oplus X_{ip+p-1}$$  \hspace{1cm} \text{Equation 11}
According to [78] the redundancy and entropy of bit skipping and bit counting are identical. But also it shows that bit counting is better. In [78] $p$ is selected such that,

$$2 \leq p \leq 6$$

3. Von Neumann

According to [40], this is the first introduced post-processing technique and it was originally introduced by J. Von Neumann in 1951. This technique converts the bit pairs as shown in the following table,

Table 3: Von Neumann mapping technique for balancing the distribution

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Discard</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Discard</td>
</tr>
</tbody>
</table>

4. Feedback Shift Registers

This technique was introduced in [80], in which linear feedback shift registers is used as a post-processing stage for RNG. An extension for the technique was introduced by the same group in [81], in which modulo-2 addition is used as the feedback of the shift register. Both techniques were tested using FIPS-140-2.
5. Fast Fourier Transform (FFT)

According to [40], Fast Fourier Transform (FFT) has been used before as post-processing techniques for obtaining uncorrelated and unbiased numbers.

Since the defects of the proposed PRNG system primarily exists in the most significant bits and only subset of the generated bits is sufficient to build our encryption system, as we shall see, hence the decision to deploy bit skipping technique for post processing. The proposed post processing technique improves distribution of the output iterations as could be seen in Figure 10. Clearly, we can see that the post processing technique has helped uniformly shaping the distribution over the entire range. Moreover, in Figure 11 we can see the system has a desired delta-like shape of auto-correlation which indicates the effectiveness of the adopted post processing technique.
Figure 10: histogram of the output after post processing

Figure 11: auto-correlation coefficient of the post processed output
This chapter reports a novel video encryption algorithm that is based on the chaos theory presented earlier. The system was realized on the hardware and the realization was presented and evaluated. Furthermore, the security of the system was thoroughly discussed and analyzed. The results have shown the system's performance and security superiority over other systems presented in the literature.

I. Encryption Algorithm

a. Introduction

The chaotic generator used in this algorithm is based on random number generator built using digital differential chaos implemented and reported in [72]. The main purpose of having this generator is to produce random number sequence with positive MLE. Other alternatives of chaos-based PRNGs could be used as well like the ones reported in [63, 66, 73]. The used generator is based on differential based chaotic generator system discussed in the previous chapter.

This random number generator is used to construct a stream sequence of chaotically generated bits used to manipulate the plain video stream. The post processing technique applied to remove bias and short-term predictability from the original chaos output and was presented in the previous chapter, too. Figure 12 and
Figure 13 show how the output sequence from the generator is used for both naïve and selective video encryption.
The selection process is conducted to encrypt parts of the video stream and leave those bit words related to video codec untouched. This happens to maintain the requirement format compliance discussed earlier. Now let us assume a consecutive byte stream sequence of plain video forming an array $M$ of size $L$. Let $M(i)$ denotes a plain video byte in the $i^{th}$ location where $i$ could range from 0 to $L$. Likewise, let $E$ denotes an encrypted video bytes array of the same size $L$. And let $E(i)$ to be the encrypted byte of the in the same $i^{th}$ location of the encrypted video stream. As shall be seen, the confusion technique used in this system ensures the continuity of encryption, i.e., each encrypted byte has a direct impact on the following to be encrypted bytes.

b. The Proposed Video Encryption Algorithm

As in [82], most video encryption algorithms in the literature modify the MPEG standard bit stream and therefore it becomes not suitable for streaming and playback on the conventional MPEG decoder and hence need a custom codec. Moreover, complex algorithms are too complicated and needs too much of processing power which invalidates the real-time requirement. Our proposed encryption algorithm here takes care of the all these concerns.

