SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION

By

GEETHA.P

Reg. No. 1020106005

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University, Coimbatore)

COIMBATORE - 641049

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

APRIL 2012

BONAFIDE CERTIFICATE

Certified that, this project report entitled "**SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION**" is the bonafide work of Ms.Geetha.P [Reg. no. 1020106005] who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide	Head of the Department
Prof.S.Govindaraju	Dr. Ms. Rajeswari Mariappan

The candidate with university Register no. 1020106005 is examined by us in the project viva-voce examination held on ……

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I express my profound gratitude to our director **J.Shanmugham**, for giving this opportunity to pursue this course

At this pleasing moment of having successfully completed the project work, I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved Principal **Prof.Ramachandran,** for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr.Rajeswari Mariyappan Ph.D.,** the ever active, Head of the Department of Electronics and Communication Engineering, who rendering us all the time by helps throughout this project.

I extend my heartfelt thanks to my internal guide **Prof.S.Govindaraju,** for his ideas and suggestion, which have been very helpful for the completion of this project work. His careful supervision has ensured me in the attaining perfection of work.

In particular, I wish to thank and everlasting gratitude to the project coordinator **Asst.Prof.R.Hemlatha**, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering Department for their encouragement and support throughout the course of this project.

ABSTRACT

In this project, a secure image cryptosystem based on chaos and Brahmagupta and Bhaskara (BB) equation is designed and its hardware architecture are proposed. In a chaotic binary sequence, the gray level of each pixel is XORed or XNORed bit-by-bit to one of the two predetermined keys. This new scheme is used to modify the pixel values with pseudorandom sequences generated by linear feedback shift register (LFSR). Based on the characteristics of the pseudo-random, unpredictability of chaotic sequences, extreme sensitivity to the initial conditions are very high. The simulation results show that the image encrypted by this new scheme cannot be recognized. Its features are low computational complexity, high security and no distortion. In order to implement the algorithm, VLSI architecture with low hardware cost, high computing speed, and high hardware utilization efficiency is also proposed. Further the proposed cryptosystem is compared with the existing Image Encryption Using Transformation Algorithm(IETA).

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF ABBREVIATIONS	xi
1	**INTRODUCTION**	1
	1.1 Overview of the Project	2
	1.2 Introduction to VHDL	3
	1.2.1 Structural Descriptions	3
	1.2.2 Dataflow Descriptions	5
	1.2.3 Behavioral Descriptions	5
	1.3 Softwares used	7
	1.4 Project Goal	7
2	**CRYPTOSYSTEM**	7
	2.1 Cryptosystem	7
	2.2 Cryptography	7
	2.3 Fundamentals of cryptography	7
	2.4 Modern cryptography	8
	2.4.1 Symmetric key cryptography	8
	2.4.2 Public key cryptography	8
	2.5 Cryptanalysis	9
	2.6 Particularities of Image Encryption	11
	2.7. Existing Image Encryption Schemes	14
	2.8. Chaos-Based Encryption Schemes	16
	2.8.1. Basic Features of Chaos	17
	2.8.2. Definition of discrete chaos	17
	2.8.3. Relationships Between Chaos and Cryptography	18
	2.8.4. Chaos for Cryptography	18

	2.9 Chaos based Image Encryption	19
	2.10 BB Equation overview	19
	2.10.1 Application of BB equation in cryptography	20
3	**LITERATURE SURVEY**	
	3.1 IMAGE ENCRYPTION ALGORITHMS	21
4	**IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM**	25
	4.1 Description of the transformation algorithm	25
	4.2 Algorithm for creating transformation table	27
	4.3 Algorithm for performing transformation	27
5	**ENCRYPTION & DECRYPTION ALGORITHM**	
	5.1 Proposed algorithm for Encryption	28
	5.1.1 Architecture of Encryption Processing Unit	31
	5.2 Proposed algorithm for Decryption	32
	5.2.1 Architecture of Decryption Processing Unit	33
6	**PROPOSED CRYPTOSYSTEM**	35
	6.1 Generation of Pseudorandom sequence generator	36
	6.1.1 Uses in Cryptography	37
7	**RESULTS & DISCUSSION**	38
	7.1 Simulation Results	40
	7.1.1 Simulation result of existing IETA algorithm	40
	7.1.2 Simulation result of Proposed Encryption Algorithm	41
	7.1.3 Simulation result of Proposed Decryption Algorithm	42

	7.2 Synthesis Results	43
	7.2.1 Power Report	43
	7.2.1.1 Existing IETA Algorithm	43
	7.2.1.2 Proposed Chaos & BB Equation	44
	7.2.2 Area Report	45
	7.2.2.1 Existing IETA Algorithm	45
	7.2.2.2 Proposed Chaos & BB Equation	46
	7.2.3 Delay Report	47
	7.2.3.1 Existing IETA Algorithm	47
	7.2.3.2 Proposed Chaos & BB Equation	47
	7.3 Comparison	48
8	**RESULTS**	
	8.1 Image Results of Existing IETA	49
	8.2 Image Results of Proposed Chaos & BB Equation	51
	8.2.1. Results for Grayscale image	50
	8.2.2. Results for Color image	51
	8.2.3. Results for fingerprint image	52
9	**CONCLUSION & FUTURE WORK**	53
	REFERENCES	54

LIST OF FIGURES

FIGURE NO	CAPTION	PAGE NO
1.1	Project flow	2
1.2	Schematic SR latch	4
4.1	General Block Diagram of Transformation Algorithm	26
5.1	Architecture of the Encryption Unit	30
5.2	Cascade Architecture of EPE	31
5.3	Architecture of EPE 1	31
5.4	Architecture of EPE 2	31
5.5	Cascade Architecture of DPE	33
5.6	Architecture of DPE 1	34
5.7	Architecture of DPE 2	34
6.1	Proposed Cryptosystem	35
6.2	8 bit LFSR Structure	36
7.1	Conceptual Overview of Modelsim	38
7.2	Simulation Result of IETA Algorithm	40
7.3	Simulation Result of Proposed Encryption Algorithm	41
7.4	Simulation Result of Proposed Encryption Algorithm	42
7.5	Power Report of IETA Algorithm	43
7.6	Power Report of Proposed Cryptosystem	44
8.1	Lena image	49
8.2	Encrypted Image	49
8.3	Decrypted Image	49
8.4	Lena image	50
8.5	Encrypted Image	50
8.6	Decrypted Image	50
8.7	Lena image	51

8.8	Encrypted Image	51
8.9	Decrypted Image	51
8.10	Lena image	52
8.11	Encrypted Image	52
8.12	Decrypted Image	52

LIST OF TABLES

TABLE NO	CAPTION	PAGE NO
1	Similarities and differences between chaos and cryptography	18
2	Comparison between IETA based Cryptosystem and proposed Cryptosystem	48

LIST OF ABBREVIATIONS

BB	--------	**Encryption Processing Element**
ISE	--------	**Integrated Software Environment**
IETA	--------	**Image Encryption using Transformation Algorithm**
EPU	--------	**Encryption Processing Unit**
EPE	--------	**Encryption Processing Element**
DPU	--------	**Decryption Processing Unit**
DPE	--------	**Decryption Processing Element**
VLSI	--------	**Very Large Scale Integration**
FPGA	--------	**Field Programmable Gate Array**
VHDL	--------	**Very High Speed Integrated Circuit Hardware Description Language**

CHAPTER 1
INTRODUCTION

The major concern while transmitting signals is the security. The security concerns are growing due to the rampant illegal data access. To protect the valuable information in many applications like medical imaging, military image database, communications and confidential video conferencing, there is a need to secure the images by the use of encryption and decryption algorithms. In such a scenario, to avoid information leakage to both active and passive attackers, encryption of the medical images is very important. The sensitivity to initial conditions and control parameters has led to the development of chaos-based encryption and decryption algorithms.

The use of chaotic signal for secure data transmission has seen a significant growth in developing chaos-based encryption and decryption algorithms. However, a number of chaos-based algorithms have been shown to be insecure. A modified chaotic key based algorithm with increased key size is developed in for improved security and VLSI architecture of it is developed and realized using Xilinx ISE VLSI software. A cryptosystem based on Brahmagupta-Bhaskara (BB) equation was vulnerable to known plaintext attacks. Equation-based approaches, with moderate size of keys, it is possible to develop algorithms with high security.

Hence, in this project based on chaos and the BB equation, new algorithms are developed for image encryption and decryption. Further, the hardware implementation and VLSI architectures of the proposed algorithms are developed and realized using Xilinx ISE VLSI software. Furthermore, the proposed algorithm is compared with the existing IETA algorithm in terms of area, speed and power.

1.1. OVERVIEW OF THE PROJECT

The block diagram of the proposed cryptosystem for encryption and decryption is shown in Figure 1.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XORed or XNORed bit-by-bit to one of the predetermined keys.

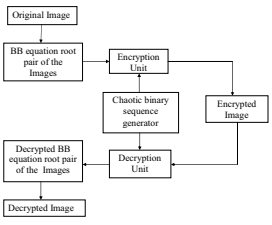

Fig.1.1: Project flow

1.2. INTRODUCTION TO VHDL

VHDL is an acronym which stands for VHSIC Hardware Description Language. VHSIC is yet another acronym which stands for Very High Speed Integrated Circuits. It is being used for documentation, verification, and synthesis of large digital designs. VHDL is a standard (VHDL-1076) developed by IEEE. The different approaches in VHDL are structural, data flow, and behavioral methods of hardware description.

1.2.1 STRUCTURAL DESCRIPTIONS

The structural descriptions are explained below with examples. Every portion of a VHDL design is considered a Block (Building Blocks). A VHDL design may be completely described in a single block, or it may be decomposed in several blocks. Each block in VHDL is analogous to an off-the-shelf part and is called an entity. The entity describes the interface to that block and a separate part associated with the entity describes how that block operates. The interface description is like a pin description in a data book, specifying the inputs and outputs to the block.

The following is an example of an entity declaration in VHDL.

```
entity latch is
    port (s,r: in bit;
          q,nq: out bit);
end latch;
```

The first line indicates a definition of a new entity, whose name is latch. The last line marks the end of the definition. The lines in between, called the port clause, describe the interface to the design. The port clause contains a list of interface declarations. Each interface declaration defines one or more signals that are inputs or outputs to the design. Each interface declaration contains a list of names, a mode, and a type.

The following is an example of an architecture declaration for the latch entity.

```
Architecture dataflow of latch is
  signal q0 : bit := '0';
  signal nq0 : bit := '1';
begin
  q0<=r nor nq0;
```



```
  nq0<=s nor q0;
  nq<=nq0;
  q<=q0;
end dataflow;
```

The first line of the declaration indicates that this is the definition of a new architecture called dataflow and it belongs to the entity named latch. So this architecture describes the operation of the latch entity. The schematic for the latch might be

Fig 1.2: Schematic SR Latch

We can specify the same connections that occur in the schematic using VHDL with the following architecture declaration:

```
Architecture structure of latch is
  component nor_gate
  port (a,b: in bit;  c: out bit);
  end component;
  begin
    n1: nor_gate
    port map (r,nq,q);
    n2: nor_gate
  port map (s,q,nq);
  end structure;
```

The lines between the first and the keyword begin are a component declaration. A list of components and there connections in any language is sometimes called a netlist. The structural description of a design in VHDL is one of many means of specifying netlists.

1.2.2 DATA FLOW DESCRIPTIONS

In the data flow approach, circuits are described by indicating how the inputs and outputs of built-in primitive components are connected together. Suppose we were to describe the following SR latch using VHDL as in the following schematic.

```
entity latch is
        port (s,r : in bit;
        q,nq : out bit);
end latch;
architecture dataflow of latch is
         begin
        q<=r nor nq;
        nq<=s nor q;
end dataflow;
```

The signal assignment operator in VHDL specifies a relationship between signals, not a transfer of data as in programming languages. The architecture part describes the internal operation of the design. The scheme used to model a VHDL design is called discrete event time simulation. In this the values of signals are only updated when certain events occur and events occur at discrete instances of time. The above mentioned SR latch works with this type of simulation.

1.2.3. BEHAVIORAL DESCRIPTIONS

The behavioral approach to modeling hardware components is different from the other two methods in that it does not necessarily in any way reflect how the design is implemented. **Since encryption and decryption are stop by stop process, it needs behavioral description in VHDL.**

A. Behavioral descriptions are supported with the Process Statements. The process statement can appear in the body of an architecture declaration just as the signal assignment statement does. The process statement can also contain signal assignments in order to specify the outputs of the process.

B. A Variable is used to hold data and also it behaves like you would expect in a software programming language, which is much different than the behavior of a signal. Although variables represent data like the signal, they do not have or cause events and are modified differently. Variables are modified with the variable assignment.

There are several statements that may only be used in the body of a process.These statements are called *Sequential Statements* because they are executed sequentially. The types of statements used here are if, if else, for and loop.

Next are about the *Signals and Processes*. This section is short, but contains important information about the use of signals in the process statement. A signal assignment, if anything, merely schedules an event to occur on a signal and does not have an immediate effect. When a process is resumed, it executes from top to bottom and no events are processed until after the process is complete.

Final Section in the behavioral description discusses about the *Program Output Methods*. In most programming languages there is a mechanism for printing text on the monitor and getting input from the user through the keyboard. It can able to give output certain information during simulation. A standard library that comes with every VHDL language system. In VHDL, common code can be put in a separate file to be used by many designs. This common code is called a library. In order to use the library that provides input and output capabilities you must add the statement use textio.all; immediately before every architecture that uses input and output. The write statement can be used to append constant values and the value of variables and signals of the types bit, bit_vector, time, integer, and real.

The security of a cryptosystem usually relies on the key only. In other words, it is assumed that the opponent knows the structure of the encryption system, has the ciphering algorithm, and has access to the transmission channel to obtain an arbitrary segment of the ciphertext c.

A good cipher should have strong ability to withstand all kinds of cryptanalysis and attacks that try to break the system. To a certain extent, the resistance against attacks is a good measure of the performance of a cryptosystem; thus, it is often used to evaluate cryptosystems.

2.4. MODERN CRYPTOGRAPHY

2.4.1. SYMMETRIC-KEY CRYPTOGRAPHY

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government. Despite its deprecation as an official standard, DES remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access.

2.4.2. PUBLIC-KEY CRYPTOGRAPHY

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. In a public-key encryption system, the *public key* is used for encryption, while the *private* or *secret key* is used for decryption.

2.5. CRYPTANALYSIS

Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

According to the method of the opponent's access to additional information, attacks on a cryptosystem may be classified into four classes:

• **Ciphertext-only attack:** Opponent has access to communication channel and can eavesdrop some segments of the ciphertext, encrypted by a certain key. The task of the opponent is to reveal as much plaintext as possible, and even to be able to deduce the cipher key.
• **Known-plaintext attack:** In addition to the obtained ciphertext segments, the opponent knows also an associated piece of plaintext. The task of the opponent is then to deduce the cipher key.
• **Chosen-plaintext attack:** The opponent not only has access to some segments of the cipher and the plaintext, but also can choose plaintext to encrypt and accordingly gets some corresponding ciphertext that he wants for comparison. This kind of attack is more intensive than the known-plaintext attack.
• **Chosen-ciphertext attack:** The opponent can choose different segments of the ciphertext and accordingly get its corresponding plaintext.

Apart from the aforementioned typical attacks, there is a type of attack named exhaustive key search, which tries all possibilities for the key in the keyspace to completely decrypt the plain message. If the keyspace of a cipher is relatively small, this exhaustive searching works quite well, given the availability of supercomputing power today. It should be emphasized that any encryption algorithm, traditional or chaos-based, should obey basic cryptographical principles in order to be able to resist serious attacks.

1.3 SOFTWARES USED

- Matlab 7 R2008b
- Xilinx ISE 9.2i

1.4. PROJECT GOAL

The goal of the project is to provide a secure cryptosystem for images. A new algorithm based on chaos and BB equation are proposed for image encryption and decryption. For practical use, VLSI architectures of the proposed algorithms are designed and realized using Xilinx ISE VLSI software for hardware implementation. Finally the proposed algorithm is compared with the existing transformation algorithm in terms of area, speed and power.

CHAPTER 2
CRYPTOSYSTEM

2.1. CRYPTOSYSTEM

A cryptographic system is any computer system that involves cryptography. Such systems include for instance, a system for secure electronic mail which might include methods for digital signatures, cryptographic hash functions, key management techniques, and so on.

Typically, a cryptosystem consists of three algorithms: one for key generation, one for encryption, and one for decryption.

2.2. CRYPTOGRAPHY

Cryptography is the study of techniques for secure communication in the presence of third parties generally, it is related to various aspects in information security such as data confidentiality, data integrity, and authentication. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

2.3. FUNDAMENTALS OF CRYPTOLOGY

The basic idea of encryption is to modify the message in such a way that its content can be reconstructed only by a legal recipient. A discrete-valued cryptosystem can be characterized by

- a set of possible plaintexts, P
- a set of possible ciphertexts, C
- a set of possible cipherkeys, K
- a set of possible encryption and decryption transformations, E and D

For each key, k ∈ K, there exists an encryption function e(k) ∈ E and a corresponding decryption function d(k) ∈ D, such that for each plaintext p ∉ P the condition for unique decoding, d(k, e(k, p)) = p, is satisfied.

2.6. Salient Features of Image Encryption

Unlike text messages, image data have special features such as high redundancy and high correlation among pixels, not to mention that they usually are huge in size, which together make traditional encryption methods difficult to apply and slow to process. Sometimes image applications also have their own requirements like real-time processing, fidelity reservation, image format consistency, and data compression for transmission. Simultaneous fulfillments of these requirements, alongwith high security and high quality demands, have presented great challenges to real-time imaging practice.

One example is the case where one needs to manage both encryption and compression. In doing so, if an image is to be encrypted after its format is converted, say from a TIFF file to a GIF file, encryption has to be implemented before compression. However, a conventional encrypted image has very little compressibility. On the other hand, compression will make a correct and loss-less decipher impossible, particularly when a highly secure image encryption scheme is used. This conflict between the compressibility and the security is very difficult, if not impossible, to completely resolve.

The salient features of image encryption may be summarized as follows:

- High redundancy and large size generally make encrypted image data vulnerable to attacks via cryptanalysis. Due to its size, the opponent can gain enough cipher text samples (even from one picture) for statistical analysis. Meanwhile, since data in images have high redundancy, adjacent pixels likely have similar grayscale values, or image blocks have similar patterns, which usually embed the image with certain patterns that result in secret leakage.

- Image data have strong correlations among adjacent pixels, which makes fast data-shuffling quite difficult. Statistical analysis on large numbers of images shows that averagely adjacent 8 to 16 pixels are correlative in the horizontal, vertical, and also diagonal directions for both natural and computer-graphical images. According to Shannon's information theory, a secure cryptosystem

should fulfill a condition on the information entropy,E(P|C) = E(P) where P stands for plain message and C for ciphered message; that is, the ciphered (i.e., encrypted) image should not provide any information about the plain image. To meet this requirement, therefore, the ciphered image should be presented as randomly as possible. Since a uniformly distributed message source has a maximum uncertainty, an ideal cipher image should have an equilibrium histogram, and any two adjacent pixels should be uncorrelated statistically. This goal is not easy to achieve under only a few rounds of permutation and diffusion.

- Huge size image data also makes real-time encryption difficult. Compared with texts, image data capacity is horrendously large. For example, a common 24-bit true-color image of 512-pixel height and 512-pixel width occupies 512 × 512 × 24/8 = 768 KB in space. Thus, a one-second motion picture will reach up to about 19 MB. Real-time processing constraints are often required for imaging applications, such as video conferencing, image surveillance, and so on. Vast amounts of image data put a great burden on the encoding and decoding processes. Encryption during or after the encoding phase, and decryption during or after the decoding phase, will aggravate the problem. If an encryption algorithm runs very slowly, even with high security, it would have little practical value for real-time imaging applications. That is the reason why current encryption methods such as DES, IDEA, and RSA are not the best candidates for this consideration.

- Image encryption is often to be carried out in combination with data compression. In almost all cases, the data are compressed before they are stored or transmitted due to the huge amount of image data and their very high redundancy. Thus, directly incorporating security requirements in the data compression system is a very attractive approach. The main challenge is how to ensure reasonable security while reducing the computational cost without downgrading the compression performance.

- In image usage, file format conversion is a frequent operation. It is desirable that image encryption not affect such an operation. Thus, directly treating image data as ordinary data for encryption will make file format conversion impossible. In this scenario, content encryption, where only the image data are encrypted, leaving file header and control information unencrypted, is preferable.

- Human vision has high robustness to image degradation and noise. Only encrypting those data bits tied with intelligibility can efficiently accomplish image protection. However, conventional cryptography treats all image data bits equally in importance, and thus requires a considerable amount of computational power to encrypt all of them, which has often proved unnecessary.