The proposed encryption algorithm is applicable to MPEG-2 video streams and can be easily customized to all be applicable to all MPEG video streams. The input to the encryption system is an MPEG-2 stream and the output is still a valid MPEG-2 stream. The encryption system has a controller module to recognize the part of the video input stream that is related to the MPEG-2 standard. That is mainly
any of the start code words mentioned in Table 1. Based on some design conditions this controller module will output these start code words untouched whereas it will encrypt all other parts of the video input stream. Figure 14 and Equation 12 below show the incorporation with the controller module with the overall system and how the output encrypted video is being generated:

\[
O(i) = \begin{cases} 
M(i), & \text{if } M(i) = Q_j \\
E(i), & \text{if } M(i) \neq Q_j
\end{cases}
\]  

Equation 12

Where \(O(i)\) is the output byte at the \(i^{th}\) position, \(M(i)\) is the input byte of video stream, \(E(i)\) is the encrypted byte and \(Q_j\) equals to any of the start code word mention in Table 1. \(E(i)\), is the encrypted version of the output and is generated after applying several operations on the input \(M(i)\). The encryption process is mathematically defined as:

\[
E(i) = \begin{cases} 
\text{Rotate}_{\text{right}}(M(i) \otimes \text{mask}, \text{ro}_{\text{int}}) \otimes D, & i = 1 \\
\text{Rotate}_{\text{right}}(M(i) \otimes \text{mask} \otimes M(i), \text{ro}_{\text{int}}) \otimes E(R(k)), & i > 1
\end{cases}
\]  

Equation 13

\[
R(k) = K \mod (2^n - 1)
\]  

Equation 14

Figure 14: selective encryption
Each byte of the plain video $M(i)$ is first bitwisely XOR’ed with the new chaotic mask coming from the $X[8:1]$ which belongs to the used chaotic generator. Then, the result is rotated chaotically by $ro_{int}$ which is a 3-bit integer obtained from the $Z[3:1]$ of the chaotic generator. Figure 15, depicts the overall encryption process more clearly.

The encrypted output stream is then stored in an array of size $2^n - 1$. At the beginning of the operation, $D$ is initialized by zero as the array does not hold values yet. Once the array started getting populated, $R(K)$ is used as random feedback to index the array and is controlled by $k$ which is the $Y[8:1]$ output of the PRNG. As can be seen, the feedback process depends on the changes with $i$ which is the order of the randomly generated output. This feedback operation guarantees the continuity of encryption property discussed earlier in this chapter and generally makes the system more chaotic and secure.

The decryption process is the reverse order of operations of the encryption process. The decryptor manipulates the incoming encrypted video stream in the same manner as the encrypotor does to get the stream back to its original values. The decryptor first populates the array with the encrypted video stream, and then it
rotates it before doing the final XOR'ing operating. Subsequently, the operations of
the decryptor can be mathematically expressed as:

\[
M(i) = \begin{cases} 
    \text{mask} \otimes \text{Rotate}_{\text{left}}(E(i) \otimes D, ro_{\text{int}}), & i = 1 \\
    \text{mask} \otimes \text{Rotate}_{\text{left}}(E(i) \otimes E(R(k)), ro_{\text{int}}), & i > 1 
\end{cases}
\]

Equation 15

II. Hardware Realization

Figure 16: hardware design of the encryption system
Both the encryptor and decryptor hardware architecture are described in Verilog VHDL code. As Figure 16 shows, the encryptor hardware consists of chaos generator block, post processing block and control unit. The chaos generator's design was fully explained in chapter II in addition to the post processing block. One advantage of abstracting chaos generator architecture as a black box is to make it possible and easy to replace this chaos generator with another one as plug and play fashion. The control unit contains two main submodules, i.e., parser unit and feedback unit in addition to basic logic blocks. The input video stream is processed one byte at a time and a general clock is used to derive the whole circuit. The decryptor architecture is almost similar to the encryptor design but exactly performing opposite operations' order.

a. Parser Unit

This parser was introduced to the encryption system in general as a mean to interface the encryptor/decryptor with the standard MPEG-2 codec. It allows encryptor to recognize the required parts of the MPEG-2 standard and replaces them with the encrypted version.

The parser unit consists of four 8-bit shift registers used to pipeline the input video stream in order to form 32-bit long word. Using 32-bit comparator, this formed 32-bit word is going to be compared against a set of stored start code words shown in Table 1. The output of this comparator is used to drive a 2 32-bit-input multiplexer. The output of the multiplexer could be either the encrypted input if the comparator outcome is zero or the original input if otherwise.
The same concept could be further extended, as we shall see later in this chapter, to exploit more of performing selective encryption on MPEG-2 standard video streams. Selecting to encrypt I-frames only, for example, is done a similar mean as this parser is implemented. This is conducted by only editing the set of rules fed to the comparator in addition to adding some simple extra logic to the integration.

b. Feedback Unit

Feedback unit consists of an array of size \((2^n-1)\) of shift registers each of size 8 bits. This \(n\) is a design parameter; the higher it goes the larger the area and complexity of the design will be but more secure it gets. At each clock cycle the encrypted output is ready and pipelined into this array of registers. Based on the value of \(Y[8:1]\) the feedback unit will index this array and feedback to XOR'ing the value of the specific indexed register. The design parameter \((n)\) could range from 1 to 8 and it is limited by the size of output chaotic sequence taken from \(Y\).

c. Hardware Realization Results

This system was implemented using Xilinx ISE 12.3 package and synthesized on FPGA Vertix IV. Table 4 and Table 5 below shows synthesis report for both the encryptor and decryptor and for different \(n\) sizes. As can be seen, the size of the system is directly proportional to the array size as the memory area requirements increases. This results in decreasing the frequency operation and the processing speed in general. Since the decryption is the reverse process of encryption the input
gets pipelined in the array first which helps moving some of the bottlenecks to the last and hence making the whole process execute faster. Figure 17, shows the output signal from the encryptor and the decryptor for an array size of 256 simulated on Xilinx ISE Design Suit 11.1 software package.

Table 4: synthesis report for different array sizes in the encryptor

<table>
<thead>
<tr>
<th>Components utilization</th>
<th>Array Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Slices</td>
<td>141</td>
</tr>
<tr>
<td>Slice Flip-Flops</td>
<td>133</td>
</tr>
<tr>
<td>4 input LUTs</td>
<td>261</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>215.086</td>
</tr>
<tr>
<td>Throughput (Gbits/s)</td>
<td>1.68</td>
</tr>
</tbody>
</table>

Table 5: synthesis report for different array sizes in the decryptor

<table>
<thead>
<tr>
<th>Components utilization</th>
<th>Array Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Slices</td>
<td>145</td>
</tr>
<tr>
<td>Slice Flip-Flops</td>
<td>103</td>
</tr>
<tr>
<td>4 input LUTs</td>
<td>255</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>227.009</td>
</tr>
<tr>
<td>Throughput (Gbits/s)</td>
<td>1.77</td>
</tr>
</tbody>
</table>
Figure 17: simulation results of input video stream and output video stream from both encryptor and decryptor
III. System Analysis

The encryption system is essentially designed to encrypt MPEG-2 video stream as this was assumed by the parser of the system. However, this system can be easily customized to fit any other MPEG standard by just editing the set of rules in the parser. By completely disabling the parser of the system, it will not be able to recognize the headers of MPEG-2 standard and hence will encrypt the whole video stream blindly. This method was used to modify the system to be capable of decrypting uncompressed digital videos in order to use them for those analysis related to quantitatively assessing perceptual distortion [48]. Table 8, lists all digital uncompressed videos used for the purpose of perceptually analyzing the output of this system and they are available for download through [83]. The system was realized on FPGA Virtex IV and then the outputs were collected and all the subsequent experiments were conducted using MATLAB.