- In terms of security, image data are not as sensitive as text information. Security of images is largely determined by the real situation in an application. Usually, the value of the image information is relatively low, except in some specific situations like military and espionage applications or video conferencing in business. A very expensive attack of encrypted median data is generally not worthwhile. In practice, many image applications do not have very strict security requirements. Under certain circumstances, protection of the fidelity of an image object is more important than its secrecy. An example is electronic signatures.

Currently, there does not seem to be any image encryption algorithm that can fulfill all the aforementioned specifications and requirements. Chaos-based image encryption, can provide a class of very promising methods that can partially fulfill many of these requirements and demonstrate superiority over the conventional encryption methods, particularly with a good combination of speed, security, and flexibility.

2.7. Existing Image Encryption Schemes

Some image encryption methods have been proposed in the current literature. In order to inspire the development of better chaotic ciphers, this review is not only intended for chaos-based methods, but is also meant for understanding image encryption technology in general. Image encryption algorithms, which can be classified with respect to the approach in constructing the scheme, are divided into two groups here: chaos- based methods and non-chaos-based methods. Image encryption also can be divided into full encryption and partial encryption (also called selective encryption) according to the percentage of the data encrypted. Moreover, they can be classified into compression-combined methods and non compression methods.

Some existing proposals of chaos-based image encryption algorithms are now introduced. In this two kinds of schemes based on higher-dimensional chaotic maps were proposed. By using a discretized chaotic map, pixels in an image are permuted in shuffling after several rounds of operations. Between every two adjacent rounds of permutations, a diffusion process is performed, which can significantly change the distribution of the image histogram that makes statistical attack infeasible. Empirical testing as well as cryptanalysis both demonstrated that the chaotic baker map and cat map are good candidates for this kind of image encryption. The aforementioned schemes are block cipher, and they have some prominent merits, including high security and fast processing. However, their defects are also significant since the encrypted image has very little compressibility and is unable to abide any lossy compression (e.g., JPEG). To alleviate the conflict between compressibility and encryption, several suggestions of combining compression and encryption have been proposed. The so-called MHT scheme was proposed that enerypts image via a manipulation of Huffman coding tables in the image coding system. The MHT scheme chooses several different Huffman tables from a large number of possible candidates, and uses them alternatively to encode the image data. The choice of Huffman tables and the order in which they are used are kept secret as the key. It was advocated that the method requires very little computational overhead and can be applied to MPEG and JPEG/JPEG 2000, but it cannot resist chosen-plaintext attack.

2.8.1. Basic Features of Chaos

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behaviors. Since its discovery by Edward N. Lorenz in 1963, chaos theory has become a branch of scientific studies today. Since discrete chaotic dynamic systems (i.e., maps) are used in cryptography, this notion is briefly introduced.

2.8.2. Definition of discrete chaos

There are several definitions of chaos, which are similar but are actually not equivalent.

For simplicity, one-dimensional maps are discussed. Consider a discrete dynamical system in the general form of

$$x_{k+1} = f([x_k]), \quad f : I \longrightarrow I, \ x_0 \in I \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where f is a continuous map on the interval I = [0, 1]. This system is said to be chaotic if the following conditions are satisfied:

1. Sensitive to initial conditions:
$$\exists \delta > 0 \ \forall x_0 \in I, \varepsilon > 0 \ \exists n \in \mathbb{N}, y_0 \in I : |x_0 - y_0| < \varepsilon \Rightarrow |f^n(x_0) - f^n(y_0)| > \delta. \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$
2. Topological transitivity:
$$\forall I_1, I_2 \subset I \ \exists x_0 \in I_1, n \in \mathbb{N} : f^n(x_0) \in I_2. \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$
3. Density of periodic points in I:

Let,

$$P = \{p \in I | \exists n \in \mathbb{N} : f^n(p) = p\}$$

be the set of periodic points of f. Then P is dense in I: P = I.

2.8.3. Relationships Between Chaos and Cryptography

The similarities and differences between the two subjects can be shown in Table 1. Chaotic maps and cryptographic algorithms have some similar properties: both are sensitive to tiny changes in initial conditions and parameters; both have random like behaviors; and cryptographic algorithms shuffle and diffuse data by rounds of encryption, while chaotic maps spread a small region of data over the entire phase space via iterations. The only difference in this regard is that encryption operations are defined on finite sets of integers while chaos is defined on real numbers.

2.8.4. Chaos for Cryptography

It is natural to apply the discrete chaos theory to cryptography for the following reasons :

The property of sensitive dependence of orbits on initial conditions makes the nature of encryption very complicated. Suppose that one has the following chaos-based encryption scheme:

Chaotic systems	Cryptographic algorithms
Phase space: set of real numbers	Phase space: finite set of integers
Iterations	Rounds
Parameters	Key
Sensitivity to initial conditions and Parameters	Diffusion

Table 1. Similarities and differences between chaos and cryptography

A somewhat different chaos-based image encryption method was proposed in that makes use of the SCAN language. Through substitution of each pixel based on an additive noise vector and scramble scanning patterns, an image can be encrypted and compressed simultaneously. The idea seems to be quite good, but it was pointed out in [17] that this method is weak against exhaustive key searching and chosen-plaintext attacks. In another image compression and encryption algorithm was proposed based on the lossless quadtree image compression scheme. The quadtree data structure is used to represent the image, and the scanning sequences of image data comprise a private key for encryption. Also numerous attacks on the proposed algorithm were tested and presented, which include key space reduction, histogram attack, known-plaintext attack, and chosen-plaintext attack.

In order to speed up encryption processes so as to make them feasible for real-time applications, most of the existing schemes follow the idea of selective encryption. Actually, according to Shannon's theory, both encryption and compression are processes of redundancy reduction, but their purposes are different. Several partial encryption schemes were provided. It was reported that by a partial encryption, only 13% to 27% of the output from a quadtree compression algorithm is encrypted for a typical image, and less than 2% is encrypted for a 512×512 image compressed by set-partitioning in the hierarchical trees algorithm. There are also several proposed schemes that consider matching the compatibility to current international standards. Since many international standards on videos and images use block-based discrete cosine transform (DCT), including the familiar JPEG, MPEG-1, MPEG-2, H.261, and H.263 formats, the current research has been concentrated on selective encryptions within the framework of DCT. However, with the emergence of MPEG-4 and JPEG-2000, research emphasis may soon be redirected to a combination of encryption and wavelet compression.

To achieve high encryption speed, in the early stage some elementary cryptographic methods using random permutation lists were suggested. Since the operations are simple, the encryption does not require high computational cost. The challenge is how to achieve reasonable security with such simple operations.

2.8. Chaos-Based Encryption Schemes

A great deal of work on application of chaos to cryptography has been carried out in the last decade. Early works on chaos in cryptography were connected with encrypting messages through modulation of chaotic orbits of continuous-time dynamical systems. These methods are strongly related to the concept of synchronization of two chaotic systems and to chaos control. Several different ways have been proposed to achieve synchronization of chaotic systems, thereby transmitting information on a chaotic carrier signal.

The following technical problems were listed in :

• It is difficult to determine the synchronization time; therefore, the message during the transient period will be lost, sometimes cause fairly long transient times.

• Noise throughout the transmission significantly affects the intended synchronization. This means the synchronization noise intensity should be small compared to the signal level, or the desired synchronization will not be achieved.

• Technically, it is difficult to implement two well-matched analog chaotic systems, which are required in synchronization, and if this is not required (i.e., with certain robustness) then the opponent can also easily achieve the same synchronization for attack. In contrast to synchronization-based techniques, a direct application of a chaotic transformation to a plaintext, or applying a chaotic signal in the design of an encryption algorithm, seems to be a more promising approach.

The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

2.9. CHAOS-BASED IMAGE ENCRYPTION

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behaviors. The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

Many chaos-based encryption schemes have been proposed, and some of them have been extended from text encryption to image encryption. A direct extension of a chaos-based text encryption scheme to also work for images is possible, but this simple modification may not provide an efficient solution to these image encryption problems. Image encryption has its own specifications such as encryption speed, compatibility to image format and compression standards, and real-time implementation, therefore it requires a special design of the encryption algorithm.

An encryption method called chaotic key-based algorithm (CKBA) was proposed. The algorithm first generates a time series based on a chaotic map, and then uses it to create a binary sequence as a key. According to the binary sequence generated, image pixels are rearranged and XOR or XNOR operated with the selected key. This method is very simple but has defects in Security. This method is very weak to the chosen/known-plaintext attack with only one plain image. Moreover, its security to brute-force attack is also questionable.

2.10. BB EQUATION OVERVIEW

The Brahmagupta–Bhaskara (BB) equation is a Quadratic Diophantine equation of the form $NX^2 + k = Y^2$, Where k is an integer and N is a positive integer such that $\sqrt{N}$

is irrational. A particular case of the BB equation with k = 1 is also known as Pell equation. This equation in the Galois Field GF(p), has some useful properties. Application of these properties in two different fields of cryptography, namely, digital encryption and user authentication. For those applications, where software computation of the roots of the BB equation is unacceptable for being too slow, a hardware architecture for using the BB equation in GF(p) is given that is useful for implementation in VLSI form.

2.10.1. APPLICATION OF BB EQUATION IN CRYPTOGRAPHY

The BB-equation in Galois Field GF(p), can be written as,

$$[n(x^*)_p]_p + 1 = (y^2)_p \qquad (1)$$

where p is an odd prime.The alternative representation of BB equation in GF (p) can be rewritten as,

$$(nq_x+1)_p=(q_y)_p \qquad (2)$$

where $q_x = (x^*)_p$, $q_y = (y^2)_p$ and the subscript p stands for modulo operation by p on the argument values of the expressions. The application of the BB equation for encryption depends on the following two properties.

1. Given n and p, with p>n>0, it is always possible to obtain q_x and q_y corresponding to the roots of the BB equation $(nx^2 + 1)p = (y^2)p$.
2. Given q_x and q_y corresponding to any root of the BB equation $(nx^2 + 1)p = (y^2)p$, it is always possible to compute uniquely, the corresponding value of n , only with the knowledge of p.

The encryption process based on the BB equation is as follows.

1. n corresponds to the cleartext or plaintext in a block that is being encrypted.
2. p corresponds to the primary secret key used in the encryption of the plaintext in a block.
3. The ciphertext corresponding to n is the pair (q_x, q_y) of the corresponding BB equation.

CHAPTER 3
LITERATURE SURVEY

Aloka Sinha and Kehar Singh [18] have proposed a new technique to encrypt an image for secure image transmission.The digital signature of the original image is added to the encoded version of the original image. Image encoding is done by using an appropriate error control code, such as a Bose-Chaudhuri Hocheguenghem (BCH) code. At the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image.

S.S. Maniccam and N.G. Bourbakis [19] have presented a new methodology which performs both lossless compression and encryption of binary and gray-scale images. The compression and encryption schemes are based on SCAN patterns generated by the SCAN methodology. The SCAN is a formal language-based two-dimensional spatial-accessing methodology which can efficiently specify and generate a wide range of scanning paths or space filling curves.

Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen [20] use one of the popular image compression techniques, vector quantization to design an efficient cryptosystem for images. The scheme is based on vector quantization (VQ), cryptography, and other number theorems.In VQ, the images are first decomposed into vectors and then sequentially encoded vector by vector. Then traditional cryptosystems from commercial applications can be used.

Jiun-In Guo and Jui-Cheng Yen [21] have presented an efficient mirror-like image encryption algorithm. Based on a binary sequence generated from a chaotic system, an image is scrambled according to the algorithm. This algorithm consists of 7 steps.Step-1 determines a 1-D chaotic system and its initial point x(0) and sets k = 0.
Step-2 generates the chaotic sequence from the chaotic system.
Step-3 generates binary sequence from chaotic system.
Steps-4,5,6, and 7 rearrange image pixels using swap function according to the binary sequence.

Jui-Cheng Yen and Jiun-In Guo [22] have proposed a new image encryption scheme based on a chaotic system. In their method, an unpredictable chaotic sequence is generated. It is used to create a binary sequence again. According to the binary sequence, an image's pixels are rearranged. This algorithm has four steps.
Step-1 determines a chaotic system and its initial point x(0), row size M and column size N of the image f, iteration number no, and constants α, β and μ used to determine the rotation number.
Step-2 generates the chaotic sequence from the chaotic system.
Step-3 generates the binary sequence.
Step-4 includes special functions to rearrange image pixels. These functions are;

$ROLR_i^{\alpha,p}: f \rightarrow f'$ is defined to rotate each pixel in the ith row in f, $0 \le i \le M-1$, in the left direction p pixels if1 equals 0 or in the right direction p pixels if1 equals 1.

$ROUD_i^{\alpha,p}: f \rightarrow f'$ is defined to rotate each pixel in the ith column in f, $0 \le j \le N-1$, in the up direction p pixels if1 equals 0 or in the down direction p pixels if1 equals 1.

$ROUR_i^{\alpha,p}: f \rightarrow f'$ is defined to rotate each pixel at position (x,y) in the image f such that x+y=k, $0 \le k \le M+N-2$, , in the upper-right direction p pixels if1 is equal to 1 or in the lower-left direction p pixels if1 is equal to 0.

Shuqin Zhang and Mohammad A. Karim [23] have proposed a new method to encrypt color images using existing optical encryption systems for gray-scale images. The color images are converted to their indexed image formats before they are encoded. In the encoding subsystem, image is encoded to stationary white noise with two random phase masks, one in the input plane and the other in the Fourier plane. At the decryption end, the color images are recovered by converting the decrypted indexed images back to their RGB (Red-Green-Blue) formats. The proposed single-channel color image encryption method is more compact and robust than the multi channels methods.

Visual cryptography uses the characteristics of human vision to decrypt encrypted images. It needs neither cryptography knowledge nor complex computation. For security concerns, it also ensures that hackers cannot perceive any clues about a secret image from individual cover images. Young-Chang Hou [24] have proposed three methods for visual cryptography.

CHAPTER 4
IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM

In most of the natural images, the values of the neighboring pixels are strongly correlated (i.e. the value of any given pixel can be reasonably predicted from the values of its neighbors) . In order to dissipate the high correlation among pixels and increase the entropy value, we propose a transformation algorithm that divides the image into blocks and then shuffles their positions before it can be encrypted with the secrete key. By using the correlation and entropy as a measure of security, this process results in a lower correlation, a higher entropy value and thus improving the security level of the encrypted images. The variable secret key of the transformation process is the main key is used to increase the entropy.

4.1. DESCRIPTION OF THE TRANSFORMATION ALGORITHM

The transformation technique works as follows:

The original image is divided into a random number of blocks that are then shuffled within the image. The generated (or transformed) image is then fed to the encryption algorithm. The main idea is that an image can be viewed as an arrangement of blocks. The intelligible information present in an image is due to the correlation among the image elements in a given arrangement. This perceivable information can be reduced by decreasing the correlation among the image elements using certain transformation techniques.

The secret key of this approach is used to determine the seed. The seed plays a main role in building the transformation table, which is then used to generate the transformed image with different random number of block sizes. The transformation process refers to the operation of dividing and replacing an arrangement of the original image.The image can be decomposed into blocks; each one contains a specific number of

pixels. The blocks are transformed into new locations. For better transformation the block size should be small, because fewer pixels keep their neighbors. In this case, the correlation will be decreased and thus it becomes difficult to predict the value of any given pixel from the values of its neighbors. At the receiver side, the original image can be obtained by the inverse transformation of the blocks. A general block diagram of the transformation method is shown in Fig.4.1.

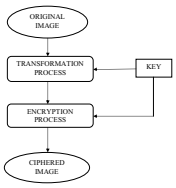

Fig.4.1. General block diagram of the transformation algorithm

Gray-level visual cryptography method first transforms the gray-level image into a halftone image and then generates two transparencies of visual cryptography. Obviously, we indeed cannot detect any information about the secret image from the two sharing transparencies individually, but when stacking them together, the result clearly shows the secret image.Method 1 uses four halftone images, cyan, magenta, yellow and black, to share the secret image. The codes of the four sharing images are fully disordered, and we cannot perceive any clue of the original secret image from any single sharing image. Method 2 reduces the inconvenience of Method 1 and requires only two sharing images to encrypt a secret image. However, after stacking the sharing images generated by Method 2, the range of color contrast will be 25% of that of the original image. Method 3 loses less image contrast, which is better than Method 2.

In cryptography, ICE (Information Concealment Engine) is a block cipher published by Kwan in 1997. The algorithm is similar in structure to DES, but with the addition of a key-dependent bit permutation in the round function. The key-dependent bit permutation is implemented efficiently in software. The ICE algorithm is not subject to patents, and the source code has been placed into the public domain.The ICE algorithm was designed for use in software applications. Those applications however are slow due to the use of modular arithmetic [25]. So the need for faster implementations is great. That can be achieved through hardware implementations. Considering the fact that hardware implementations are generally faster and more Ease of Use reliable than software implementations the outcome of a hardware design is even more interesting.

The system operates for the both encryption and decryption processes and has been optimized for low hardware resources and for high-speed performance. The proposed architecture has very encouraging performance result in terms of speed and throughput. This makes the design very useful in current applications that use DES as the base of a cryptographic protocol [25].

ICE is a Feistel network with a block size of 64 bits. The standard ICE algorithm takes a 64-bit key and has 16 rounds. A fast variant, Thin-ICE, uses only 8 rounds. An open-ended variant ICE-n, uses 16n rounds with 64n bit key. They described an attack on Thin-ICE which recovers the secret key using 223 chosen plaintexts with a 25% success probability. If 227 chosen plaintexts are used, the probability can be improved to 95%. For the standard version of ICE, an attack on 15 out of 16 rounds was found, requiring 256 work and at most 256 chosen plaintexts.

ICE is a standard Feistel block cipher [25] with a structure similar to DES. It takes a 64-bit plaintext, which is split into two 32-bit halves. In each round of the algorithm the right half and a 60-bit subkeys are fed into the function F. The output of F is XORed with the left half, and then the halves are swapped. This is the Transformation Round of the ICE algorithm. This process is repeated for 16 rounds. However the final swap is left out. The decryption process is the same, except that the sub keys are used in reverse order.

The advantages of Feistel structure are one-to- one mapping between plaintext and cipher text, which is necessary for a cipher to be decryptable. Secondly, Feistel ciphers have been publicly cryptanalysed for more than two decades, and no systematic weakness has been uncovered. And finally, Feistel ciphers are reasonably fast and simple to implement in software. Speed and simplicity were two important design aims for ICE.

4.2. ALGORITHM FOR CREATING TRANSFORMATION TABLE

1: Load Image
2: Input key
3: Randomize ()
3.1: HorizontalNoBlocks = RandomNum between
(LowerHorizontalNoBlocks and ImageWidth)
3.2: VerticalNoBlocks = RandomNum between
(LowerVerticalNoBlocks and ImageHeight)
4: NoBlocks = HorizontalNoBlocks * VerticalNoBlocks
END CREATE_TRANSFORMATION_TABLE
Input: Original Image
Output: Transformation table

4.3. ALGORITHM FOR PERFORMING TRANSFORMATION

1: For I = 0 to NoBlocks -1
1.1: Get the new location of block from the transformation table
1.2: Do XOR operation between a key and a value from the new location
1.3: Set block I in its new location
END PERFORM_TRANSFORMATION
Input: Original Image, a string key and Transformation table
Output: Transformed Image.

- For ex, if the input pixel value is FF, then from the transformation table the value corresponding to the FF can be replaced by the stored value in the table.
- After transforming, the transformed image can be encrypted by doing XOR operation with the key.

CHAPTER 5
ENCRYPTION & DECRYPTION ALGORITHM

5.1. Proposed Algorithm For Encryption

The proposed encryption algorithm is as follows.

Step 1 : Choose P, key 1 and key 2 and set j=0.

Step 2 : Choose the initial point x (0) and generate the chaotic sequence x (0), x (1), x(2), ... , x (MN/16-1) using eq.(1) and then create b (0), b (1), b (2), ... , b (2MN-1) from x(O), x (1), x (2),……, x(MN/16 - 1) by the generating scheme such that b(32i+0), b(32i+1). . . b(32i+ 29) b(32i+ 30) b(32i+ 31) . . . is the binary representation of x (i) for i = 0, 1, 2, ... (MN/16-1).