Table 6: uncompressed digital video list

<table>
<thead>
<tr>
<th>Video Name</th>
<th>Ratio</th>
<th>Number of Frames</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesman (QCIF)</td>
<td>4:3</td>
<td>449</td>
<td>W176 H144</td>
</tr>
<tr>
<td>Foreman (QCIF)</td>
<td>4:3</td>
<td>300</td>
<td>W176 H144</td>
</tr>
<tr>
<td>Mobile (QCIF)</td>
<td>4:3</td>
<td>300</td>
<td>W176 H144</td>
</tr>
<tr>
<td>Football (CIF)</td>
<td>4:3</td>
<td>260</td>
<td>W352 H288</td>
</tr>
<tr>
<td>Foreman (CIF)</td>
<td>4:3</td>
<td>300</td>
<td>W352 H288</td>
</tr>
<tr>
<td>Mobile (CIF)</td>
<td>4:3</td>
<td>300</td>
<td>W352 H288</td>
</tr>
<tr>
<td>Flower garden (QCIF)</td>
<td>4:3</td>
<td>360</td>
<td>W176 H144</td>
</tr>
</tbody>
</table>
a. Key Analysis

The mathematical operations conducted by the encryption system can conceal the statical information of the video completely. However, the security depends basically on the randomness of PRNG system used. The post-processed PRNG used in our system [72] has passed all the 15 NIST sp. 800-22 tests with success rate of 100%. NIST sp 800-22 is a random statistical test used to evaluate the randomness of any PRNG [84].

For a secure video encryption system the key space should be large enough to make the system secure against brute force attacks. In this system, the number of bits composing the initial conditions of the chaotic system is considered to be key space. This makes it a total of three parameters each of 16 bits long. Therefore the system uses 48 bits in total as a key and hence the key space is \(2^{48} = 281474976710656\). This huge number makes brute force attacks infeasible to crack our system with the current available computing resources. To even make the problem harder for statistical attacker an exchange of symmetric key could be conducted periodically to avoid any leakage of information caused by studying correlation between two encrypted video streams.

A system is said to have high key sensitivity if a small change in the key results in a big difference in the encrypted video streams. Figure 18, shows a plain frame from the original uncompressed digital video first encrypted with \(K=1000\), and then decrypted with \(K=1001, K=999\) and finally \(K=1000\). As can be seen, a one
bit difference in the input key results in high perceptual difference. This experiment proof that the encryption system used in this experiment has a very highly sensitive key and therefore the statistical and differential attacks becomes difficult. Table 7, seconds this conclusion as it shows the correlation coefficient, defined in Equation 16, between original selected frames and their corresponding wrongly decrypted ones.

\[
CC = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right) \left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)}}
\]

Equation 16

Figure 18: test of key’s sensitivity. (a) Plain video frame encrypted with k=1000, (b) The wrongly decrypted video with k=1001, (c) the wrongly decrypted video with 999 and (d) Video frame correctly decrypted with 1000

Where \(A\) and \(B\) are the two frames to be compared each of size \(m \times n\) and \(\bar{A}\) and \(\bar{B}\) are the mean of the two frames respectively.
Table 7: correlation coefficient between selected plain frames from the uncompressed digital video and their corresponding wrongly decrypted ones

<table>
<thead>
<tr>
<th>Frame Number</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0024</td>
</tr>
<tr>
<td>47</td>
<td>0.0091</td>
</tr>
<tr>
<td>119</td>
<td>0.0045</td>
</tr>
<tr>
<td>254</td>
<td>-0.0026</td>
</tr>
</tbody>
</table>

b. Histogram Analysis

The histogram of a frame in a video depicts the statistical distribution of colors in this frame over whole scale or range. Frames histogram before encryption is characterized by unique sharp rises followed by sharp declines and therefore is considered a footprint for each frame shot. To prevent any information leakage, the histogram of the encrypted frame should be evenly distributed over the whole scale to appear as a white noise. Using an uncompressed digital video, we test the perceptual distortion of the encrypted video as a practical indication performance of the encryption system [48]. Figure 19, shows an even distribution of the encrypted video frame and the wrongly decrypted on compared to the original plain video frame.
Figure 19: uncompressed digital video encryption
c. Chi-square Analysis ($\chi^2$)