Step 3: For x=0 to M-1 ;
For y = 0 to N- 1;

obtain $q_x(x,y)$, $q_y(x,y)$ for chosen p and given f(x, y) from the solution of the BB equation,

Switch (2nb(j) + b (j+1))

Case 3: q_{xx} (x,y) = mod ((q$_x$ (x,y) + key0), 2n-l)
q_{xx} (x,y) = q_{xx} (x,y) XOR key1
q_{yy}(x,y) = mod ((q(x,y) + key1), 2n-1)
q_{yy} (x, y) = q_{yy} (x,y)XOR key1

Case 2: q_{xx}(x,y) = mod((q$_x$ (x,y) +key0), 2n-1)
q_{xx}(x,y) = q_{xx}(X,y) XNOR key1
q_{yy}(x,y) = mod ((q(x,y) +key0), 2n-1)
q_{yy}(x,y) = q_{yy} (x,y) XNOR key1

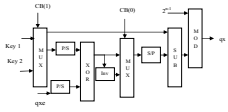

Fig.5.6: Architecture of DPE1

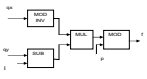

Fig.5.7: Architecture of DPE2

Case 1: q_{xx}(x,y) = mod((q$_x$ (x,y) + key2), 2n-l)
q_{xx}(x,y) = q_{xx}(x,y) XOR key2
q_{yy}(x,y) = mod ((q(x,y) + key2), 2n-1)
q_{yy} (x,y) = q_{yy} (x, y) XOR key2

Case 0: q_{xx}(x, y) = mod ((q$_x$ (x, y) +key2), 2n-1)
q_{xx}(x,y) = q_{xx}(x,y) XNOR key2
q_{yy}(x, y) = mod ((q(x, y) +key2), 2n-l)
q_{yy}(x, y) = q_{yy} (X,y) XNOR key2

j = j + 2;
End;
End;

Step 4: The result q_{xx} (x,y) ,q_{yy} (x,y) is obtained and stop the algorithm.

The proposed VLSI architectures have two key modules, one for the generation of chaotic bits (CB), and the other for encryption or decryption. The architecture of the chaotic bits (CB) generator is 32. The concept of parallel processing is adopted so that the encryption or decryption of 16 data values can be performed at the same time. Fig.5.1 shows the hardware architecture of the encryption unit (EU). This architecture consists of one 32 bit parallel-in parallel-out register, and 16 encryption processing elements (EPEs). The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input.

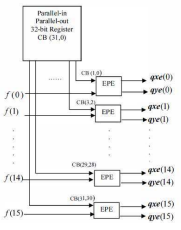

Fig.5.1. Architecture of the encryption unit

The cascade architecture of the encryption processing element (EPE) is shown in Fig.5.2. The architecture of EPE 1 is shown in Fig.5.3. It consists of three multipliers, one adder, two Mod operations and one comparator. The architecture of EPE2 is shown in Fig.5.4. It consists of two data multiplexers, two adders, two XOR gates, two MOD operations, and two inverters, four parallel-to-serial converters, and two serial-to parallel converters.

$q_y(x,y)$= mod((q$_y$ (x,y) - key2),2^{n-1})
f(x, y) = (q_x (x,y))^3 (q_y(x,y) - 1) mod(p).

j= j+ 2;
End;
End;

Step 4: The result f is obtained and stop the algorithm.

The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input. The cascade architecture of the decryption processing element (DPE) is shown in Fig.5.5

The architecture of DPE 1 is shown in Fig.5.6. The architecture of DPE2 is shown in Fig.5.7. It consists of one subtractor, one Modulo inverse operation, one multiplier and one Mod operation.

5.2.1. ARCHITECTURE OF DECRYPTION PROCESSING UNIT

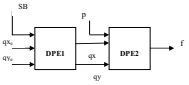

Fig.5.5: Cascade architecture of DPE

5.1.1. ARCHITECTURE OF ENCRYPTION PROCESSING UNIT

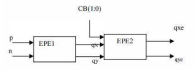

Fig.5.2. Cascade architecture of EPE

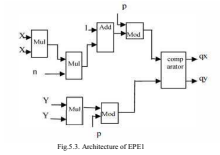

Fig.5.3. Architecture of EPE1

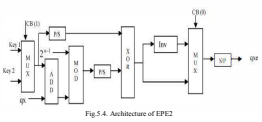

Fig.5.4. Architecture of EPE2

5.2. Proposed Algorithm For Decryption

Steps 1 and 2 are the same as in the above encryption algorithm. Steps 3 and 4 for the decryption are as follows.

Step 3: For x=0 to M-1
For y = 0 to N-1

Switch (2*b(j)) + b(j+1))

Case 3 : q_x (x,y) = q_{xx} (x,y)XOR key1
q_x (x,y) = mod(q_x (x,y) - key1),2^{n-1})
q_y (x,y) = q_{yy} (x,y)XOR key1
q_y(x,y) = mod((q_y(x,y)-key1),2^{n-1})
f(x,y) = (q_x (i))^3 (q_y(i)-1)mod(p).

Case 2 : q_x (x,y) = q_{xx} (x,y) XNOR key1
q_x (x,y) = mod(q_x (x,y) -key1),2^{n-1})
q_y(x,y)= q_{xx}(x ,y)XNOR key1
q_y(x,y)= mod((q_y(x,y)-key1),2^{n-1})
ff x,y) = (q_x(i))^3(q_y(i)-1)mod(p).

Case 1: q_x (x,y) = q_{xx} (x,y)XOR key2
q_x (x,y) = mod(q_x (x,y) - key2),2^{n-1})
q_y(x,y)= q_{xx} (x,y) XOR key2
q_y(x,y)= mod((q_y (x,y) - key2),2^{n-1})
ff(x, y) = (q_x (i))^3 (q_y (i) - 1) mod(p).

Case 0: q_x (x,y) = q_{xx} (x,y)XNOR key2
q_x (x,y) = mod(q_x (x,y) - key2),2^{n-1})
q_y(x,y)= q_{xx} (x,y) XNOR key2

CHAPTER 6
PROPOSED CRYPTOSYSTEM

It describes a cryptosystem based on the use of combined chaos and BB equation in addition to the random sequence generator for providing high security. The block diagram of the proposed cryptosystem for encryption and decryption is shown in fig.6.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XOR-ed or XNOR-ed bit-by-bit to one of the predetermined keys. Then the encrypted image again processed by pseudorandom sequence generator to provide high encryption. The key is very sensitive to initial conditions.

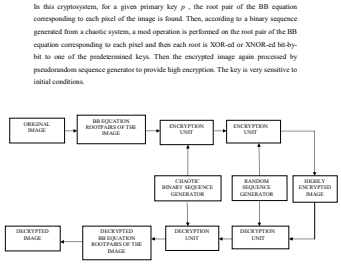

Fig.6.1: Proposed Secure Image Cryptosystem

6.1. GENERATION OF PSEUDO RANDOM SEQUENCE GENERATOR

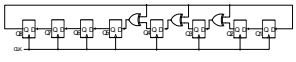

Fig.6.2: 8 bit LFSR structure

The pseudo random sequences are generated by Linear Feedback Shift Register. LFSR is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. And also LFSR is extremely sensitive to initial conditions hence finding the seed is very difficult in this case.

The Fig.8.2 shows an 8-bit LFSR structure used to generate a polynomial of $x^8 + x^6 + x^5 + x^4 + 1$ for 8 – bits. Hence it can generate 256 combinations for a bit single input.

6.1.1. Uses in cryptography

LFSRs have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very uniformly distributed output streams. However, an LFSR is a linear system, leading to fairly easy cryptanalysis. For example, given a stretch of known plaintext and corresponding ciphertext, an attacker can intercept and recover a stretch of LFSR output stream used in the system described, and from that stretch of the output stream can construct an LFSR of minimal size that simulates the intended receiver by using the Berlekamp-Massey algorithm. This LFSR can then be fed the intercepted stretch of output stream to recover the remaining plaintext.

CHAPTER 7
RESULTS & DISCUSSION

The simulation of this project has been done using MODELSIM 7 R2008b and XILINX ISE 9.1i.

ModelSim is a simulation tool for programming [VLSI] [ASIC]s, [FPGA]s, [CPLD]s, and [SoC]s. Modelsim provides a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support is provided for multiple languages including Verilog, SystemVerilog, VHDL and SystemC. The Modelsim conceptual overview is shown below.

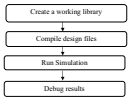

Fig.7.1: Conceptual Overview of Modelsim

In ModelSim, all designs, be they VHDL, Verilog, or some combination thereof, are compiled into a library. We can start a new simulation in ModelSim by creating a working library called "work". "Work" is the library name used by the compiler as the default destination for compiled design units. After creating the working library, we compile our design units into it.

The ModelSim library format is compatible across all supported platforms. We can simulate our design on any platform without having to recompile your design. With the design compiled, invoke the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the simulation time is set to zero, and enter a run command to begin simulation. If the results are not as expected, use ModelSim's robust debugging environment to track down the cause of the problem.

Xilinx, Inc. is an American technology company, which designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing The Virtex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores. Xilinx has offered two main FPGA families: the high-performance Virtex series and the high-volume Spartan series. With the introduction of its 28 nm FPGAs in June 2010, Xilinx replaced the high-volume Spartan family with a Kintex family and the low-cost Artix family. The Spartan series targets applications with a low-power footprint, extreme cost sensitivity and high-volume; e.g. displays, set-top boxes, wireless routers and other applications.

The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using Chip Scope Pro tools, and creation of the bit files that are used to configure the chip.

Xilinx is a synthesis tool which converts Schematic/HDL design entry into functionally equivalent logic gates on Xilinx FPGA, with optimized speed & area. So, after specifying behavioral description for HDL, the designer merely has to select the library and specify optimization criteria; and Xilinx synthesis tool determines the net list to meet the specification; which is then converted into bit-file to be loaded onto FPGA-PROM. Also, Xilinx tool generates post-process simulation model after every implementation step, which is used to functionally verify generated net list after processes, like map, place & route.

The synthesis and the simulation results of the proposed and the existing cryptosystems are shown below.

7.1. SIMULATION RESULTS

7.1.1. Simulation Result of Existing IETA Algorithm

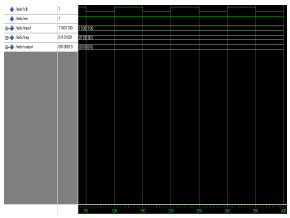

Fig.7.2: Simulation Result of IETA Algorithm

In the above figure input and key are the inputs and output can be produced according to value stored in the transformation table and the value of 8 bit key.

7.1.2. Simulation Result of Proposed Encryption Algorithm

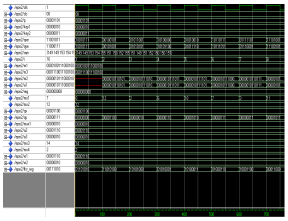

Fig.7.3: Simulation Result of Proposed Encryption Algorithm

In the above figure c_0 is the combinational register used to select the Encryption Processing Element in the encryption unit, n is the plain text and p is the secret key, key 1 and key 2 are the additional keys used in the encryption process. q_{0e} and q_{1e} are the encrypted output values. All the keys used here are 8 bits. LFSR is the feedback shift register used to provide more encryption.

7.1.3. Simulation Result of Proposed Decryption Algorithm

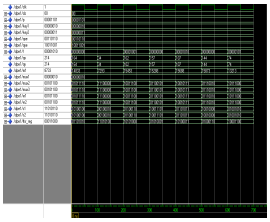

Figure.7.4. Simulation Result of Proposed Decryption Algorithm

In the above figure Encrypted image pixel values can be given as input to the Decryption Unit(DU). Here q_{0e}, q_{1e}, primary secret key p_1 key 1 key 2 are the inputs to the DU. Output pixel values are exactly same as that of input pixel values after decryption process.

7.2. SYNTHESIS RESULTS

The proposed design has been implemented on Xilinx Spartan 2E device and simulated by Xilinx 9.2i design tool. Therefore the device utilization summary and the timing summary can be obtained from the *Synthesis Report* as shown below.

7.2.1. Power Report

7.2.1.1. Image Encryption using Transformation Algorithm

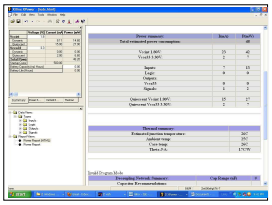

Fig.7.5: Power Report of IETA Algorithm

7.2.1.2. Proposed Cryptosystem based on Chaos & BB Equation

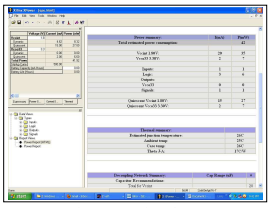

Fig.7.6: Power Report of Proposed Cryptosystem

7.2.2. AREA REPORT

7.2.2.1. Existing IETA Algorithm

Design Summary

Number of errors: 0
Number of warnings: 8

Logic Utilization:
Number of 4 input LUTs: 16 out of 13,824 2%

Logic Distribution:
Number of Slices containing only related logic: 0 out of 0 0%
Number of Slices containing unrelated logic: 0 out of 0 0%

Number of bonded IOBs: 17 out of 510 3%
Number of Block RAMs: 1 out of 72 1%
Number of GCLKs: 1 out of 4 25%
Number of GCLKIOBs: 1 out of 4 25%

Total equivalent gate count for design: 16,384
Additional JTAG gate count for IOBs: 864

7.2.2.2. Proposed Chaos & BB Combined Algorithm

Area report:
Design Summary

Number of errors: 0
Number of warnings: 2

Logic Utilization:

Number of 4 input LUTs: 8 out of 13,824 1%

Logic Distribution:

Number of occupied Slices: 4 out of 6,912 1%
Number of Slices containing only related logic: 4 out of 4 100%
Number of Slices containing unrelated logic: 0 out of 4 0%

Total Number of 4 input LUTs: 8 out of 13,824 1%
Number of bonded IOBs: 34 out of 510 6%

Total equivalent gate count for design: 48
Additional JTAG gate count for IOBs: 1,632

7.2.3. Delay Report

7.2.3.1. Image Encryption Using Transformation Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 7.548ns (Maximum Frequency: 132.481MHz)
Minimum input arrival time before clock: 8.535ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **2.319ns**

7.2.3.2. Proposed Chaos & BB Combined Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 2.956ns (Maximum Frequency: 338.295MHz)
Minimum input arrival time before clock: 2.948ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **3.268ns**

7.3. COMPARISON

SUMMARY	IETA BASED CRYPTOSYSTEM	PROPOSED CHAOS & BB BASED CRYPTOSYSTEM	% OF RESULTS
POWER	48Mw	42mW	**12.5**
SPEED	2.319ns	3.268ns	**40**
AREA Total Gate Count	16384	48	**99.7**

Table 2.Comparison between IETA based Cryptosystem and proposed Cryptosystem

CHAPTER 8
RESULTS

8.1. IMAGE RESULTS OF EXISTING IETA

INPUT IMAGE

Fig.8.1. lena image

ENCRYPTED IMAGE

Fig.8.2.Encrypted Image

OUTPUT IMAGE

Fig.8.3.Decrypted Image

8.2. IMAGE RESULTS OF PROPOSED CHAOS & BB EQUATION

8.2.1. Results for Grayscale image

INPUT IMAGE

Fig.8.4. lena image

ENCRYPTED IMAGE

Fig.8.5.Encrypted Image

OUTPUT IMAGE

Fig.8.6.Decrypted Image

8.2.2. Results for Color image

INPUT IMAGE

Fig.8.7. lena image

ENCRYPTED IMAGE

Fig.8.8.Encrypted Image

OUTPUT IMAGE

Fig.8.9.Decrypted Image

8.2.3. Results for fingerprint image

INPUT IMAGE

Fig.8.10. fp image

ENCRYPTED IMAGE

Fig.8.11.Encrypted Image

OUTPUT IMAGE

Fig.8.12..Decrypted Image

CHAPTER 9
CONCLUSION & FUTURE WORK

In this project, we proposed a robust and efficient cryptosystem to transmit a digital image in a secure way. A secure cryptosystem based on Chaos & BB equation with pseudorandom sequence is proposed for image encryption & decryption. The new quadratic equation based encryption & decryption algorithm alters the positions of the pixels by XOR and Mod operations without changing its correlation with neighbouring pixels. This method is highly sensitive to initial conditions. The VLSI architecture of the proposed cryptosystem is designed and realized using Xilinx ISE VLSI software for an image. The proposed algorithm is highly promising for high security in real time applications. The proposed algorithm also compared with existing IETA algorithm.

It should be proved that this system is secure against all kinds of attacks. Cryptanalysis of the proposed cryptosystem specifically for cipher text attacks and known plain text attacks is to be provided.

REFERENCES :

1. K.Deergha Rao and Ch. Gangadhar, "VLSI realization of a secure cryptosystem for image encryption and decryption" in Proceedings of IEEE Int. Conf. on Communication and signal processing, 2011

2. K.Deergha Rao, K.Praveen Kumar,and P. V.Muralikrishna,"A New and Secure Cryptosystem for Image Encryption and decryption", appear in IETE Journal of Research, March-Apr. 2011

3. Mohammad Ali Bani Younes and Aman Jantan, "Image Encryption Using Block-Based Transformation Algorithm" Proc. IAENG International Journal of Computer Science, Feb 2008, 35:1, IJCS_35_1_03

4. G.Alvarez L.H.Encinas, and M.Masque, "Known-Plaintext Attack to Two Cryptosystems Based on the BB Equation", IEEE Transactions on Circuits and Systems II: Express Briefs Volume 55, Issue 5, May 2008

5. R.Rhouma, S.Meherzi, S.Belghith, "CML-based colour image encryption" in International Journal of Chaos, Soltons and Fractales 2007

6. Alvrez, G. and Shujun L. "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystem" in Int. Journal of Bifurcation and Chaos, 2006

7. N.Rama Murthy, M.N.S.Swamy, "Cryptographic Applications of Brahmagupta-Bhaskara Equation", IEEE Transactions on circuits -I, 2006

8. G.Zhi-Hong, H.Fangjun, and G.Wenjie,"Chaos - based image encryption algorithm,"Department of Electrical and computer Engineering,University of Waterloo, ON N2L 3G1, Canada.Published by: Elsevier, 2005, pp. 153-157.

9. I.Ozturk, I.Sogukpinar, "Analysis and comparison of image encryption algorithm,"Journal of transactions on engineering, computingand technology December, vol. 3, 2004, p.38.

10. B.Furht and D.Socek, "Multimedia security encryption techniques" IEC Int. Engg. Consortium, Chicago, IL, pages 335-349, 2004

11. Sinha, K. Singh, "A technique for image encryption using digi tal signature," Source:OpticsCommunications,vol.218,no..2003

12. G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalyzing a discrete-time chaos synchronization secure communication system,"Chaos, Soltons &fractals, 2003,pp. 689-694.

13. S.I.Li and X.Zheng , "On the security of an image encryption method," Proc. IEEE International Conference on Image Processing (ICIP 2002), vol.2, pp. 925-928, 2002.

14. S.J.Li and X. Zheng, "Cryptanalysis of a Chaotic Image Encryption Method", IEEE International Symposium on Circuits and Systems (ISCAS 2002)

15. M.I.Sobhy, and A.R.Shehata, "Methods of attacking chaotic encryption and countermeasures," Proc. IEEE Inernational Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)

16. H.Cheng and X.Li, "Partial encryption of compressed images and videos" in IEEE Transactions on Signal Processing, volume 48, pages 2439-2451, 2000

17. Dachselt F, Schwarz W (2001) Chaos and cryptography. IEEE Trans Circuits and Systems-I 48(12):1498 – 1509

18. Aloha Sinha, Kehar Singh, "A technique for image encryption using digital signature", Optics Communications, ARTICLE IN PRESS, 2003,1-6,

19. S.S.Maniccam, N.G. Bourbakis, "Lossless image compression and encryption using SCAN", Pattern Recognition 34 (2001), 1229-1245

20. Chin-Chen Chang, Min-Shian Hwang, Chen,"A new encryption algorithm for image cryptosystems", The Journal of Systems and Software 58 (2001), 83-91

21. Jiun-In Guo, Jui-Cheng Yen, "A new mirror-like image encryption algorithm and its VLSI architecture", Department of Electronics Engineering National Lien-Ho College of Technology and Commerce,Miaoli,(2001) Taiwan, Republic of China

22. Jui-Cheng Yen, Jiun-In Guo, "A new chaotic image encryption algorithm",Department of Electronics Engineering National Lien-Ho College of Technology and Commerce, Miaoli, Taiwan,(2001) Republic of China

23. Shuqun Zhang and A. Karim, "Color image encryption using double random phase encoding",Microwave and Optical Technology, Vol.21, No.5, June 5 1999

24. Young-Chang Hou, "Visual cryptography for color images", Pattern Recognition 36 (2003), www.elsevier.com/locate/patcog, 1619-1629

25. M Kwan, ―The Design of the ICE Encryption Algorith, in proc. Of Fast software Encryption workshop, 1997.

SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION

By

GEETHA.P

Reg. No. 1020106005

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University, Coimbatore)

COIMBATORE - 641049

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

APRIL 2012

BONAFIDE CERTIFICATE

Certified that, this project report entitled "**SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION**" is the bonafide work of Ms.Geetha.P [Reg. no. 1020106005] who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide	Head of the Department
Prof.S.Govindaraju	Dr. Ms. Rajeswari Mariappan

The candidate with university Register no. 1020106005 is examined by us in the project viva-voce examination held on ……

Internal Examiner	External Examiner

ACKNOWLEDGEMENT

I express my profound gratitude to our director **J.Shanmugham.** for giving this opportunity to pursue this course

At this pleasing moment of having successfully completed the project work, I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved Principal **Prof.Ramachandran**, for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr.Rajeswari Mariyappan Ph.D.,** the ever active, Head of the Department of Electronics and Communication Engineering, who rendering us all the time by helps throughout this project.

I extend my heartfelt thanks to my internal guide **Prof.S.Govindaraju,** for his ideas and suggestion, which have been very helpful for the completion of this project work. His careful supervision has ensured me in the attaining perfection of work.

In particular, I wish to thank and everlasting gratitude to the project coordinator **Asst.Prof.R.Hemlatha**, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering Department for their encouragement and support throughout the course of this project.

ABSTRACT

In this project, a secure image cryptosystem based on chaos and Brahmagupta and Bhaskara (BB) equation is designed and its hardware architecture are proposed. In a chaotic binary sequence, the gray level of each pixel is XORed or XNORed bit-by-bit to one of the two predetermined keys. This new scheme is used to modify the pixel values with pseudorandom sequences generated by linear feedback shift register (LFSR). Based on the characteristics of the pseudo-random, unpredictability of chaotic sequences, extreme sensitivity to the initial conditions are very high. The simulation results show that the image encrypted by this new scheme cannot be recognized. Its features are low computational complexity, high security and no distortion. In order to implement the algorithm, VLSI architecture with low hardware cost, high computing speed, and high hardware utilization efficiency is also proposed. Further the proposed cryptosystem is compared with the existing Image Encryption Using Transformation Algorithm(IETA).

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF ABBREVIATIONS	xi
1	**INTRODUCTION**	1
	1.1 Overview of the Project	2
	1.2 Introduction to VHDL	3
	1.2.1 Structural Descriptions	3
	1.2.2 Dataflow Descriptions	5
	1.2.3 Behavioral Descriptions	5
	1.3 Softwares used	7
	1.4 Project Goal	7
2	**CRYPTOSYSTEM**	7
	2.1 Cryptosystem	7
	2.2 Cryptography	7
	2.3 Fundamentals of cryptography	7
	2.4 Modern cryptography	8
	2.4.1 Symmetric key cryptography	8
	2.4.2 Public key cryptography	8
	2.5 Cryptanalysis	9
	2.6 Particularities of Image Encryption	11
	2.7. Existing Image Encryption Schemes	14
	2.8. Chaos-Based Encryption Schemes	16
	2.8.1. Basic Features of Chaos	17
	2.8.2. Definition of discrete chaos	17
	2.8.3. Relationships Between Chaos and Cryptography	18
	2.8.4. Chaos for Cryptography	18

	2.9 Chaos based Image Encryption	19
	2.10 BB Equation overview	19
	2.10.1 Application of BB equation in cryptography	20
3	**LITERATURE SURVEY**	
	3.1 IMAGE ENCRYPTION ALGORITHMS	21
4	**IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM**	25
	4.1 Description of the transformation algorithm	25
	4.2 Algorithm for creating transformation table	27
	4.3 Algorithm for performing transformation	27
5	**ENCRYPTION & DECRYPTION ALGORITHM**	
	5.1 Proposed algorithm for Encryption	28
	5.1.1 Architecture of Encryption Processing Unit	31
	5.2 Proposed algorithm for Decryption	32
	5.2.1 Architecture of Decryption Processing Unit	33
6	**PROPOSED CRYPTOSYSTEM**	35
	6.1 Generation of Pseudorandom sequence generator	36
	6.1.1 Uses in Cryptography	37
7	**RESULTS & DISCUSSION**	38
	7.1 Simulation Results	40
	7.1.1 Simulation result of existing IETA algorithm	40
	7.1.2 Simulation result of Proposed Encryption Algorithm	41
	7.1.3 Simulation result of Proposed Decryption Algorithm	42

	8.8 Encrypted Image	51
	8.9 Decrypted Image	51
	8.10 Lena image	52
	8.11 Encrypted Image	52
	8.12 Decrypted Image	52

LIST OF TABLES

TABLE NO	CAPTION	PAGE NO
1	Similarities and differences between chaos and cryptography	18
2	Comparison between IETA based Cryptosystem and proposed Cryptosystem	48

	7.2 Synthesis Results	43
	7.2.1 Power Report	43
	7.2.1.1 Existing IETA Algorithm	43
	7.2.1.2 Proposed Chaos & BB Equation	44
	7.2.2 Area Report	45
	7.2.2.1 Existing IETA Algorithm	45
	7.2.2.2 Proposed Chaos & BB Equation	46
	7.2.3 Delay Report	47
	7.2.3.1 Existing IETA Algorithm	47
	7.2.3.2 Proposed Chaos & BB Equation	47
	7.3 Comparision	48
8	**RESULTS**	
	8.1 Image Results of Existing IETA	49
	8.2 Image Results of Proposed Chaos & BB Equation	51
	8.2.1. Results for Grayscale image	50
	8.2.2. Results for Color image	51
	8.2.3. Results for fingerprint image	52
9	**CONCLUSION & FUTURE WORK**	53
	REFERENCES	54

LIST OF FIGURES

FIGURE NO	CAPTION	PAGE NO
1.1	Project flow	2
1.2	Schematic SR latch	4
4.1	General Block Diagram of Transformation Algorithm	26
5.1	Architecture of the Encryption Unit	30
5.2	Cascade Architecture of EPE	31
5.3	Architecture of EPE 1	31
5.4	Architecture of EPE 2	31
5.5	Cascade Architecture of DPE	33
5.6	Architecture of DPE 1	34
5.7	Architecture of DPE 2	34
6.1	Proposed Cryptosystem	35
6.2	8 bit LFSR Structure	36
7.1	Conceptual Overview of Modelsim	38
7.2	Simulation Result of IETA Algorithm	40
7.3	Simulation Result of Proposed Encryption Algorithm	41
7.4	Simulation Result of Proposed Encryption Algorithm	42
7.5	Power Report of IETA Algorithm	43
7.6	Power Report of Proposed Cryptosystem	44
8.1	Lena image	49
8.2	Encrypted Image	49
8.3	Decrypted Image	49
8.4	Lena image	50
8.5	Encrypted Image	50
8.6	Decrypted Image	50
8.7	Lena image	51

LIST OF ABBREVIATIONS

BB	--------	**Encryption Processing Element**
ISE	--------	**Integrated Software Environment**
IETA	--------	**Image Encryption using Transformation Algorithm**
EPU	--------	**Encryption Processing Unit**
EPE	--------	**Encryption Processing Element**
DPU	--------	**Decryption Processing Unit**
DPE	--------	**Decryption Processing Element**
VLSI	--------	**Very Large Scale Integration**
FPGA	--------	**Field Programmable Gate Array**
VHDL	--------	**Very High Speed Integrated Circuit Hardware Description Language**

CHAPTER 1
INTRODUCTION

The major concern while transmitting signals is the security. The security concerns are growing due to the rampant illegal data access. To protect the valuable information in many applications like medical imaging, military image database, communications and confidential video conferencing, there is a need to secure the images by the use of encryption and decryption algorithms. In such a scenario, to avoid information leakage to both active and passive attackers, encryption of the medical images is very important. The sensitivity to initial conditions and control parameters has led to the development of chaos-based encryption and decryption algorithms.

The use of chaotic signal for secure data transmission has seen a significant growth in developing chaos-based encryption and decryption algorithms. However, a number of chaos-based algorithms have been shown to be insecure. A modified chaotic key based algorithm with increased key size is developed in for improved security and VLSI architecture of it is developed and realized using Xilinx ISE VLSI software. A cryptosystem based on Brahmagupta-Bhaskara (BB) equation was vulnerable to known plaintext attacks. Equation-based approaches, with moderate size of keys; it is possible to develop algorithms with high security.

Hence, in this project based on chaos and the BB equation, new algorithms are developed for image encryption and decryption. Further, the hardware implementation and VLSI architectures of the proposed algorithms are developed and realized using Xilinx ISE VLSI software. Furthermore, the proposed algorithm is compared with the existing IETA algorithm in terms of area, speed and power.

1.1. OVERVIEW OF THE PROJECT

The block diagram of the proposed cryptosystem for encryption and decryption is shown in Figure 1.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XORed or XNORed bit-by-bit to one of the predetermined keys.

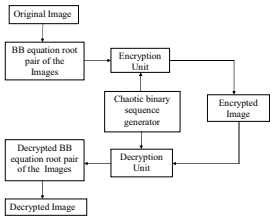

Fig.1.1: Project flow

1.2. INTRODUCTION TO VHDL

VHDL is an acronym which stands for VHSIC Hardware Description Language. VHSIC is yet another acronym which stands for Very High Speed Integrated Circuits. It is being used for documentation, verification, and synthesis of large digital designs. VHDL is a standard (VHDL-1076) developed by IEEE. The different approaches in VHDL are structural, data flow, and behavioral methods of hardware description.

1.2.1 STRUCTURAL DESCRIPTIONS

The structural descriptions are explained below with examples. Every portion of a VHDL design is considered a Block (Building Blocks). A VHDL design may be completely described in a single block, or it may be decomposed in several blocks. Each block in VHDL is analogous to an off-the-shelf part and is called an entity. The entity describes the interface to that block and a separate part associated with the entity describes how that block operates. The interface description is like a pin description in a data book, specifying the inputs and outputs to the block.

The following is an example of an entity declaration in VHDL.

```
entity latch is
    port (s,r: in bit;
          q,nq: out bit);
end latch;
```

The first line indicates a definition of a new entity, whose name is latch. The last line marks the end of the definition. The lines in between, called the port clause, describe the interface to the design. The port clause contains a list of interface declarations. Each interface declaration defines one or more signals that are inputs or outputs to the design. Each interface declaration contains a list of names, a mode, and a type.

The following is an example of an architecture declaration for the latch entity.

```
Architecture dataflow of latch is
    signal q0 : bit := '0';
    signal nq0 : bit := '1';
begin
    q0<=r nor nq0;
```



```
    nq0<=s nor q0;
    nq<=nq0;
    q<=q0;
end dataflow;
```

The first line of the declaration indicates that this is the definition of a new architecture called dataflow and it belongs to the entity named latch. So this architecture describes the operation of the latch entity. The schematic for the latch might be

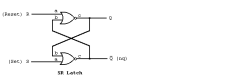

Fig 1.2: Schematic SR Latch

We can specify the same connections that occur in the schematic using VHDL with the following architecture declaration:

```
Architecture structure of latch is
    component nor_gate
    port (a,b: in bit; c: out bit);
    end component;
    begin
      n1: nor_gate
       port map (r,nq,q);
      n2: nor_gate
     port map (s,q,nq);
     end structure;
```

The lines between the first and the keyword begin are a component declaration. A list of components and there connections in any language is sometimes called a netlist. The structural description of a design in VHDL is one of many means of specifying netlists.

1.2.2 DATA FLOW DESCRIPTIONS

In the data flow approach, circuits are described by indicating how the inputs and outputs of built-in primitive components are connected together. Suppose we were to describe the following SR latch using VHDL as in the following schematic.

```
entity latch is
      port (s,r : in bit;
      q,nq : out bit);
end latch;
architecture dataflow of latch is
      begin
      q<=r nor nq;
      nq<=s nor q;
end dataflow;
```

The signal assignment operator in VHDL specifies a relationship between signals, not a transfer of data as in programming languages. The architecture part describes the internal operation of the design. The scheme used to model a VHDL design is called discrete event time simulation. In this the values of signals are only updated when certain events occur and events occur at discrete instances of time. The above mentioned SR latch works with this type of simulation.

1.2.3. BEHAVIORAL DESCRIPTIONS

The behavioral approach to modeling hardware components is different from the other two methods in that it does not necessarily in any way reflect how the design is implemented. **Since encryption and decryption are step by step process, it needs behavioral description in VHDL.**

A. Behavioral descriptions are supported with the Process Statements. The process statement can appear in the body of an architecture declaration just as the signal assignment statement does. The process statement can also contain signal assignments in order to specify the outputs of the process.

B. A Variable is used to hold data and also it behaves like you would expect in a software programming language, which is much different than the behavior of a signal. Although variables represent data like the signal, they do not have or cause events and are modified differently. Variables are modified with the variable assignment.

There are several statements that may only be used in the body of a process.These statements are called *Sequential Statements* because they are executed sequentially. The types of statements used here are if, if else, for and loop.

Next are about the *Signals and Processes*. This section is short, but contains important information about the use of signals in the process statement. A signal assignment, if anything, merely schedules an event to occur on a signal and does not have an immediate effect. When a process is resumed, it executes from top to bottom and no events are processed until after the process is complete.

Final Section in the behavioral description discusses about the *Program Output Methods*. In most programming languages there is a mechanism for printing text on the monitor and getting input from the user through the keyboard. It can able to give output certain information during simulation. A standard library that comes with every VHDL language system. In VHDL, common code can be put in a separate file to be used by many designs. This common code is called a library. In order to use the library that provides input and output capabilities you must add the statement use textio.all; immediately before every architecture that uses input and output. The write statement can be used to append constant values and the value of variables and signals of the types bit, bit_vector, time, integer, and real.

1.3. SOFTWARES USED

- Matlab 7 R2008b
- Xilinx ISE 9.2i

1.4. PROJECT GOAL

The goal of the project is to provide a secure cryptosyitem for images. A new algorithm based on chaos and BB equation are proposed for image encryption and decryption. For practical use, VLSI architectures of the proposed algorithms are designed and realized using Xilinx ISE VLSI software for hardware implementation. Finally the proposed algorithm is compared with the existing transformation algorithm in terms of area, speed and power.

CHAPTER 2
CRYPTOSYSTEM

2.1. CRYPTOSYSTEM

A cryptographic system is any computer system that involves cryptography. Such systems include for instance, a system for secure electronic mail which might include methods for digital signatures, cryptographic hash functions, key management techniques, and so on.

Typically, a cryptosystem consists of three algorithms: one for key generation, one for encryption, and one for decryption.

2.2. CRYPTOGRAPHY

Cryptography is the study of techniques for secure communication in the presence of third parties generally, it is related to various aspects in information security such as data confidentiality, data integrity, and authentication. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

2.3. FUNDAMENTALS OF CRYPTOLOGY

The basic idea of encryption is to modify the message in such a way that its content can be reconstructed only by a legal recipient. A discrete-valued cryptosystem can be characterized by

- a set of possible plaintexts, P
- a set of possible ciphertexts, C
- a set of possible cipherkeys, K
- a set of possible encryption and decryption transformations, E and D

For each key, k ∈ K, there exists an encryption function e(k) ∈ E and a corresponding decryption function d(k) ∈ D, such that for each plaintext p ∈ P the condition for unique decoding, d(k, e(k, p)) = p, is satisfied.

The security of a cryptosystem usually relies on the key only. In other words, it is assumed that the opponent knows the structure of the encryption system, has the ciphering algorithm, and has access to the transmission channel to obtain an arbitrary segment of the ciphertext c.

A good cipher should have strong ability to withstand all kinds of cryptanalysis and attacks that try to break the system. To a certain extent, the resistance against attacks is a good measure of the performance of a cryptosystem; thus, it is often used to evaluate cryptosystems.

2.4. MODERN CRYPTOGRAPHY

2.4.1. SYMMETRIC-KEY CRYPTOGRAPHY

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government. Despite its deprecation as an official standard, DES remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access.

2.4.2. PUBLIC-KEY CRYPTOGRAPHY

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. In a public-key encryption system, the *public key* is used for encryption, while the *private* or *secret key* is used for decryption.

2.5. CRYPTANALYSIS

Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

According to the method of the opponent's access to additional information, attacks on a cryptosystem may be classified into four classes:

• **Ciphertext-only attack:** Opponent has access to communication channel and can eavesdrop some segments of the ciphertext, encrypted by a certain key. The task of the opponent is to reveal as much plaintext as possible, and even to be able to deduce the cipher key.
• **Known-plaintext attack:** In addition to the obtained ciphertext segments, the opponent knows also an associated piece of plaintext. The task of the opponent is then to deduce the cipher key.
• **Chosen-plaintext attack:** The opponent not only has access to some segments of the cipher and the plaintext, but also can choose plaintext to encrypt and accordingly gets some corresponding ciphertext that he wants for comparison. This kind of attack is more intensive than the known-plaintext attack.
• **Chosen-ciphertext attack:** The opponent can choose different segments of the ciphertext and accordingly get its corresponding plaintext.

Apart from the aforementioned typical attacks, there is a type of attack named exhaustive key search, which tries all possibilities for the key in the keyspace to completely decrypt the plain message. If the keyspace of a cipher is relatively small, this exhaustive searching works quite well, given the availability of supercomputing power today. It should be emphasized that any encryption algorithm, traditional or chaos-based, should obey basic cryptographical principles in order to be able to resist serious attacks.

2.6. Salient Features of Image Encryption

Unlike text messages, image data have special features such as high redundancy and high correlation among pixels, not to mention that they usually are huge in size, which together make traditional encryption methods difficult to apply and slow to process. Sometimes image applications also have their own requirements like real-time processing, fidelity reservation, image format consistency, and data compression for transmission. Simultaneous fulfillments of these requirements, alongwith high security and high quality demands, have presented great challenges to real-time imaging practice.

One example is the case where one needs to manage both encryption and compression. In doing so, if an image is to be encrypted after its format is converted, say from a TIFF file to a GIF file, encryption has to be implemented before compression. However, a conventional encrypted image has very little compressibility. On the other hand, compression will make a correct and loss-less decipher impossible, particularly when a highly secure image encryption scheme is used. This conflict between the compressibility and the security is very difficult, if not impossible, to completely resolve.

The salient features of image encryption may be summarized as follows:

- High redundancy and large size generally make encrypted image data vulnerable to attacks via cryptanalysis. Due to its size, the opponent can gain enough cipher text samples (even from one picture) for statistical analysis. Meanwhile, since data in images have high redundancy, adjacent pixels likely have similar grayscale values, or image blocks have similar patterns, which usually embed the image with certain patterns that result in secret leakage.

- Image data have strong correlations among adjacent pixels, which makes fast data-shuffling quite difficult. Statistical analysis on large numbers of images shows that averagely adjacent 8 to 16 pixels are correlative in the horizontal, vertical, and also diagonal directions for both natural and computer-graphical images. According to Shannon's information theory, a secure cryptosystem

should fulfill a condition on the information entropy,E(P|C) = E(P) where P stands for plain message and C for ciphered message; that is, the ciphered (i.e., encrypted) image should not provide any information about the plain image. To meet this requirement, therefore, the ciphered image should be presented as randomly as possible. Since a uniformly distributed message source has a maximum uncertainty, an ideal cipher image should have an equilibrium histogram, and any two adjacent pixels should be uncorrelated statistically. This goal is not easy to achieve under only a few rounds of permutation and diffusion.

- Huge size image data also makes real-time encryption difficult. Compared with texts, image data capacity is horrendously large. For example, a common 24-bit true-color image of 512-pixel height and 512-pixel width occupies 512 × 512 × 24/8 = 768 KB in space. Thus, a one-second motion picture will reach up to about 19 MB. Real-time processing constraints are often required for imaging applications, such as video conferencing, image surveillance, and so on. Vast amounts of image data put a great burden on the encoding and decoding processes. Encryption during or after the encoding phase, and decryption during or after the decoding phase, will aggravate the problems. If an encryption algorithm runs very slowly, even with high security, it would have little practical value for real-time imaging applications. That is the reason why current encryption methods such as DES, IDEA, and RSA are not the best candidates for this consideration.

- Image encryption is often to be carried out in combination with data compression. In almost all cases, the data are compressed before they are stored or transmitted due to the huge amount of image data and their very high redundancy. Thus, directly incorporating security requirements in the data compression system is a very attractive approach. The main challenge is how to ensure reasonable security while reducing the computational cost without downgrading the compression performance.

- In image usage, file format conversion is a frequent operation. It is desirable that image encryption not affect such an operation. Thus, directly treating image data as ordinary data for encryption will make file format conversion impossible. In this scenario, content encryption, where only the image data are encrypted, leaving file header and control information unencrypted, is preferable.

- Human vision has high robustness to image degradation and noise. Only encrypting those data bits tied with intelligibility can efficiently accomplish image protection. However, conventional cryptography treats all image data bits equally in importance, and thus requires a considerable amount of computational power to encrypt all of them, which has often proved unnecessary.

- In terms of security, image data are not as sensitive as text information. Security of images is largely determined by the real situation in an application. Usually, the value of the image information is relatively low, except in some specific situations like military and espionage applications or video conferencing in business. A very expensive attack of encrypted median data is generally not worthwhile. In practice, many image applications do not have very strict security requirements. Under certain circumstances, protection of the fidelity of an image object is more important than its secrecy. An example is electronic signatures.

Currently, there does not seem to be any image encryption algorithm that can fulfill all the aforementioned specifications and requirements. Chaos-based image encryption, can provide a class of very promising methods that can partially fulfill many of these requirements and demonstrate superiority over the conventional encryption methods, particularly with a good combination of speed, security, and flexibility.