Chi-square ($\chi^2$) test is often used by scientists to measure the goodness of distribution fit between theoretical and experimental data [85]. In our case here we are using this test to analytically examine how close the histogram of the encrypted frame is to the theoretical uniform distribution. More specifically, the test is used to determine if there is a significance difference between the expected counts of intensity levels and the observed counts in the encrypted frame considering a uniform distribution and is mathematically defined as:

$$\chi^2 = \sum_{i=1}^{L} \frac{(O_i - E_i)^2}{E_i}$$  \hspace{1cm} \text{Equation 17}

Where $O_i$ is the observed intensity counts and $E_i$ is the expected occurrences of each intensity level. $L$ is the intensity levels in a frame and it is equal to 255 in our digital uncompressed video frames. Applying this test to the two histograms of the encrypted and wrongly encrypted frames presented in Figure 19 with significance level of 0.05 and degree of freedom 255, it is found that $\chi^2_{\text{test}} < \chi^2(255,0.05)$. This implies that the null hypothesis is not rejected and the two histograms is said to be uniformly distributed.
d. Entropy Analysis

The entropy is a measure of the predictability of a random source. An unencrypted video frame has low entropy due to high correlation between adjacent pixels and bytes which reflects its high predictability. Encrypted video data, however, should perceptually appear as random noise to avoid any information leakage. For a binary source producing $2^8$ symbols of equal probabilities and each symbol is 8 bits wide as it is the case with input video, the entropy of this source is defined as [19]:

$$\text{Entropy} = - \sum_{i=0}^{2^8} P(S_i) \cdot \log_2 P(S_i)$$  \hspace{1cm} \text{Equation 18}

Where $P(S_i)$ is the probability of a symbol $S_i$. Ideally, a source frame with entropy equals to 8 is said to be truly random.

Table 8 shows entropy values of some selected frames of the plain uncompressed digital video presented against the corresponding encrypted frames. As can be seen, the average entropy of the plain frames is 7.331 which reflect high predictability and therefore allowing adversaries to visually recognize the frame. This is expected because this value belongs to plain unencrypted frame. On the other hand, the average entropy value of the encrypted frames is 7.993 which indicates the high randomness the algorithm produces as this value is close to 8 which makes the video frames perceptually highly unpredictable.
Table 8: entropy values of for the original and encrypted selected frames from uncompressed digital video

<table>
<thead>
<tr>
<th>Frame Number</th>
<th>Plain Frame Entropy</th>
<th>Encrypted Frame Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7.3117</td>
<td>7.9932</td>
</tr>
<tr>
<td>47</td>
<td>7.3483</td>
<td>7.9927</td>
</tr>
<tr>
<td>119</td>
<td>7.4266</td>
<td>7.9933</td>
</tr>
<tr>
<td>254</td>
<td>7.2399</td>
<td>7.9926</td>
</tr>
<tr>
<td>Average</td>
<td>7.331625</td>
<td>7.99295</td>
</tr>
</tbody>
</table>

**e. PSNR Analysis**

Peak signal to noise ratio or PSNR is an engineering metric often used to often express the ratio between the maximum power of a signal and the corrupting noise added to this signal. Due to wide range signals may have the PSNR value most of the time is reported in terms of the logarithmic decibel scale (dB).