A somewhat different chaos-based image encryption method was proposed in that makes use of the SCAN language. Through substitution of each pixel based on an additive noise vector and scramble scanning patterns, an image can be encrypted and compressed simultaneously. The idea seems to be quite good, but it was pointed out in [17] that this method is weak against exhaustive key searching and chosen-plaintext attacks. In another image compression and encryption algorithm was proposed based on the lossless quadtree image compression scheme. The quadtree data structure is used to represent the image, and the scanning sequences of image data comprise a private key for encryption. Also numerous attacks on the proposed algorithm were tested and presented, which include key space reduction, histogram attack, known-plaintext attack, and chosen-plaintext attack.

In order to speed up encryption processes so as to make them feasible for real-time applications, most of the existing schemes follow the idea of selective encryption. Actually, according to Shannon's theory, both encryption and compression are processes of redundancy reduction, but their purposes are different. Several partial encryption schemes were provided. It was reported that by a partial encryption, only 13% to 27% of the output from a quadtree compression algorithm is encrypted for a typical image, and less than 2% is encrypted for a 512×512 image compressed by set-partitioning in the hierarchical trees algorithm. There are also several proposed schemes that consider matching the compatibility to current international standards. Since many international standards on videos and images use block-based discrete cosine transform (DCT), including the familiar JPEG, MPEG-1, MPEG-2, H.261, and H.263 formats, the current research has been concentrated on selective encryptions within the framework of DCT. However, with the emergence of MPEG-4 and JPEG-2000, research emphasis may soon be redirected to a combination of encryption and wavelet compression.

To achieve high encryption speed, in the early stage some elementary cryptographic methods using random permutation lists were suggested. Since the operations are simple, the encryption does not require high computational cost. The challenge is how to achieve reasonable security with such simple operations.

2.7. Existing Image Encryption Schemes

Some image encryption methods have been proposed in the current literature. In order to inspire the development of better chaotic ciphers, this review is not only intended for chaos-based methods, but is also meant for understanding image encryption technology in general. Image encryption algorithms, which can be classified with respect to the approach in constructing the scheme, are divided into two groups here: chaos- based methods and non-chaos-based methods. Image encryption also can be divided into full encryption and partial encryption (also called selective encryption) according to the percentage of the data encrypted. Moreover, they can be classified into compression-combined methods and non compression methods.

Some existing proposals of chaoi-based image encryption algorithms are now introduced. In this two kinds of schemes based on higher-dimensional chaotic maps were proposed. By using a discretized chaotic map, pixels in an image are permuted in shuffling after several rounds of operations. Between every two adjacent rounds of permutations, a diffusion process is performed, which can significantly change the distribution of the image histogram that makes statistical attack infeasible. Empirical testing as well as cryptanalysis both demonstrated that the chaotic baker map and cat map are good candidates for this kind of image encryption. The aforementioned schemes are block cipher, and they have some prominent merits, including high security and fast processing. However, their defects are also significant since the encrypted image has very little compressibility and is unable to abide any lossy compression (e.g., JPEG). To alleviate the conflict between compressibility and encryption, several suggestions of combining compression and encryption have been proposed. The so-called MHT scheme was proposed that encrypts image via a manipulation of Huffman coding tables in the image coding system. The MHT scheme chooses several different Huffman tables from a large number of possible candidates, and uses them alternatively to encode the image data. The choice of Huffman tables and the order in which they are used are kept secret as the key. It was advocated that the method requires very little computational overhead and can be applied to MPEG and JPEG/JPEG 2000, but it cannot resist chosen-plaintext attack.

2.8. Chaos-Based Encryption Schemes

A great deal of work on application of chaos to cryptography has been carried out in the last decade. Early works on chaos in cryptography were connected with encrypting messages through modulation of chaotic orbits of continuous-time dynamical systems. These methods are strongly related to the concept of synchronization of two chaotic systems and to chaos control. Several different ways have been proposed to achieve synchronization of chaotic systems, thereby transmitting information on a chaotic carrier signal.

The following technical problems were listed in :

• It is difficult to determine the synchronization time; therefore, the message during the transient period will be lost, sometimes cause fairly long transient times.

• Noise throughout the transmission significantly affects the intended synchronization. This means the synchronization noise intensity should be small compared to the signal level, or the desired synchronization will not be achieved.

• Technically, it is difficult to implement two well-matched analog chaotic systems, which are required in synchronization, and if this is not required (i.e., with certain robustness) then the opponent can also easily achieve the same synchronization for attack. In contrast to synchronization-based techniques, a direct application of a chaotic transformation to a plaintext, or applying a chaotic signal in the design of an encryption algorithm, seems to be a more promising approach.

The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

2.8.1. Basic Features of Chaos

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behaviors. Since its discovery by Edward N. Lorenz in 1963, chaos theory has become a branch of scientific studies today. Since discrete chaotic dynamic systems (i.e., maps) are used in cryptography, this notion is briefly introduced.

2.8.2. Definition of discrete chaos

There are several definitions of chaos, which are similar but are actually not equivalent.

For simplicity, one-dimensional maps are discussed. Consider a discrete dynamical system in the general form of

$$x_{k+1} = f(x_k), \quad f : I \longrightarrow I, \; x_0 \in I \qquad \text{..... (1)}$$

where f is a continuous map on the interval I = [0, 1]. This system is said to be chaotic if the following conditions are satisfied:

1. Sensitive to initial conditions:
$$\exists \delta > 0 \; \forall x_0 \in I, \varepsilon > 0 \; \exists n \in \mathbb{N}, y_0 \in I : |x_0 - y_0| < \varepsilon \;\Rightarrow\; |f^n(x_0) - f^n(y_0)| > \delta. \qquad \text{..... (2)}$$
2. Topological transitivity:
$$\forall I_1, I_2 \subset I \; \exists n_0 \in I_1, n \in \mathbb{N} : f^n(x_0) \in I_2. \qquad \text{.....(3)}$$
3. Density of periodic points in I:
Let,
$$P = \{p \in I | \exists n \in \mathbb{N} : f^n(p) = p\}$$
be the set of periodic points of f. Then P is dense in I: P = I.

2.8.3. Relationships Between Chaos and Cryptography

The similarities and differences between the two subjects can be shown in Table 1. Chaotic maps and cryptographic algorithms have some similar properties: both are sensitive to tiny changes in initial conditions and parameters; both have random like behaviors; and cryptographic algorithms shuffle and diffuse data by rounds of encryption, while chaotic maps spread a small region of data over the entire phase space via iterations. The only difference in this regard is that encryption operations are defined on finite sets of integers while chaos is defined on real numbers.

2.8.4. Chaos for Cryptography

It is natural to apply the discrete chaos theory to cryptography for the following reasons :

The property of sensitive dependence of orbits on initial conditions makes the nature of encryption very complicated. Suppose that one has the following chaos-based encryption scheme:

Chaotic systems	Cryptographic algorithms
Phase space: set of real numbers	Phase space: finite set of integers
Iterations	Rounds
Parameters	Key
Sensitivity to initial conditions and Parameters	Diffusion

Table 1. Similarities and differences between chaos and cryptography

CHAPTER 3
LITERATURE SURVEY

Aloka Sinha and Kehar Singh [18] have proposed a new technique to encrypt an image for secure image transmission.The digital signature of the original image is added to the encoded version of the original image. Image encoding is done by using an appropriate error control code, such as a Bose-Chaudhuri Hocheuenghem (BCH) code. At the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image.

S.S. Maniccam and N.G. Bourbakis [19] have presented a new methodology which performs both lossless compression and encryption of binary and gray-scale images. The compression and encryption schemes are based on SCAN patterns generated by the SCAN methodology. The SCAN is a formal language-based two-dimensional spatial-accessing methodology which can efficiently specify and generate a wide range of scanning paths or space filling curves.

Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen [20] use one of the popular image compression techniques, vector quantization to design an efficient cryptosystem for images. The scheme is based on vector quantization (VQ), cryptography, and other number theoreems.In VQ, the images are first decomposed into vectors and then sequentially encoded vector by vector. Then traditional cryptosystems from commercial applications can be used.

Jiun-In Guo and Jui-Cheng Yen [21] have presented an efficient mirror-like image encryption algorithm. Based on a binary sequence generated from a chaotic system, an image is scrambled according to the algorithm. This algorithm consists of 7 steps.Step-1 determines a 1-D chaotic system and its initial point x(0) and sets k = 0.

Step-2 generates the chaotic sequence from the chaotic system.

Step-3 generates binary sequence from chaotic systems.

Steps-4,5,6, and 7 rearrange image pixels using swap function according to the binary sequence.

Jui-Cheng Yen and Jiun-In Guo [22] have proposed a new image encryption scheme based on a chaotic system. In their method, an unpredictable chaotic sequence is generated. It is used to create a binary sequence again. According to the binary sequence, an image's pixels are rearranged. This algorithm has four steps.

Step-1 determines a chaotic system and its initial point x(0), row size M and column size N of the image f, iteration number no, and constants α, β and μ used to determine the rotation number.

Step-2 generates the chaotic sequence from the chaotic system.

Step-3 generates the binary sequence.

Step-4 includes special functions to rearrange image pixels. These functions are;

$ROLR_i^{\pm P} : f \rightarrow f'$ is defined to rotate each pixel in the ith row in f, $0 \le i \le M \le 1$, in the left direction p pixels if 1 equals 0 or in the right direction p pixels if1 equals 1.

$ROUD_j^{\pm P} : f \rightarrow f'$ is defined to rotate each pixel in the ith column in f, $0 \le j \le N \le 1$, in the up direction p pixels if1 equals 0 or in the down direction p pixels if1 equals 1 .

$ROUR_i^{k,P} : f \rightarrow f'$ is defined to rotate each pixel at position (x,y) in the image f such that x+y=k, $0 \le k \le M + N - 2$, , in the upper-right direction p pixels if1 is equal to 1 or in the lower-left direction p pixels if1 is equal to 0.

Shuqun Zhang and Mohammad A. Karim [23] have proposed a new method to encrypt color images using existing optical encryption systems for gray-scale images. The color images are converted to their indexed image formats before they are encoded. In the encoding subsystem, image is encoded to stationary white noise with two random phase masks, one in the input plane and the other in the Fourier plane. At the decryption end, the color images are recovered by converting the decrypted indexed images back to their RGB (Red-Green-Blue) formats. The proposed single-channel color image encryption method is more compact and robust than the multi channels methods.

Visual cryptography uses the characteristics of human vision to decrypt encrypted images. It needs neither cryptography knowledge nor complex computation. For security concerns, it also ensures that hackers cannot perceive any clues about a secret image from individual cover images. Young-Chang Hou [24] have proposed three methods for visual cryptography.

2.9. CHAOS-BASED IMAGE ENCRYPTION

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behavions. The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

Many chaos-based encryption schemes have been proposed, and some of them have been extended from text encryption to image encryption. A direct extension of a chaos-based text encryption scheme to also work for images is possible, but this simple modification may not provide an efficient solution to these image encryption problems. Image encryption has its own specifications such as encryption speed, compatibility to image format and compression standards, and real-time implementation, therefore it requires a special design of the encryption algorithm.

An encryption method called chaotic key-based algorithm (CKBA) was proposed. The algorithm first generates a time series based on a chaotic map, and then uses it to create a binary sequence as a key. According to the binary sequence generated, image pixels are rearranged and XOR or XNOR operated with the selected key. This method is very simple but has defects in Security. This method is very weak to the chosen/known-plaintext attack with only one plain image. Moreover, its security to brute-force attack is also questionable.

2.10. BB EQUATION OVERVIEW

The Brahmagupta–Bhaskara (BB) equation is a Quadratic Diophantine equation of the form $NX^2 + k = Y^2$,Where k is an integer and N is a positive integer such that $\sqrt{N}$

is irrational. A particular case of the BB equation with k = 1 is also known as Pell equation. This equation in the Galois Field GF(p), has some useful properties. Application of these properties in two different fields of cryptography, namely, digital encryption and user authentication. For those applications, where software computation of the roots of the BB equation is unacceptable for being too slow, a hardware architecture for using the BB equation in GF(p) is given that is useful for implementation in VLSI form.

2.10.1. APPLICATION OF BB EQUATION IN CRYPTOGRAPHY

The BB-equation in Galois Field GF(p), can be written as,

$$(nx^2)_p + 1 = (y^2)_p \qquad (1)$$

where p is an odd prime.The alternative representation of BB equation in GF (p) can be rewritten as,

$$(nq_x+1)_p=(q_y)_p \qquad (2)$$

where $q_x = (x^2)_p$, $q_y = (y^2)_p$ and the subscript p stands for modulo operation by p on the argument values of the expressions. The application of the BB equation for encryption depends on the following two properties.

1. Given n and p, with p>n>0, it is always possible to obtain q_x and q_y corresponding to the roots of the BB equation $(nx^2 + 1)p = (y^2)p$.
2. Given q_x and q_y corresponding to any root of the BB equation $(nx^2 + 1)p = (y^2)p$, it is always possible to compute uniquely, the corresponding value of n , only with the knowledge of p.

The encryption process based on the BB equation is as follows.

1. n corresponds to the cleartext or plaintext in a block that is being encrypted.
2. p corresponds to the primary secret key used in the encryption of the plaintext in a block.
3. The ciphertext corresponding to n is the pair (q_x,q_y) of the corresponding BB equation.

Gray-level visual cryptography method first transforms the gray-level image into a halftone image and then generates two transparencies of visual cryptography. Obviously, we indeed cannot detect any information about the secret image from the two sharing transparencies individually, but when stacking them together, the result clearly shows the secret image.Method 1 uses four halftone images, cyan, magenta, yellow and black, to share the secret image. The codes of the four sharing images are fully disordered, and we cannot perceive any clue of the original secret image from any single sharing image. Method 2 reduces the inconvenience of Method 1 and requires only two sharing images to encrypt a secret image. However, after stacking the sharing images generated by Method 2, the range of color contrast will be 25% of that of the original image. Method 3 loses less image contrast, which is better than Method 2.

In cryptography, ICE (Information Concealment Engine) is a block cipher published by Kwan in 1997. The algorithm is similar in structure to DES, but with the addition of a key-dependent bit permutation in the round function. The key-dependent bit permutation is implemented efficiently in software. The ICE algorithm is not subject to patents, and the source code has been placed into the public domain.The ICE algorithm was designed for use in software applications. Those applications however are slow due to the use of modular arithmetic [25]. So the need for faster implementations is great. That can be achieved through hardware implementations. Considering the fact that hardware implementations are generally faster and more Ease of Use reliable than software implementations the outcome of a hardware design is even more interesting.

The system operates for the both encryption and decryption processes and has been optimized for low hardware resources and for high–speed performance. The proposed architecture has very encouraging performance result in terms of speed and throughput. This makes the design very useful in current applications that use DES as the base of a cryptographic protocol [25].

ICE is a Feistel network with a block size of 64 bits. The standard ICE algorithm takes a 64-bit key and has 16 rounds. A fast variant, Thin-ICE, uses only 8 rounds. An open-ended variant ICE-n, uses 16n rounds with 64n bit key. They described an attack on Thin-ICE which recovers the secret key using 223 chosen plaintexts with a 25% success probability. If 227 chosen plaintexts are used, the probability can be improved to 95%. For the standard version of ICE, an attack on 15 out of 16 rounds was found, requiring 256 work and at most 256 chosen plaintexts.

ICE is a standard Feistel block cipher [25] with a structure similar to DES. It takes a 64-bit plaintext, which is split into two 32-bit halves. In each round of the algorithm the right half and a 60-bit subkeys are fed into the function F. The output of F is XORed with the left half, and then the halves are swapped. This is the Transformation Round of the ICE algorithm. This process is repeated for 16 rounds. However the final swap is left out. The decryption process is the same, except that the sub keys are used in reverse order.

The advantages of Feistel structure are one-to- one mapping between plaintext and cipher text, which is necessary for a cipher to be decryptable. Secondly, Feistel ciphers have been publicly cryptanalysed for more than two decades, and no systematic weakness has been uncovered. And finally, Feistel ciphers are reasonably fast and simple to implement in software. Speed and simplicity were two important design aims for ICE.

CHAPTER 4
IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM

In most of the natural images, the values of the neighboring pixels are strongly correlated (i.e. the value of any given pixel can be reasonably predicted from the values of its neighbors) . In order to dissipate the high correlation among pixels and increase the entropy value, we propose a transformation algorithm that divides the image into blocks and then shuffles their positions before it can be encrypted with the secrete key. By using the correlation and entropy as a measure of security, this process results in a lower correlation, a higher entropy value and thus improving the security level of the encrypted images. The variable secret key of the transformation process is the main key is used to increase the entropy.

4.1. DESCRIPTION OF THE TRANSFORMATION ALGORITHM

The transformation technique works as follows:

The original image is divided into a random number of blocks that are then shuffled within the image. The generated (or transformed) image is then fed to the encryption algorithm. The main idea is that an image can be viewed as an arrangement of blocks. The intelligible information present in an image is due to the correlation among the image elements in a given arrangement. This perceivable information can be reduced by decreasing the correlation among the image elements using certain transformation techniques.

The secret key of this approach is used to determine the seed. The seed plays a main role in building the transformation table, which is then used to generate the transformed image with different random number of block sizes. The transformation process refers to the operation of dividing and replacing an arrangement of the original image.The image can be decomposed into blocks; each one contains a specific number of

pixels. The blocks are transformed into new locations. For better transformation the block size should be small, because fewer pixels keep their neighbors. In this case, the correlation will be decreased and thus it becomes difficult to predict the value of any given pixel from the values of its neighbors. At the receiver side, the original image can be obtained by the inverse transformation of the blocks. A general block diagram of the transformation method is shown in Fig.4.1.

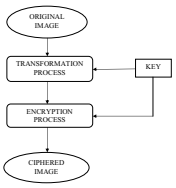

Fig.4.1. General block diagram of the transformation algorithm

Case 1: $q_{ex}(x,y) = \text{mod}((q_x (x,y) + \text{key2}), 2n\text{-}1)$
$q_{ey}(x,y) = q_{ex}(x,y)$ XOR key2
$q_{ex}(x,y) = \text{mod}((q(x,y) + \text{key2}), 2n\text{-}1)$
$q_{ex}(x,y) = q_{ex}(x, y)$ XOR key2

Case 0: $q_{ex}(x, y) = \text{mod}((q_x (x, y) +\text{key2}), 2n\text{-}1)$
$q_{ex}(x,y) = q_{ex}(x,y)$ XNOR key2
$q_{ey}(x, y) = \text{mod}((q(x, y) +\text{key2}), 2n\text{-}1)$
$q_{ey}(x, y) = q_{ex}(X,y)$ XNOR key2

j = j + 2;
End;
End;

Step 4: The result $q_{ex}(x,y)$ $q_{ey}(x,y)$ is obtained and stop the algorithm.

The proposed VLSI architectures have two key modules, one for the generation of chaotic bits (CB), and the other for encryption or decryption. The architecture of the chaotic bits (CB) generator is 32. The concept of parallel processing is adopted so that the encryption or decryption of 16 data values can be performed at the same time. Fig.5.1 shows the hardware architecture of the encryption unit (EU). This architecture consists of one 32 bit parallel-in parallel-out register, and 16 encryption processing elements (EPEs). The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input.

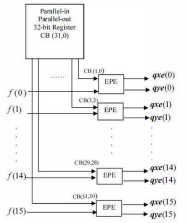

Fig.5.1. Architecture of the encryption unit

The cascade architecture of the encryption processing element (EPE) is shown in Fig.5.2. The architecture of EPE 1 is shown in Fig.5.3. It consists of three multipliers, one adder, two Mod operations and one comparator. The architecture of EPE2 is shown in Fig.5.4. It consists of four data multiplexers, two adders, two XOR gates, two MOD operations, and two inverters, four parallel-to-serial converters, and two serial-to parallel converters.

4.2. ALGORITHM FOR CREATING TRANSFORMATION TABLE

1: Load Image
2: Input key
3: Randomize ()
3.1: HorizontalNoBlocks = RandomNum between
(LowerHorizontalNoBlocks and ImageWidth)
3.2: VerticalNoBlocks = RandomNum between
(LowerVerticalNoBlocks and ImageHeight)
4: NoBlocks = HorizontalNoBlocks * VerticalNoBlocks
END CREATE_TRANSFORMATION_TABLE
Input: Original Image
Output: Transformation table

4.3. ALGORITHM FOR PERFORMING TRANSFORMATION

1: For I = 0 to NoBlocks -1
1.1: Get the new location of block from the transformation table
1.2: Do XOR operation between a key and a value from the new location
1.3: Set block I in its new location
END PERFORM_TRANSFORMATION
Input: Original Image, a string key and Transformation table
Output: Transformed Image.

- For ex, if the input pixel value is FF, then from the transformation table the value corresponding to the FF can be replaced by the stored value in the table.
- After transforming, the transformed image can be encrypted by doing XOR operation with the key.

CHAPTER 5
ENCRYPTION & DECRYPTION ALGORITHM

5.1. Proposed Algorithm For Encryption

The proposed encryption algorithm is as follows.

Step 1 : Choose P, key 1 and key 2 and set j=0.

Step 2 : Choose the initial point x (0) and generate the chaotic sequence x (0), x (1), x(2), ... , x (MN/16-1) using eq.(1) and then create b (0), b (1), b (2), ... , b (2MN-1) from x(0), x (1), x (2),....., x(MN/16 - 1) by the generating scheme such that b(32i+0), b(32i+ 1). . . b(32i+ 29) b(32i+ 30) b(32i+ 31) . . . is the binary representation of x (i) for i = 0, 1, 2, ... (MN/16-1).

Step 3: For x=0 to M-1 ;
For y = 0 to N- 1;

obtain $q_x(x,y)$, $q_y(x,y)$ for chosen p and given f(x, y) from the solution of the BB equation,

Switch (2xb(j) + b (j+1))

Case 3: q_{ex} (x,y) = mod ((q_x (x,y) + key1), 2n-1)
q_{ex} (x,y) = q_{ex} (x,y) XOR key1
q_{ey}(x,y) = mod ((q_x(x,y) + key1), 2n-1)
q_{ey} (x, y) = q_{ex} (x,y)XOR key1

Case 2: q_{ex}(x,y) = mod((q_x (x,y) +key1), 2n-1)
q_{ex}(x,y) = q_{ex}(X,y) XNOR key1
q_{ey}(x,y) = mod ((q_x(x,y) +key1), 2n-1)
q_{ey}(x,y) = q_{ex} (x,y) XNOR key1

5.1.1. ARCHITECTURE OF ENCRYPTION PROCESSING UNIT

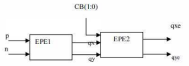

Fig.5.2. Cascade architecture of EPE

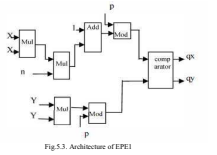

Fig.5.3. Architecture of EPE1

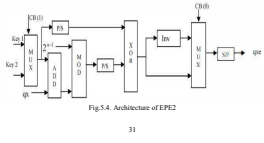

Fig.5.4. Architecture of EPE2

5.2. Proposed Algorithm For Decryption

Steps 1 and 2 are the same as in the above encryption algorithm. Steps 3 and 4 for the decryption are as follows.

Step 3: For x=0 to M-1
For y = 0 to N-1

Switch (2*b(j)) + b(j+1))

Case 3 : q_x (x,y) = q_{ex} (x,y)XOR key1
q_x (x,y) = mod((q_x (x,y) - key1),2^{n-1})
q_y (x,y) = q_{ey} (x,y)XOR key1
q_y(x,y) = mod((q_y(x,y)-key1),2^{n-1})
f(x,y) = (q_x (i)) $^{-1}$ (q_y(i)-1)mod(p).

Case 2 : q_x (x,y) = q_{ex} (x,y) XNOR key1
q_x (x,y) = mod((q_x (x,y) -key1),2^{n-1})
q_y(x,y)= q_{ex}(x ,y) XNOR key1
q_y(x,y)= mod((q_y(x,y)-key1),2^{n-1})
f(x,y) = (q_x(i)) $^{-1}$ (q_y(i)-1)mod(p).

Case 1 : q_x (x,y) = q_{ex} (x,y)XOR key2
q_x (x,y) = mod((q_x (x,y) - key2),2^{n-1})
q_y(x,y)= q_{ex} (x,y) XOR key2
q_y(x,y)= mod((q_y (x,y) - key2),2^{n-1})
f(x, y) = (q_x (i)) $^{-1}$ (q_y (i) - 1) mod(p).

Case 0: q_x (x,y) = q_{ex} (x,y)XNOR key2
q_x (x,y) = mod((q_x (x,y) - key2),2^{n-1})
q_y(x,y)= q_{ex} (x,y) XNOR key2

$q_y(x,y)$= mod(q_y (x,y) - key2),2^{n-1})

f(x, y) = (q_y (x,y))^2 (q_y(x,y) - 1) mod(p).

j= j+ 2;
End;
End;

Step 4: The result f is obtained and stop the algorithm.

The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input. The cascade architecture of the decryption processing element (DPE) is shown in Fig.5.5

The architecture of DPE 1 is shown in Fig.5.6. The architecture of DPE2 is shown in Fig.5.7. It consists of one subtractor, one Modulo inverse operation, one multiplier and one Mod operation.

5.2.1. ARCHITECTURE OF DECRYPTION PROCESSING UNIT

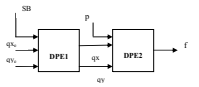

Fig.5.5: Cascade architecture of DPE

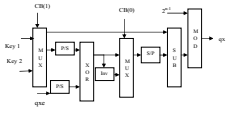

Fig.5.6: Architecture of DPE1

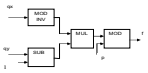

Fig.5.7: Architecture of DPE2

CHAPTER 6

PROPOSED CRYPTOSYSTEM

It describes a cryptosystem based on the use of combined chaos and BB equation in addition to the random sequence generator for providing high security. The block diagram of the proposed cryptosystem for encryption and decryption is shown in fig.6.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XOR-ed or XNOR-ed bit-by-bit to one of the predetermined keys. Then the encrypted image again processed by pseudorandom sequence generator to provide high encryption. The key is very sensitive to initial conditions.

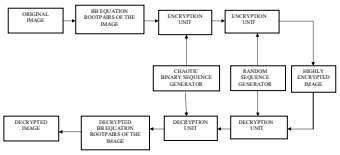

Fig.6.1: Proposed Secure Image Cryptosystem

6.1. GENERATION OF PSEUDO RANDOM SEQUENCE GENERATOR

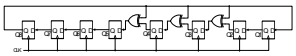

Fig.6.2: 8 bit LFSR structure

The pseudo random sequences are generated by Linear Feedback Shift Register. LFSR is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. And also LFSR is extremely sensitive to initial conditions hence finding the seed is very difficult in this case.

The Fig.8.2 shows an 8-bit LFSR structure used to generate a polynomial of $x^8 + x^6 + x^5 + x^4 + 1$ for 8 – bits. Hence it can generate 256 combinations for a bit single input.

6.1.1. Uses in cryptography

LFSRs have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very uniformly distributed output streams. However, an LFSR is a linear system, leading to fairly easy cryptanalysis. For example, given a stretch of known plaintext and corresponding ciphertext, an attacker can intercept and recover a stretch of LFSR output stream used in the system described, and from that stretch of the output stream can construct an LFSR of minimal size that simulates the intended receiver by using the Berlekamp-Massey algorithm. This LFSR can then be fed the intercepted stretch of output stream to recover the remaining plaintext.

Xilinx, Inc. is an American technology company, which designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing The Virtex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores. Xilinx has offered two main FPGA families: the high-performance Virtex series and the high-volume Spartan series. With the introduction of its 28 nm FPGAs in June 2010, Xilinx replaced the high-volume Spartan family with a Kintex family and the low-cost Artix family. The Spartan series targets applications with a low-power footprint, extreme cost sensitivity and high-volume; e.g. displays, set-top boxes, wireless routers and other applications.

The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using Chip Scope Pro tools, and creation of the bit files that are used to configure the chip.

Xilinx is a synthesis tool which converts Schematic/HDL design entry into functionally equivalent logic gates on Xilinx FPGA, with optimized speed & area. So, after specifying behavioral description for HDL, the designer merely has to select the library and specify optimization criteria; and Xilinx synthesis tool determines the net list to meet the specification; which is then converted into bit-file to be loaded onto FPGA-PROM. Also, Xilinx tool generates post-process simulation model after every implementation step, which is used to functionally verify generated net list after processes, like map, place & route.

The synthesis and the simulation results of the proposed and the existing cryptosystems are shown below.

CHAPTER 7

RESULTS & DISCUSSION

The simulation of this project has been done using MODELSIM 7 R2008b and XILINX ISE 9.11.

Modelsim is a simulation tool for programming {VLSI} {ASIC}s, {FPGA}s, {CPLD}s, and {SoC}s. Modelsim provides a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support is provided for multiple languages including Verilog, SystemVerilog, VHDL and SystemC. The Modelsim conceptual overview is shown below.

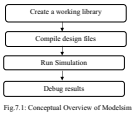

Fig.7.1: Conceptual Overview of Modelsim

In ModelSim, all designs, be they VHDL, Verilog, or some combination thereof, are compiled into a library. We can start a new simulation in ModelSim by creating a working library called "work". "Work" is the library name used by the compiler as the default destination for compiled design units. After creating the working library, we compile our design units into it.

The ModelSim library format is compatible across all supported platforms. We can simulate our design on any platform without having to recompile your design. With the design compiled, invoke the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the simulation time is set to zero, and enter a run command to begin simulation. If the results are not as expected, use ModelSim's robust debugging environment to track down the cause of the problem.

7.1. SIMULATION RESULTS

7.1.1. Simulation Result of Existing IETA Algorithm

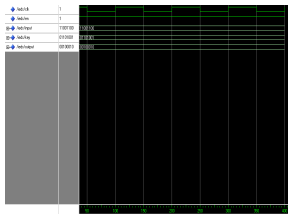

Fig.7.2: Simulation Result of IETA Algorithm

In the above figure input and key are the inputs and output can be produced according to value stored in the transformation table and the value of 8 bit key.

7.1.2. Simulation Result of Proposed Encryption Algorithm

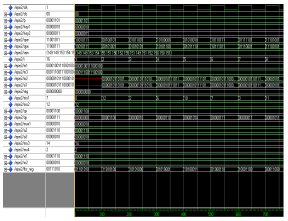

Fig.7.3: Simulation Result of Proposed Encryption Algorithm

In the above figure cb is the combinational register used to select the Encryption Processing Element in the encryption unit, n is the plain text and p is the secret key, key 1 and key 2 are the additional keys used in the encryption process. q_{xe} and q_{ye} are the encrypted output values. All the keys used here are 8 bits. LFSR is the feedback shift register used to provide more encryption.

7.1.3. Simulation Result of Proposed Decryption Algorithm

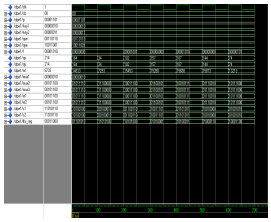

Figure.7.4. Simulation Result of Proposed Decryption Algorithm

In the above figure Encrypted image pixel values can be given as input to the Decryption Unit(DU). Here qxe, qye, primary secret key p, key 1 key 2 are the inputs to the DU. Output pixel values are exactly same as that of input pixel values after decryption process.

7.2.2. AREA REPORT

7.2.2.1. Existing IETA Algorithm

Design Summary

Number of errors: 0
Number of warnings: 8

Logic Utilization:
Number of 4 input LUTs: 16 out of 13,824 2%

Logic Distribution:
Number of Slices containing only related logic: 0 out of 0 0%
Number of Slices containing unrelated logic: 0 out of 0 0%

Number of bonded IOBs: 17 out of 510 3%
Number of Block RAMs: 1 out of 72 1%
Number of GCLKs: 1 out of 4 25%
Number of GCLKIOBs: 1 out of 4 25%

Total equivalent gate count for design: 16,384
Additional JTAG gate count for IOBs: 864

7.2.2.2. Proposed Chaos & BB Combined Algorithm

Area report:
Design Summary

Number of errors: 0
Number of warnings: 2

Logic Utilization:

Number of 4 input LUTs: 8 out of 13,824 1%

Logic Distribution:

Number of occupied Slices: 4 out of 6,912 1%
Number of Slices containing only related logic: 4 out of 4 100%
Number of Slices containing unrelated logic: 0 out of 4 0%

Total Number of 4 input LUTs: 8 out of 13,824 1%
Number of bonded IOBs: 34 out of 510 6%

Total equivalent gate count for design: 48
Additional JTAG gate count for IOBs: 1,632

7.2. SYNTHESIS RESULTS

The proposed design has been implemented on Xilinx Spartan 2E device and simulated by Xilinx 9.2i design tool. Therefore the device utilization summary and the timing summary can be obtained from the *Synthesis Report* as shown below.

7.2.1. Power Report

7.2.1.1. Image Encryption using Transformation Algorithm

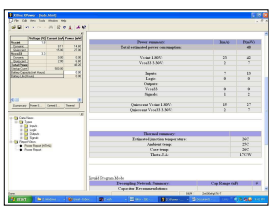

Fig.7.5: Power Report of IETA Algorithm

7.2.1.2. Proposed Cryptosystem based on Chaos & BB Equation

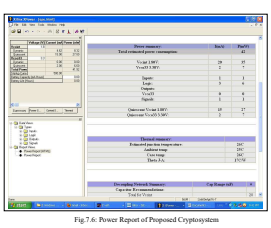

Fig.7.6: Power Report of Proposed Cryptosystem

7.2.3. Delay Report

7.2.3.1. Image Encryption Using Transformation Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 7.548ns (Maximum Frequency: 132.481MHz)
Minimum input arrival time before clock: 8.535ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **2.319ns**

7.2.3.2. Proposed Chaos & BB Combined Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 2.956ns (Maximum Frequency: 338.295MHz)
Minimum input arrival time before clock: 2.948ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **3.268ns**

7.3. COMPARISON

SUMMARY	IETA BASED CRYPTOSYSTEM	PROPOSED CHAOS & BB BASED CRYPTOSYSTEM	% OF RESULTS
POWER	48Mw	42mW	**12.5**
SPEED	2.319ns	3.268ns	**40**
AREA Total Gate Count	16384	48	**99.7**

Table 2.Comparison between IETA based Cryptosystem and proposed Cryptosystem

CHAPTER 8
RESULTS

8.1. IMAGE RESULTS OF EXISTING IETA

INPUT IMAGE

Fig.8.1. lena image

ENCRYPTED IMAGE

Fig.8.2.Encrypted Image

OUTPUT IMAGE

Fig.8.3.Decrypted Image

8.2. IMAGE RESULTS OF PROPOSED CHAOS & BB EQUATION

8.2.1. Results for Grayscale image

INPUT IMAGE

Fig.8.4. lena image

ENCRYPTED IMAGE

Fig.8.5.Encrypted Image

OUTPUT IMAGE

Fig.8.6.Decrypted Image

8.2.2. Results for Color image

INPUT IMAGE

Fig.8.7. lena image

ENCRYPTED IMAGE

Fig.8.8.Encrypted Image

OUTPUT IMAGE

Fig.8.9.Decrypted Image

8.2.3. Results for fingerprint image

INPUT IMAGE

Fig.8.10. fp image

ENCRYPTED IMAGE

Fig.8.11.Encrypted Image

OUTPUT IMAGE

Fig.8.12..Decrypted Image

CHAPTER 9
CONCLUSION & FUTURE WORK

In this project, we proposed a robust and efficient cryptosystem to transmit a digital image in a secure way. A secure cryptosystem based on Chaos & BB equation with pseudorandom sequence is proposed for image encryption & decryption. The new quadratic equation based encryption & decryption algorithm alters the positions of the pixels by XOR and Mod operations without changing its correlation with neighbouring pixels. This method is highly sensitive to initial conditions. The VLSI architecture of the proposed cryptosystem is designed and realized using Xilinx ISE VLSI software for an image. The proposed algorithm is highly promising for high security in real time applications. The proposed algorithm also compared with existing IETA algorithm.

It should be proved that this system is secure against all kinds of attacks. Cryptanalysis of the proposed cryptosystem specifically for cipher text attacks and known plain text attacks is to be provided.

REFERENCES :

1. K.Deergha Rao and Ch. Gangadhar, "VLSI realization of a secure cryptosystem for image encryption and decryption" in Proceedings of IEEE Int. Conf. on Communication and signal processing, 2011
2. K.Deergha Rao, K.Pravveen Kumar,and P. V.Muralikrishna,"A New and Secure Cryptosystem for Image Encryption and decryption", appear in IETE Journal of Research, March-Apr. 2011
3. Mohammad Ali Bani Younes and Aman Jantan, "Image Encryption Using Block-Based Transformation Algorithm" Proc. IAENG International Journal of Computer Science, Feb 2008, 35:1, IJCS_35_1_03
4. G.Alvarez L.H.Encinas, and M.Masque, "Known-Plaintext Attack to Two Cryptosystems Based on the BB Equation", IEEE Transactions on Circuits and Systems II: Express Briefs Volume 55, Issue 5, May 2008
5. R.Rhouma, S.Meherzi, S.Belghith, "CML-based colour image encryption" in International Journal of Chaos, Solitons and Fractales 2007
6. Alvarz, G. and Shujun L. "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystem" in Int. Journal of Bifurcation and Chaos, 2006
7. N.Rama Murthy, M.N.S.Swamy, "Cryptographic Applications of Brahmagupta-Bhaskara Equation", IEEE Transactions on circuits -I, 2006
8. G.Zhi-Hong, H.Fangjun, and G.Wenjie,"Chaos - based image encryption algorithm,"Department of Electrical and computer Engineering,University of Waterloo, ON N2L 3G1, Canada.Published by: Elsevier, 2005, pp. 153-157.

9. I.Ozturk, I.Sogukpinar, "Analysis and comparison of image encryption algorithm,"Journal of transactions on engineering, computingand technology December, vol. 3, 2004, p.38.
10. B.Furht and D.Socek, "Multimedia security encryption techniques" IEC Int. Engg. Consortium, Chicago, IL pages 335-349, 2004
11. Sinha, K. Singh, "A technique for image encryption using digi tal signature," Source:Optics\Communications,vol.218,no.,2003
12. G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalyzing a discrete-time chaos synchronization secure communication system,"Chaos, Solitons &fractals, 2003.pp. 689-694.
13. S.1.Li and X.Zheng , "On the security of an image encryption method," Proc. IEEE International Conference on Image Processing (ICIP 2002), vo1.2, pp. 925-928, 2002.
14. S.J.Li and X. Zheng, "Cryptanalysis of a Chaotic Image Encryption Method", IEEE International Symposium on Circuits and Systems (ISCAS 2002)
15. M.I.Sobhy, and A.R.Shehata, "Methods of attacking chaotic encryption and countermeasures," Proc. IEEE Inernational Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)
16. H.Cheng and X.Li, "Partial encryption of compressed images and videos" in IEEE Transactions on Signal Processing, volume 48, pages 2439-2451, 2000
17. Dachselt F, Schwarz W (2001) Chaos and cryptography. IEEE Trans Circuits and Systems-I 48(12):1498 – 1509

18. Aloha Sinha, Kehar Singh, "A technique for image encryption using digital signature", Optics Communications, ARTICLE IN PRESS, 2003,1-6,
19. S.S.Maniccam, N.G. Bourbakis, "Lossless image compression and encryption using SCAN", Pattern Recognition 34 (2001), 1229-1245
20. Chin-Chen Chang, Min-Shian Hwang, Chen,"A new encryption algorithm for image cryptosystems", The Journal of Systems and Software 58 (2001), 83-91
21. Jiun-In Guo, Jui-Cheng Yen, "A new mirror-like image encryption algorithm and its VLSI architecture", Department of Electronics Engineering National Lien-Ho College of Technology and Commerce,Miaoli,(2001) Taiwan, Republic of China
22. Jui-Cheng Yen, Jiun-In Guo, "A new chaotic image encryption algorithm",Department of Electronics Engineering National Lien-Ho College of Technology and Commerce, Miaoli, Taiwan,(2001) Republic of China
23. Shuqun Zhang and A. Karim, "Color image encryption using double random phase encoding",Microwave and Optical Technology, Vol.21, No.5, June 5 1999
24. Young-Chang Hou, "Visual cryptography for color images", Pattern Recognition 36 (2003), www.elsevier.com/locate/patcog, 1619-1629
25. M Kwan, —The Design of the ICE Encryption Algorith, in proc. OfFast software Encryption workshop, 1997.

SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION

By

GEETHA.P

Reg. No. 1020106005

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University, Coimbatore)

COIMBATORE - 641049

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

APRIL 2012

BONAFIDE CERTIFICATE

Certified that, this project report entitled "**SECURE CRYPTOSYSTEM FOR IMAGE ENCRYPTION AND DECRYPTION**" is the bonafide work of Ms.Geetha.P [Reg. no. 1020106005] who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide	Head of the Department
Prof.S.Govindaraju	Dr. Ms. Rajeswari Mariappan

The candidate with university Register no. 1020106005 is examined by us in the project viva-voce examination held on ……

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I express my profound gratitude to our director **J.Shanmugham,** for giving this opportunity to pursue this course

At this pleasing moment of having successfully completed the project work, I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved Principal **Prof.Ramachandran,** for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr.Rajeswari Mariyappan Ph.D.,** the ever active, Head of the Department of Electronics and Communication Engineering, who rendering us all the time by helps throughout this project.