PSNR has been used as a metric to evaluate videos quality in general [86] [87] [88]. Considering various video samples for example, the PSNR values of (Y, U and V) of the encrypted and unencrypted videos are reported in Table 9. The experimental results show that the encrypted video's PSNR values are smaller by more than 25 dB than the ones reported for the unencrypted videos.
### Table 9: PSNR of encrypted and unencrypted videos

<table>
<thead>
<tr>
<th>PSNR(dB)</th>
<th>Salesman (QCIF)</th>
<th>Foreman (QCIF)</th>
<th>Mobile (QCIF)</th>
<th>Football (CIF)</th>
<th>Foreman (CIF)</th>
<th>Mobile (CIF)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unencrypted</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>36.20</td>
<td>36.47</td>
<td>33.70</td>
<td>36.39</td>
<td>37.45</td>
<td>34.03</td>
</tr>
<tr>
<td>U</td>
<td>39.69</td>
<td>41.05</td>
<td>36.40</td>
<td>39.06</td>
<td>41.42</td>
<td>37.09</td>
</tr>
<tr>
<td>V</td>
<td>40.58</td>
<td>42.63</td>
<td>36.09</td>
<td>40.00</td>
<td>44.17</td>
<td>36.09</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>8.70</td>
<td>8.46</td>
<td>7.59</td>
<td>9.64</td>
<td>8.44</td>
<td>7.41</td>
</tr>
<tr>
<td>U</td>
<td>7.45</td>
<td>10.19</td>
<td>7.60</td>
<td>8.55</td>
<td>10.32</td>
<td>7.49</td>
</tr>
<tr>
<td>V</td>
<td>9.81</td>
<td>10.28</td>
<td>7.33</td>
<td>9.99</td>
<td>10.22</td>
<td>7.10</td>
</tr>
</tbody>
</table>

**f. Fractional Dimension (FD) Analysis**

In basic geometry, the term fractional dimension refers to the ratio that provides a statistical measure of how detailed a pattern is and its changes with respect to the scale it is measured at. A fractal dimension is when a dimension can take a fractional value and it does not necessary need to be an integer. Moreover, it was previously used as a mean to measure the space-filling capacity of a pattern. This helps having a clue about how fractal scales differ than the spaces it resides in [89].

For images, applications fractal dimension is used by [90] to measure the strength of confusion of an image. Then the concept was extended to be used as a
metric to measure the perceptual distortion of to an encrypted image as a measure to the strength of image and video cryptographical systems [88, 91, 92]. Ideally speaking, the maximum value of FD can go for a 2-D surface is 3. So, the more the value of FD of a frame is close to 3 the more chaotically a video is encrypted. Table 10, displays the fractal values considering Y, U and V values of the same video samples presented in Table 9. It is found that the difference is more than 0.25 between the average FD of Y, U and V of those unencrypted videos and their corresponding encrypted version.

Table 10: fractal dimension results of encrypted and unencrypted videos

<table>
<thead>
<tr>
<th>FD</th>
<th>Salesman (QCIF)</th>
<th>Foreman (QCIF)</th>
<th>Mobile (QCIF)</th>
<th>Football (CIF)</th>
<th>Foreman (CIF)</th>
<th>Mobile (CIF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unencrypted AVG.</td>
<td>2.58</td>
<td>2.55</td>
<td>2.64</td>
<td>2.65</td>
<td>2.56</td>
<td>2.65</td>
</tr>
<tr>
<td>Encrypted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>2.8975</td>
<td>2.8971</td>
<td>2.8998</td>
<td>2.8926</td>
<td>2.8926</td>
<td>2.9121</td>
</tr>
<tr>
<td>U</td>
<td>2.8979</td>
<td>2.9003</td>
<td>2.8954</td>
<td>2.8919</td>
<td>2.8921</td>
<td>2.9101</td>
</tr>
<tr>
<td>V</td>
<td>2.8969</td>
<td>2.9001</td>
<td>2.8976</td>
<td>2.8931</td>
<td>2.8937</td>
<td>2.9111</td>
</tr>
<tr>
<td>AVG.</td>
<td>2.90</td>
<td>2.90</td>
<td>2.90</td>
<td>2.90</td>
<td>2.90</td>
<td>2.91</td>
</tr>
</tbody>
</table>
g. Error Robustness Analysis

One of the main prominent reasons to introduce special encryption systems to visual data and not to stick to conventional ones like AES is to achieve high robustness against transmission errors [26]. As Figure 20 shows, an error may be introduced to the encrypted video file meanwhile transmission causing partial or full distortion to the file. In this encryption system, an MPEG-2 video file is being treated at the slice level and hence the MPEG-2 standard is respected and preserved after encryption. This helps keeping each slice of the video stream being isolated and independent of encrypting other slices. In other words, if while transmission the encrypted video file was exposed to an error in some parts of it only the corresponding parts will get affected and the error will not be spread over the whole video stream.

This was tested by manually injecting errors to the encrypted video stream. To achieve this, one full slice was subjected to corruption and changing of its composing bits. In the decrypted video file, as a result, only the scene that this specific slice is taking part of was destroyed while the rest of video displays as originally. Figure 21, depicts the results of this test, compared to the non-segment mode system presented in [88] our system turns out to be more efficient.
h. Scalability Analysis

Using an MPEG-2 compressed version of (Flower Garden) video available for download at [93], the results of encrypting the whole video including all intra-macroblocks and motion vector signs are shown in Figure 22. As could be seen, the encryption affects the video intelligibility and degrades its quality that it becomes hardly recognizable.

Next, we exploit the scalability feature this system offers by modifying the parser it has to make it able to recognize I-frames, B-frames and P-frames and does the encryption accordingly. Figure 23, shows encryption of combinations of frame types of the compressed video. The encryption of only I-frames as Figure 23 reveals does not provide sufficient perceptual security against cipher text only attack [94].
That is when playing the whole video sequence the unencrypted P and B-frames can still uncover partially video visibility.

Figure 22: mpeg-2 video encryption
Figure 23: selecting frame types for encryption
i. Discussion

From the proceeding discussion on the proposed video encryption system's analysis, we can argue that our system satisfies most of the video encryption requirements mention in chapter I. This proposed system shows high security performance since it utilizes a very efficient PRNG and encryption algorithm. The key space and key sensitivity of the system in addition to the encryption algorithm all collectively results in a very high perceptually distorted encrypted video that looks like a noise to a human eye.

The system is initially designed to encrypt the video data completely while respecting MPEG-2 standard headers. Therefore, we can conclude that the system is format compliant since the encrypted video still could be played back on any conventional media player even before decryption. By simple modification to the system, we can recognize the frames type of an MPEG-2 video and consequently be able to encrypt the video partially. It was shown that we could select only all I-frames for encryption while maintaining the rest of the video frames untouched. This shows that the video encryption scheme is highly scalable and can cover wide range of practical applications providing multilevel of security strengths.

The efficient hardware implementation entertains the system with high processing capabilities with throughput greater than 1.7 Gb/s. With typical internet speed of 10-100 Mb/s available nowadays, we can see that the system can provide the real-time feature required by some applications. Also the byte-stream processing fashion of the system makes it suitable for wireless applications that
stream videos online to handheld devices. Table 11, shows a hardware comparison between the hardware implementation of proposed system and other hardware implementation of some well-known encryption algorithms. The results reflect the superiority of our system.