I extend my heartfelt thanks to my internal guide **Prof.S.Govindaraju,** for his ideas and suggestion, which have been very helpful for the completion of this project work. His careful supervision has ensured me in the attaining perfection of work.

In particular, I wish to thank and everlasting gratitude to the project coordinator **Asst.Prof.R.Hemlatha**, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering Department for their encouragement and support throughout the course of this project.

ABSTRACT

In this project, a secure image cryptosystem based on chaos and Brahmagupta and Bhaskara (BB) equation is designed and its hardware architecture are proposed. In a chaotic binary sequence, the gray level of each pixel is XORed or XNORed bit-by-bit to one of the two predetermined keys. This new scheme is used to modify the pixel values with pseudorandom sequences generated by linear feedback shift register (LFSR). Based on the characteristics of the pseudo-random, unpredictability of chaotic sequences, extreme sensitivity to the initial conditions are very high. The simulation results show that the image encrypted by this new scheme cannot be recognized. Its features are low computational complexity, high security and no distortion. In order to implement the algorithm, VLSI architecture with low hardware cost, high computing speed, and high hardware utilization efficiency is also proposed. Further the proposed cryptosystem is compared with the existing Image Encryption Using Transformation Algorithm(IETA).

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF ABBREVIATIONS	xi
1	**INTRODUCTION**	1
	1.1 Overview of the Project	2
	1.2 Introduction to VHDL	3
	1.2.1 Structural Descriptions	3
	1.2.2 Dataflow Descriptions	5
	1.2.3 Behavioral Descriptions	5
	1.3 Softwares used	7
	1.4 Project Goal	7
2	**CRYPTOSYSTEM**	7
	2.1 Cryptosystem	7
	2.2 Cryptography	7
	2.3 Fundamentals of cryptography	7
	2.4 Modern cryptography	8
	2.4.1 Symmetric key cryptography	8
	2.4.2 Public key cryptography	8
	2.5 Cryptanalysis	9
	2.6 Particularities of Image Encryption	11
	2.7. Existing Image Encryption Schemes	14
	2.8. Chaos-Based Encryption Schemes	16
	2.8.1. Basic Features of Chaos	17
	2.8.2. Definition of discrete chaos	17
	2.8.3. Relationships Between Chaos and Cryptography	18
	2.8.4. Chaos for Cryptography	18

	2.9 Chaos based Image Encryption	19
	2.10 BB Equation overview	19
	2.10.1 Application of BB equation in cryptography	20
3	**LITERATURE SURVEY**	
	3.1 IMAGE ENCRYPTION ALGORITHMS	21
4	**IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM**	25
	4.1 Description of the transformation algorithm	25
	4.2 Algorithm for creating transformation table	27
	4.3 Algorithm for performing transformation	27
5	**ENCRYPTION & DECRYPTION ALGORITHM**	
	5.1 Proposed algorithm for Encryption	28
	5.1.1 Architecture of Encryption Processing Unit	31
	5.2 Proposed algorithm for Decryption	32
	5.2.1 Architecture of Decryption Processing Unit	33
6	**PROPOSED CRYPTOSYSTEM**	35
	6.1 Generation of Pseudorandom sequence generator	36
	6.1.1 Uses in Cryptography	37
7	**RESULTS & DISCUSSION**	38
	7.1 Simulation Results	40
	7.1.1 Simulation result of existing IETA algorithm	40
	7.1.2 Simulation result of Proposed Encryption Algorithm	41
	7.1.3 Simulation result of Proposed Decryption Algorithm	42

	7.2 Synthesis Results	43
	7.2.1 Power Report	43
	7.2.1.1 Existing IETA Algorithm	43
	7.2.1.2 Proposed Chaos & BB Equation	44
	7.2.2 Area Report	45
	7.2.2.1 Existing IETA Algorithm	45
	7.2.2.2 Proposed Chaos & BB Equation	46
	7.2.3 Delay Report	47
	7.2.3.1 Existing IETA Algorithm	47
	7.2.3.2 Proposed Chaos & BB Equation	47
	7.3 Comparison	48
8	**RESULTS**	
	8.1 Image Results of Existing IETA	49
	8.2 Image Results of Proposed Chaos & BB Equation	51
	8.2.1. Results for Grayscale image	50
	8.2.2. Results for Color image	51
	8.2.3. Results for fingerprint image	52
9	**CONCLUSION & FUTURE WORK**	53
	REFERENCES	54

LIST OF FIGURES

FIGURE NO	CAPTION	PAGE NO
1.1	Project flow	2
1.2	Schematic SR latch	4
4.1	General Block Diagram of Transformation Algorithm	26
5.1	Architecture of the Encryption Unit	30
5.2	Cascade Architecture of EPE	31
5.3	Architecture of EPE 1	31
5.4	Architecture of EPE 2	31
5.5	Cascade Architecture of DPE	33
5.6	Architecture of DPE 1	34
5.7	Architecture of DPE 2	34
6.1	Proposed Cryptosystem	35
6.2	8 bit LFSR Structure	36
7.1	Conceptual Overview of Modelsim	38
7.2	Simulation Result of IETA Algorithm	40
7.3	Simulation Result of Proposed Encryption Algorithm	41
7.4	Simulation Result of Proposed Encryption Algorithm	42
7.5	Power Report of IETA Algorithm	43
7.6	Power Report of Proposed Cryptosystem	44
8.1	Lena image	49
8.2	Encrypted Image	49
8.3	Decrypted Image	49
8.4	Lena image	50
8.5	Encrypted Image	50
8.6	Decrypted Image	50
8.7	Lena image	51

8.8	Encrypted Image	51
8.9	Decrypted Image	51
8.10	Lena image	52
8.11	Encrypted Image	52
8.12	Decrypted Image	52

LIST OF TABLES

TABLE NO	CAPTION	PAGE NO
1	Similarities and differences between chaos and cryptography	18
2	Comparison between IETA based Cryptosystem and proposed Cryptosystem	48

LIST OF ABBREVIATIONS

BB	-------	**Encryption Processing Element**
ISE	-------	**Integrated Software Environment**
IETA	-------	**Image Encryption using Transformation Algorithm**
EPU	-------	**Encryption Processing Unit**
EPE	-------	**Encryption Processing Element**
DPU	-------	**Decryption Processing Unit**
DPE	-------	**Decryption Processing Element**
VLSI	-------	**Very Large Scale Integration**
FPGA	-------	**Field Programmable Gate Array**
VHDL	-------	**Very High Speed Integrated Circuit Hardware Description Language**

CHAPTER 1
INTRODUCTION

The major concern while transmitting signals is the security. The security concerns are growing due to the rampant illegal data access. To protect the valuable information in many applications like medical imaging, military image database, communications and confidential video conferencing, there is a need to secure the images by the use of encryption and decryption algorithms. In such a scenario, to avoid information leakage to both active and passive attackers, encryption of the medical images is very important. The sensitivity to initial conditions and control parameters has led to the development of chaos-based encryption and decryption algorithms.

The use of chaotic signal for secure data transmission has seen a significant growth in developing chaos-based encryption and decryption algorithms. However, a number of chaos-based algorithms have been shown to be insecure. A modified chaotic key based algorithm with increased key size is developed in for improved security and VLSI architecture of it is developed and realized using Xilinx ISE VLSI software. A cryptosystem based on Brahmagupta-Bhaskara (BB) equation was vulnerable to known plaintext attacks. Equation-based approaches, with moderate size of keys; it is possible to develop algorithms with high security.

Hence, in this project based on chaos and the BB equation, new algorithms are developed for image encryption and decryption. Further, the hardware implementation and VLSI architectures of the proposed algorithms are developed and realized using Xilinx ISE VLSI software. Furthermore, the proposed algorithm is compared with the existing IETA algorithm in terms of area, speed and power.

1.1. OVERVIEW OF THE PROJECT

The block diagram of the proposed cryptosystem for encryption and decryption is shown in Figure 1.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XORed or XNORed bit-by-bit to one of the predetermined keys.

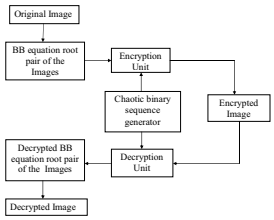

Fig.1.1: Project flow

1.2. INTRODUCTION TO VHDL

VHDL is an acronym which stands for VHSIC Hardware Description Language. VHSIC is yet another acronym which stands for Very High Speed Integrated Circuits. It is being used for documentation, verification, and synthesis of large digital designs. VHDL is a standard (VHDL-1076) developed by IEEE. The different approaches in VHDL are structural, data flow, and behavioral methods of hardware description.

1.2.1 STRUCTURAL DESCRIPTIONS

The structural descriptions are explained below with examples. Every portion of a VHDL design is considered a Block (Building Blocks). A VHDL design may be completely described in a single block, or it may be decomposed in several blocks. Each block in VHDL is analogous to an off-the-shelf part and is called an entity. The entity describes the interface to that block and a separate part associated with the entity describes how that block operates. The interface description is like a pin description in a data book, specifying the inputs and outputs to the block.
The following is an example of an entity declaration in VHDL.

```
entity latch is
    port (s,r: in bit;
          q,nq: out bit);
end latch;
```

The first line indicates a definition of a new entity, whose name is latch. The last line marks the end of the definition. The lines in between, called the port clause, describe the interface to the design. The port clause contains a list of interface declarations. Each interface declaration defines one or more signals that are inputs or outputs to the design. Each interface declaration contains a list of names, a mode, and a type.
The following is an example of an architecture declaration for the latch entity.

```
Architecture dataflow of latch is
    signal q0 : bit := '0';
    signal nq0 : bit := '1';
begin
    q0<=r nor nq0;
```



```
    nq0<=s nor q0;
    nq<=nq0;
    q<=q0;
end dataflow;
```

The first line of the declaration indicates that this is the definition of a new architecture called dataflow and it belongs to the entity named latch. So this architecture describes the operation of the latch entity. The schematic for the latch might be

Fig 1.2: Schematic SR Latch

We can specify the same connections that occur in the schematic using VHDL with the following architecture declaration:

```
Architecture structure of latch is
    component nor_gate
    port (a,b: in bit;  c: out bit);
    end component;
    begin
      n1: nor_gate
       port map (r,nq,q);
      n2: nor_gate
    port map (s,q,nq);
    end structure;
```

The lines between the first and the keyword begin are a component declaration. A list of components and there connections in any language is sometimes called a netlist. The structural description of a design in VHDL is one of many means of specifying netlists.

1.2.2 DATA FLOW DESCRIPTIONS

In the data flow approach, circuits are described by indicating how the inputs and outputs of built-in primitive components are connected together. Suppose we were to describe the following SR latch using VHDL as in the following schematic.

```
entity latch is
        port (s,r : in bit;
        q,nq : out bit);
end latch;
architecture dataflow of latch is
        begin
        q<=r nor nq;
        nq<=s nor q;
end dataflow;
```

The signal assignment operator in VHDL specifies a relationship between signals, not a transfer of data as in programming languages. The architecture part describes the internal operation of the design. The scheme used to model a VHDL design is called discrete event time simulation. In this the values of signals are only updated when certain events occur and events occur at discrete instances of time. The above mentioned SR latch works with this type of simulation.

1.2.3. BEHAVIORAL DESCRIPTIONS

The behavioral approach to modeling hardware components is different from the other two methods in that it does not necessarily in any way reflect how the design is implemented. **Since encryption and decryption are step by step process, it needs behavioral description in VHDL.**
A. Behavioral descriptions are supported with the Process Statements. The process statement can appear in the body of an architecture declaration just as the signal assignment statement does. The process statement can also contain signal assignments in order to specify the outputs of the process.

B. A Variable is used to hold data and also it behaves like you would expect in a software programming language, which is much different than the behavior of a signal. Although variables represent data like the signal, they do not have or cause events and are modified differently. Variables are modified with the variable assignment.

There are several statements that may only be used in the body of a process.These statements are called *Sequential Statements* because they are executed sequentially. The types of statements used here are if, if else, for and loop.

Next are about the *Signals and Processes*. This section is short, but contains important information about the use of signals in the process statement. A signal assignment, if anything, merely schedules an event to occur on a signal and does not have an immediate effect. When a process is resumed, it executes from top to bottom and no events are processed until after the process is complete.

Final Section in the behavioral description discusses about the *Program Output Methods*. In most programming languages there is a mechanism for printing text on the monitor and getting input from the user through the keyboard. It can able to give output certain information during simulation. A standard library that comes with every VHDL language system. In VHDL, common code can be put in a separate file to be used by many designs. This common code is called a library. In order to use the library that provides input and output capabilities you must add the statement use textio.all; immediately before every architecture that uses input and output. The write statement can be used to append constant values and the value of variables and signals of the types bit, bit_vector, time, integer, and real.

1.3. SOFTWARES USED

- Matlab 7 R2008b
- Xilinx ISE 9.2i

1.4. PROJECT GOAL

The goal of the project is to provide a secure cryptosystem for images. A new algorithm based on chaos and BB equation are proposed for image encryption and decryption. For practical use, VLSI architectures of the proposed algorithms are designed and realized using Xilinx ISE VLSI software for hardware implementation. Finally the proposed algorithm is compared with the existing transformation algorithm in terms of area, speed and power.

CHAPTER 2
CRYPTOSYSTEM

2.1. CRYPTOSYSTEM

A cryptographic system is any computer system that involves cryptography. Such systems include for instance, a system for secure electronic mail which might include methods for digital signatures, cryptographic hash functions, key management techniques, and so on.

Typically, a cryptosystem consists of three algorithms: one for key generation, one for encryption, and one for decryption.

2.2. CRYPTOGRAPHY

Cryptography is the study of techniques for secure communication in the presence of third parties generally, it is related to various aspects in information security such as data confidentiality, data integrity, and authentication. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

2.3. FUNDAMENTALS OF CRYPTOLOGY

The basic idea of encryption is to modify the message in such a way that its content can be reconstructed only by a legal recipient. A discrete-valued cryptosystem can be characterized by

- a set of possible plaintexts, P
- a set of possible ciphertexts, C
- a set of possible ciphertkeys, K
- a set of possible encryption and decryption transformations, E and D

For each key, k ∈ K, there exists an encryption function e(k) ∈ E and a corresponding decryption function d(k) ∈ D, such that for each plaintext p ∈ P the condition for unique decoding, d(k, e(k, p)) = p, is satisfied.

The security of a cryptosystem usually relies on the key only. In other words, it is assumed that the opponent knows the structure of the encryption system, has the ciphering algorithm, and has access to the transmission channel to obtain an arbitrary segment of the ciphertext c.

A good cipher should have strong ability to withstand all kinds of cryptanalysis and attacks that try to break the system. To a certain extent, the resistance against attacks is a good measure of the performance of a cryptosystem; thus, it is often used to evaluate cryptosystems.

2.4. MODERN CRYPTOGRAPHY

2.4.1. SYMMETRIC-KEY CRYPTOGRAPHY

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government. Despite its deprecation as an official standard, DES remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access.

2.4.2. PUBLIC-KEY CRYPTOGRAPHY

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. In a public-key encryption system, the *public key* is used for encryption, while the *private* or *secret key* is used for decryption.

2.5. CRYPTANALYSIS

Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

According to the method of the opponent's access to additional information, attacks on a cryptosystem may be classified into four classes:

• **Ciphertext-only attack:** Opponent has access to communication channel and can eavesdrop some segments of the ciphertext, encrypted by a certain key. The task of the opponent is to reveal as much plaintext as possible, and even to be able to deduce the cipher key.
• **Known-plaintext attack:** In addition to the obtained ciphertext segments, the opponent knows also an associated piece of plaintext. The task of the opponent is then to deduce the cipher key.
• **Chosen-plaintext attack:** The opponent not only has access to some segments of the cipher and the plaintext, but also can choose plaintext to encrypt and accordingly gets some corresponding ciphertext that he wants for comparison. This kind of attack is more intensive than the known-plaintext attack.
• **Chosen-ciphertext attack:** The opponent can choose different segments of the ciphertext and accordingly get its corresponding plaintext.

Apart from the aforementioned typical attacks, there is a type of attack named exhaustive key search, which tries all possibilities for the key in the keyspace to completely decrypt the plain message. If the keyspace of a cipher is relatively small, this exhaustive searching works quite well, given the availability of supercomputing power today. It should be emphasized that any encryption algorithm, traditional or chaos-based, should obey basic cryptographical principles in order to be able to resist serious attacks.

CHAPTER 2
CRYPTOSYSTEM

2.6. Salient Features of Image Encryption

Unlike text messages, image data have special features such as high redundancy and high correlation among pixels, not to mention that they usually are huge in size, which together make traditional encryption methods difficult to apply and slow to process. Sometimes image applications also have their own requirements like real-time processing, fidelity reservation, image format consistency, and data compression for transmission. Simultaneous fulfillments of these requirements, alongwith high security and high quality demands, have presented great challenges to real-time imaging practice.

One example is the case where one needs to manage both encryption and compression. In doing so, if an image is to be encrypted after its format is converted, say from a TIFF file to a GIF file, encryption has to be implemented before compression. However, a conventional encrypted image has very little compressibility. On the other hand, compression will make a correct and loss-less decipher impossible, particularly when a highly secure image encryption scheme is used. This conflict between the compressibility and the security is very difficult, if not impossible, to completely resolve.

The salient features of image encryption may be summarized as follows:

- High redundancy and large size generally make encrypted image data vulnerable to attacks via cryptanalysis. Due to its size, the opponent can gain enough cipher text samples (even from one picture) for statistical analysis. Meanwhile, since data in images have high redundancy, adjacent pixels likely have similar grayscale values, or image blocks have similar patterns, which usually embed the image with certain patterns that result in secret leakage.

- Image data have strong correlations among adjacent pixels, which makes fast data-shuffling quite difficult. Statistical analysis on large numbers of images shows that averagely adjacent 8 to 16 pixels are correlative in the horizontal, vertical, and also diagonal directions for both natural and computer-graphical images. According to Shannon's information theory, a secure cryptosystem

should fulfill a condition on the information entropy,E(P|C) = E(P) where P stands for plain message and C for ciphered message; that is, the ciphered (i.e., encrypted) image should not provide any information about the plain image. To meet this requirement, therefore, the ciphered image should be presented as randomly as possible. Since a uniformly distributed message source has a maximum uncertainty, an ideal cipher image should have an equilibrium histogram, and any two adjacent pixels should be uncorrelated statistically. This goal is not easy to achieve under only a few rounds of permutation and diffusion.

- Huge size image data also makes real-time encryption difficult. Compared with texts, image data capacity is horrendously large. For example, a common 24-bit true-color image of 512-pixel height and 512-pixel width occupies 512 × 512 × 24/8 = 768 KB in space. Thus, a one-second motion picture will reach up to about 19 MB. Real-time processing constraints are often required for imaging applications, such as video conferencing, image surveillance, and so on. Vast amounts of image data put a great burden on the encoding and decoding processes. Encryption during or after the encoding phase, and decryption during or after the decoding phase, will aggravate the problem. If an encryption algorithm runs very slowly, even with high security, it would have little practical value for real-time imaging applications. That is the reason why current encryption methods such as DES, IDEA, and RSA are not the best candidates for this consideration.

- Image encryption is often to be carried out in combination with data compression. In almost all cases, the data are compressed before they are stored or transmitted due to the huge amount of image data and their very high redundancy. Thus, directly incorporating security requirements in the data compression system is a very attractive approach. The main challenge is how to ensure reasonable security while reducing the computational cost without downgrading the compression performance.

- In image usage, file format conversion is a frequent operation. It is desirable that image encryption not affect such an operation. Thus, directly treating image data as ordinary data for encryption will make file format conversion impossible. In this scenario, content encryption, where only the image data are encrypted, leaving file header and control information unencrypted, is preferable.

- Human vision has high robustness to image degradation and noise. Only encrypting those data bits tied with intelligibility can efficiently accomplish image protection. However, conventional cryptography treats all image data bits equally in importance, and thus requires a considerable amount of computational power to encrypt all of them, which has often proved unnecessary.

- In terms of security, image data are not as sensitive as text information. Security of images is largely determined by the real situation in an application. Usually, the value of the image information is relatively low, except in some specific situations like military and espionage applications or video conferencing in business. A very expensive attack of encrypted median data is generally not worthwhile. In practice, many image applications do not have very strict security requirements. Under certain circumstances, protection of the fidelity of an image object is more important than its secrecy. An example is electronic signatures.

Currently, there does not seem to be any image encryption algorithm that can fulfill all the aforementioned specifications and requirements. Chaos-based image encryption, can provide a class of very promising methods that can partially fulfill many of these requirements and demonstrate superiority over the conventional encryption methods, particularly with a good combination of speed, security, and flexibility.

2.7. Existing Image Encryption Schemes

Some image encryption methods have been proposed in the current literature. In order to inspire the development of better chaotic ciphers, this review is not only intended for chaos-based methods, but is also meant for understanding image encryption technology in general. Image encryption algorithms, which can be classified with respect to the approach in constructing the scheme, are divided into two groups here: chaos- based methods and non-chaos-based methods. Image encryption also can be divided into full encryption and partial encryption (also called selective encryption) according to the percentage of the data encrypted. Moreover, they can be classified into compression-combined methods and non compression methods.

Some existing proposals of chaos-based image encryption algorithms are now introduced. In this two kinds of schemes based on higher-dimensional chaotic maps were proposed. By using a discretized chaotic map, pixels in an image are permuted in shuffling after several rounds of operations. Between every two adjacent rounds of permutations, a diffusion process is performed, which can significantly change the distribution of the image histogram that makes statistical attack infeasible. Empirical testing as well as cryptanalysis both demonstrated that the chaotic baker map and cat map are good candidates for this kind of image encryption. The aforementioned schemes are block cipher, and they have some prominent merits, including high security and fast processing. However, their defects are also significant since the encrypted image has very little compressibility and is unable to abide any lossy compression (e.g., JPEG). To alleviate the conflict between compressibility and encryption, several suggestions of combining compression and encryption have been proposed. The so-called MHT scheme was proposed that encrypts image via a manipulation of Huffman coding tables in the image coding system. The MHT scheme chooses several different Huffman tables from a large number of possible candidates, and uses them alternatively to encode the image data. The choice of Huffman tables and the order in which they are used are kept secret as the key. It was advocated that the method requires very little computational overhead and can be applied to MPEG and JPEG/JPEG 2000, but it cannot resist chosen-plaintext attack.

2.8.1. Basic Features of Chaos

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behaviors. Since its discovery by Edward N. Lorenz in 1963, chaos theory has become a branch of scientific studies today. Since discrete chaotic dynamic systems (i.e., maps) are used in cryptography, this notion is briefly introduced.

2.8.2. Definition of discrete chaos

There are several definitions of chaos, which are similar but are actually not equivalent.

For simplicity, one-dimensional maps are discussed. Consider a discrete dynamical system in the general form of

$$x_{k+1} = f(x_k), \quad f : I \to I, \ x_0 \in I \qquad \text{..... (1)}$$

where f is a continuous map on the interval I = [0, 1]. This system is said to be chaotic if the following conditions are satisfied:

1. Sensitive to initial conditions:
$$\exists \delta > 0 \ \forall x_0 \in I, \varepsilon > 0 \ \exists n \in \mathbb{N}, y_0 \in I : |x_0 - y_0| < \varepsilon \Rightarrow |f^n(x_0) - f^n(y_0)| > \delta \qquad \text{..... (2)}$$
2. Topological transitivity:
$$\forall I_1, I_2 \subset I \ \exists x_0 \in I_1, n \in \mathbb{N} : f^n(x_0) \in I_2. \qquad \text{.....(3)}$$
3. Density of periodic points in I:
Let,
$$P = \{p \in I | \exists n \in \mathbb{N} : f^n(p) = p\}$$
be the set of periodic points of f. Then P is dense in I. P = I

2.8.3. Relationships Between Chaos and Cryptography

The similarities and differences between the two subjects can be shown in Table 1. Chaotic maps and cryptographic algorithms have some similar properties: both are sensitive to tiny changes in initial conditions and parameters; both have random like behaviors; and cryptographic algorithms shuffle and diffuse data by rounds of encryption, while chaotic maps spread a small region of data over the entire phase space via iterations. The only difference in this regard is that encryption operations are defined on finite sets of integers while chaos is defined on real numbers.

2.8.4. Chaos for Cryptography

It is natural to apply the discrete chaos theory to cryptography for the following reasons :

The property of sensitive dependence of orbits on initial conditions makes the nature of encryption very complicated. Suppose that one has the following chaos-based encryption scheme:

Chaotic systems	Cryptographic algorithms
Phase space: set of real numbers	Phase space: finite set of integers
Iterations	Rounds
Parameters	Key
Sensitivity to initial conditions and Parameters	Diffusion

Table 1. Similarities and differences between chaos and cryptography

A somewhat different chaos-based image encryption method was proposed in that makes use of the SCAN language. Through substitution of each pixel based on an additive noise vector and scramble scanning patterns, an image can be encrypted and compressed simultaneously. The idea seems to be quite good, but it was pointed out in [17] that this method is weak against exhaustive key searching and chosen-plaintext attacks. In another image compression and encryption algorithm was proposed based on the lossless quadtree image compression scheme. The quadtree data structure is used to represent the image, and the scanning sequences of image data comprise a private key for encryption. Also numerous attacks on the proposed algorithm were tested and presented, which include key space reduction, histogram attack, known-plaintext attack, and chosen-plaintext attack.

In order to speed up encryption processes so as to make them feasible for real-time applications, most of the existing schemes follow the idea of selective encryption. Actually, according to Shannon's theory, both encryption and compression are processes of redundancy reduction, but their purposes are different. Several partial encryption schemes were provided. It was reported that by a partial encryption, only 13% to 27% of the output from a quadtree compression algorithm is encrypted for a typical image, and less than 2% is encrypted for a 512×512 image compressed by set-partitioning in the hierarchical trees algorithm. There are also several proposed schemes that consider matching the compatibility to current international standards. Since many international standards on videos and images use block-based discrete cosine transform (DCT), including the familiar JPEG, MPEG-1, MPEG-2, H.261, and H.263 formats, the current research has been concentrated on selective encryptions within the framework of DCT. However, with the emergence of MPEG-4 and JPEG-2000, research emphasis may soon be redirected to a combination of encryption and wavelet compression.

To achieve high encryption speed, in the early stage some elementary cryptographic methods using random permutation lists were suggested. Since the operations are simple, the encryption does not require high computational cost. The challenge is how to achieve reasonable security with such simple operations.

2.8. Chaos-Based Encryption Schemes

A great deal of work on application of chaos to cryptography has been carried out in the last decade. Early works on chaos in cryptography were connected with encrypting messages through modulation of chaotic orbits of continuous-time dynamical systems. These methods are strongly related to the concept of synchronization of two chaotic systems and to chaos control. Several different ways have been proposed to achieve synchronization of chaotic systems, thereby transmitting information on a chaotic carrier signal.

The following technical problems were listed in :

• It is difficult to determine the synchronization time; therefore, the message during the transient period will be lost, sometimes cause fairly long transient times.

• Noise throughout the transmission significantly affects the intended synchronization. This means the synchronization noise intensity should be small compared to the signal level, or the desired synchronization will not be achieved.

• Technically, it is difficult to implement two well-matched analog chaotic systems, which are required in synchronization, and if this is not required (i.e., with certain robustness) then the opponent can also easily achieve the same synchronization for attack. In contrast to synchronization-based techniques, a direct application of a chaotic transformation to a plaintext, or applying a chaotic signal in the design of an encryption algorithm, seems to be a more promising approach.

The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

2.9. CHAOS-BASED IMAGE ENCRYPTION

Chaos is a ubiquitous phenomenon existing in deterministic nonlinear systems that exhibit extreme sensitivity to initial conditions and have random like behaviors. The sensitivity to initial conditions and parameters as well as the mixing (ergodicity) characteristics of chaos are very beneficial to cryptosystems. The main difference is that cryptosystems are operated on a finite set of integers, while chaotic maps are defined on an infinite set of real numbers. Therefore, how to merge these two kinds of systems so as to take advantage of the good properties of chaos is worthy of further exploration.

Many chaos-based encryption schemes have been proposed, and some of them have been extended from text encryption to image encryption. A direct extension of a chaos-based text encryption scheme to also work for images is possible, but this simple modification may not provide an efficient solution to these image encryption problems. Image encryption has its own specifications such as encryption speed, compatibility to image format and compression standards, and real-time implementation, therefore it requires a special design of the encryption algorithm.

An encryption method called chaotic key-based algorithm (CKBA) was proposed. The algorithm first generates a time series based on a chaotic map, and then uses it to create a binary sequence as a key. According to the binary sequence generated, image pixels are rearranged and XOR or XNOR operated with the selected key. This method is very simple but has defects in Security. This method is very weak to the chosen/known-plaintext attack with only one plain image. Moreover, its security to brute-force attack is also questionable.

2.10. BB EQUATION OVERVIEW

The Brahmagupta–Bhaskara (BB) equation is a Quadratic Diophantine equation of the form $NX^2 + k = Y^2$, Where k is an integer and N is a positive integer such that $\sqrt{N}$ is irrational. A particular case of the BB equation with k = 1 is also known as Pell equation. This equation in the Galois Field GF(p), has some useful properties. Application of these properties in two different fields of cryptography, namely, digital encryption and user authentication. For those applications, where software computation of the roots of the BB equation is unacceptable for being too slow, a hardware architecture for using the BB equation in GF(p) is given that is useful for implementation in VLSI form.

2.10.1. APPLICATION OF BB EQUATION IN CRYPTOGRAPHY

The BB-equation in Galois Field GF(p), can be written as,

$$[n(x^2)_p + 1 = (y^2)_p \qquad (1)$$

where p is an odd prime.The alternative representation of BB equation in GF (p) can be rewritten as,

$$(nq_x+1)_p=(q_y)_p \qquad (2)$$

where $q_x = (x^2)_p$, $q_y = (y^2)_p$ and the subscript p stands for modulo operation by p on the argument values of the expressions. The application of the BB equation for encryption depends on the following two properties.

1. Given n and p, with p>n>0, it is always possible to obtain q_x and q_y corresponding to the roots of the BB equation $(nx^2 + 1)p = (y^2)p$.
2. Given q_x and q_y corresponding to any root of the BB equation $(nx^2 + 1)p = (y^2)p$, it is always possible to compute uniquely, the corresponding value of n , only with the knowledge of p.

The encryption process based on the BB equation is as follows.

1. n corresponds to the cleartext or plaintext in a block that is being encrypted.
2. p corresponds to the primary secret key used in the encryption of the plaintext in a block.
3. The ciphertext corresponding to n is the pair (q_x,q_y) of the corresponding BB equation.

CHAPTER 3
LITERATURE SURVEY

Aloka Sinha and Kehar Singh [18] have proposed a new technique to encrypt an image for secure image transmission.The digital signature of the original image is added to the encoded version of the original image. Image encoding is done by using an appropriate error control code, such as a Bose-Chaudhuri Hocheguenghem (BCH) code. At the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image.

S.S. Maniccam and N.G. Bourbakis [19] have presented a new methodology which performs both lossless compression and encryption of binary and gray-scale images. The compression and encryption schemes are based on SCAN patterns generated by the SCAN methodology. The SCAN is a formal language-based two-dimensional spatial-accessing methodology which can efficiently specify and generate a wide range of scanning paths or space filling curves.

Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen [20] use one of the popular image compression techniques, vector quantization to design an efficient cryptosystem for images. The scheme is based on vector quantization (VQ), cryptography, and other number theorems.In VQ, the images are first decomposed into vectors and then sequentially encoded vector by vector. Then traditional cryptosystems from commercial applications can be used.

Jiun-In Guo and Jui-Cheng Yen [21] have presented an efficient mirror-like image encryption algorithm. Based on a binary sequence generated from a chaotic system, an image is scrambled according to the algorithm. This algorithm consists of 7 steps.Step-1 determines a 1-D chaotic system and its initial point x(0) and sets k = 0.
Step-2 generates the chaotic sequence from the chaotic system.
Step-3 generates binary sequence from chaotic system.
Steps-4,5,6, and 7 rearrange image pixels using swap function according to the binary sequence.

Jui-Cheng Yen and Jiun-In Guo [22] have proposed a new image encryption scheme based on a chaotic system. In their method, an unpredictable chaotic sequence is generated. It is used to create a binary sequence again. According to the binary sequence, an image's pixels are rearranged. This algorithm has four steps.
Step-1 determines a chaotic system and its initial point x(0), row size M and column size N of the image f, iteration number no, and constants α, β and μ used to determine the rotation number.
Step-2 generates the chaotic sequence from the chaotic system.
Step-3 generates the binary sequence.
Step-4 includes special functions to rearrange image pixels. These functions are;

$ROLR_i^{\alpha,p} : f \to f'$ is defined to rotate each pixel in the ith row in f, 0 ≤ i ≤ M ≤1, in the left direction p pixels if1 equals 0 or in the right direction p pixels if1 equals 1.

$ROUD_i^{\alpha,p} : f \to f'$ is defined to rotate each pixel in the ith column in f, 0 ≤ j ≤ N ≤1, in the up direction p pixels if1 equals 0 or in the down direction p pixels if1 equals 1.

$ROUR_i^{k,p} : f \to f'$ is defined to rotate each pixel at position (x,y) in the image f such that x+y=k, 0 ≤ k ≤ M + N - 2 , , in the upper-right direction p pixels if1 is equal to 1 or in the lower-left direction p pixels if1 is equal to 0.

Shuqun Zhang and Mohammad A. Karim [23] have proposed a new method to encrypt color images using existing optical encryption systems for gray-scale images. The color images are converted to their indexed image formats before they are encoded. In the encoding subsystem, image is encoded to stationary white noise with two random phase masks, one in the input plane and the other in the Fourier plane. At the decryption end, the color images are recovered by converting the decrypted indexed images back to their RGB (Red-Green-Blue) formats. The proposed single-channel color image encryption method is more compact and robust than the multi channels methods.

Visual cryptography uses the characteristics of human vision to decrypt encrypted images. It needs neither cryptography knowledge nor complex computation. For security concerns, it also ensures that hackers cannot perceive any clues about a secret image from individual cover images. Young-Chang Hou [24] have proposed three methods for visual cryptography.

CHAPTER 4
IMAGE ENCRYPTION USING TRANSFORMATION ALGORITHM

In most of the natural images, the values of the neighboring pixels are strongly correlated (i.e. the value of any given pixel can be reasonably predicted from the values of its neighbors) . In order to dissipate the high correlation among pixels and increase the entropy value, we propose a transformation algorithm that divides the image into blocks and then shuffles their positions before it can be encrypted with the secrete key. By using the correlation and entropy as a measure of security, this process results in a lower correlation, a higher entropy value and thus improving the security level of the encrypted images. The variable secret key of the transformation process is the main key is used to increase the entropy.

4.1. DESCRIPTION OF THE TRANSFORMATION ALGORITHM

The transformation technique works as follows:

The original image is divided into a random number of blocks that are then shuffled within the image. The generated (or transformed) image is then fed to the encryption algorithm. The main idea is that an image can be viewed as an arrangement of blocks. The intelligible information present in an image is due to the correlation among the image elements in a given arrangement. This perceivable information can be reduced by decreasing the correlation among the image elements using certain transformation techniques.

The secret key of this approach is used to determine the seed. The seed plays a main role in building the transformation table, which is then used to generate the transformed image with different random number of block sizes. The transformation process refers to the operation of dividing and replacing an arrangement of the original image.The image can be decomposed into blocks; each one contains a specific number of

pixels. The blocks are transformed into new locations. For better transformation the block size should be small, because fewer pixels keep their neighbors. In this case, the correlation will be decreased and thus it becomes difficult to predict the value of any given pixel from the values of its neighbors. At the receiver side, the original image can be obtained by the inverse transformation of the blocks. A general block diagram of the transformation method is shown in Fig.4.1.

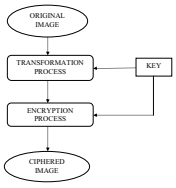

Fig.4.1. General block diagram of the transformation algorithm

Gray-level visual cryptography method first transforms the gray-level image into a halftone image and then generates two transparencies of visual cryptography. Obviously, we indeed cannot detect any information about the secret image from the two sharing transparencies individually, but when stacking them together, the result clearly shows the secret image.Method 1 uses four halftone images, cyan, magenta, yellow and black, to share the secret image. The codes of the four sharing images are fully disordered, and we cannot perceive any clue of the original secret image from any single sharing image. Method 2 reduces the inconvenience of Method 1 and requires only two sharing images to encrypt a secret image. However, after stacking the sharing images generated by Method 2, the range of color contrast will be 25% of that of the original image. Method 3 loses less image contrast, which is better than Method 2.

In cryptography, ICE (Information Concealment Engine) is a block cipher published by Kwan in 1997. The algorithm is similar in structure to DES, but with the addition of a key-dependent bit permutation in the round function. The key-dependent bit permutation is implemented efficiently in software. The ICE algorithm is not subject to patents, and the source code has been placed into the public domain.The ICE algorithm was designed for use in software applications. Those applications however are slow due to the use of modular arithmetic [25]. So the need for faster implementations is great. That can be achieved through hardware implementations. Considering the fact that hardware implementations are generally faster and more Ease of Use reliable than software implementations the outcome of a hardware design is even more interesting.

The system operates for the both encryption and decryption processes and has been optimized for low hardware resources and for high-speed performance. The proposed architecture has very encouraging performance result in terms of speed and throughput. This makes the design very useful in current applications that use DES as the base of a cryptographic protocol [25].

ICE is a Feistel network with a block size of 64 bits. The standard ICE algorithm takes a 64-bit key and has 16 rounds. A fast variant, Thin-ICE, uses only 8 rounds. An open-ended variant ICE-n, uses 16n rounds with 64n bit key. They described an attack on Thin-ICE which recovers the secret key using 223 chosen plaintexts with a 25% success probability. If 227 chosen plaintexts are used, the probability can be improved to 95%. For the standard version of ICE, an attack on 15 out of 16 rounds was found, requiring 256 work and at most 256 chosen plaintexts.

ICE is a standard Feistel block cipher [25] with a structure similar to DES. It takes a 64-bit plaintext, which is split into two 32-bit halves. In each round of the algorithm the right half and a 60-bit subkeys are fed into the function F. The output of F is XORed with the left half, and then the halves are swapped. This is the Transformation Round of the ICE algorithm. This process is repeated for 16 rounds. However the final swap is left out. The decryption process is the same, except that the sub keys are used in reverse order.

The advantages of Feistel structure are one-to- one mapping between plaintext and cipher text, which is necessary for a cipher to be decryptable. Secondly, Feistel ciphers have been publicly cryptanalysed for more than two decades, and no systematic weakness has been uncovered. And finally, Feistel ciphers are reasonably fast and simple to implement in software. Speed and simplicity were two important design aims for ICE.

4.2. ALGORITHM FOR CREATING TRANSFORMATION TABLE

1: Load Image
2: Input key
3: Randomize ()
3.1: HorizontalNoBlocks = RandomNum between (LowerHorizontalNoBlocks and ImageWidth)
3.2: VerticalNoBlocks = RandomNum between (LowerVerticalNoBlocks and ImageHeight)
4: NoBlocks = HorizontalNoBlocks * VerticalNoBlocks
END CREATE_TRANSFORMATION_TABLE
Input: Original Image
Output: Transformation table

4.3. ALGORITHM FOR PERFORMING TRANSFORMATION

1: For I = 0 to NoBlocks -1
1.1: Get the new location of block from the transformation table
1.2: Do XOR operation between a key and a value from the new location
1.3: Set block I in its new location
END PERFORM_TRANSFORMATION
Input: Original Image, a string key and Transformation table
Output: Transformed Image.

- For ex, if the input pixel value is FF, then from the transformation table the value corresponding to the FF can be replaced by the stored value in the table.
- After transforming, the transformed image can be encrypted by doing XOR operation with the key.

CHAPTER 5
ENCRYPTION & DECRYPTION ALGORITHM

5.1. Proposed Algorithm For Encryption

The proposed encryption algorithm is as follows.

Step 1 : Choose P, key 1 and key 2 and set j=0.

Step 2 : Choose the initial point x (0) and generate the chaotic sequence x (0), x (1), x(2), … , x (MN/16-1) using eq.(1) and then create b (0), b (1), b (2), ... , b (2MN-1) from x(O), x (1), x (2),……, x(MN/16 - 1) by the generating scheme such that b(32i+0), b(32i+ 1). . .b(32i+ 29) b(32i+ 30) b(32i+ 31) . . . is the binary representation of x (i) for i = 0, 1, 2, … (MN/16-1).

Step 3: For x=0 to M-1 ;
For y = 0 to N- 1;

obtain $q_x(x,y)$, $q_y(x,y)$ for chosen p and given f(x, y) from the solution of the BB equation,

Switch (2xb(j) + b (j+1))

Case 3: q_{xx} (x,y) = mod ((q$_x$ (x,y) + key1), 2n-1)
q_{xx} (x,y) = q_{xx} (x,y) XOR key1
q_{yy}(x,y) = mod ((q(x,y) + key1), 2n-1)
q_{yy} (x, y) = q_{yy} (x,y)XOR key1

Case 2: q_{xx}(x,y) = mod((q$_x$ (x,y) +key1), 2n-1)
q_{xx}(x,y) = q_{xx}(X,y) XNOR key1
q_{yy}(x,y) = mod ((q(x,y) +key1), 2n-1)
q_{yy}(x,y) = q_{yy} (x,y) XNOR key1

Case 1: $q_{