Table 11: hardware comparison between the proposed system and other conventional ones

<table>
<thead>
<tr>
<th>System</th>
<th>Area (Gc)</th>
<th>Throughput (Mb/s)</th>
<th>Throughput/Area</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES [95]</td>
<td>5398</td>
<td>311</td>
<td>0.057</td>
<td>ASIC</td>
</tr>
<tr>
<td>DES [96]</td>
<td>4536</td>
<td>974</td>
<td>0.214</td>
<td>FPGA</td>
</tr>
<tr>
<td>IDEA [97]</td>
<td>47555</td>
<td>64</td>
<td>0.00134</td>
<td>ASIC</td>
</tr>
<tr>
<td>RC-5 [98]</td>
<td>50899</td>
<td>1011.2</td>
<td>0.0198</td>
<td>FPGA</td>
</tr>
<tr>
<td>This System</td>
<td>437</td>
<td>1793.776</td>
<td>4.01</td>
<td>FPGA</td>
</tr>
</tbody>
</table>

Comparison experiments are conducted to analyze the security performance of the proposed encryption scheme and the encryption schemes in [88] and [99]. Figure 24, gives the PSNR curves of the three methods on the corresponding six video files listed in Table 6. The figure shows that encrypting videos with the proposed scheme gives lower PSNR over encrypting them with each of the methods [88] and [99]. This means that our proposed methods succeeded more in turning the original videos to look noisier to perception. Thus, our proposed method degrades the videos’ quality in perception greatly than both of the previous
methods. Figure 25, confirms the past conclusion by showing the FD curves for each of the methods compared to our proposed scheme. The proposed scheme scores on each of the videos an FD value closer to 3 than each of the two methods.

![PSNR comparison between this system and other systems](image1)

**Figure 24: PSNR comparison between this system and other systems**

![FD Comparison between this system and other systems](image2)

**Figure 25: FD Comparison between this system and other systems**
Finally, [100] provides a brief overall a comparison of different video encryption systems with respect to video requirements illustrated in chapter I. Table 12, shows this table appended to it our proposed system to show where we stand among other previously proposed systems in the literature.


<table>
<thead>
<tr>
<th>Encryption system</th>
<th>BEA</th>
<th>CC</th>
<th>FC</th>
<th>ICR</th>
<th>RT</th>
<th>TET</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVES [48]</td>
<td>Y</td>
<td>M</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>PKVE [50]</td>
<td>Y</td>
<td>H</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LCVS [49]</td>
<td>Y</td>
<td>M</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CSVE [53]</td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SEHV [54]</td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>VESC [20]</td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>VM4 [55]</td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>MCES [56]</td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td><strong>This System</strong></td>
<td>Y</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
CONCLUSION

This thesis presents a novel work done in the field of digital video encryption based on chaotic systems. An original video encryption system utilizing chaos-based pseudo random number generators was digitally designed and implemented on hardware. It was shown that the digital implementation of the chaos system has several degradation problems such as short-term predictability and un-uniform output distribution. Consequently, several post processing techniques to overcome such defects were presented including the one adopted for this thesis.

The system was described thoroughly with concentration on the hardware and circuit aspects of the design. It was primarily designed to encrypt MPEG-2 standard videos but it can be easily generalized to any MPEG video. A brief overview to MPEG-2 standard was accompanied with this thesis. Finally, several analysis tests were conducted to prove that the proposed system satisfies general video encryption requirements in addition to some comparison tests showing system’s superiority over others presented in the literature.

Designing this system on ASIC could be a future work based on this thesis as it reduces on-chip area and provides higher throughput. Integrating this encryption algorithm with the compression process is worth exploiting, in addition to considering encrypting full packaged video data along with sound systems.
REFERENCES


34. Fečkan, M., Bifurcation and chaos in discontinuous and continuous systems. 2011: Springer Berlin Heidelberg.


89. Falconer, K., **Fractal Geometry: Mathematical Foundations and Applications. 2003: Wiley.**


94. Qiao, L. and K. Nahrstedt, **A New Algorithm for MPEG Video Encryption, in Conference on Imaging Science, Systems and Technology, 1997.**

95. T. Good, M.B., **Hardware results for selected stream cipher candidates, in Workshop Record of Stream Ciphers (SASC 2007), 2007. p. 191-204.**

96. Ga, et al., **Design strategies and modified descriptions to optimize cipher FPGA implementations: fast and compact results for DES and triple-DES, in**


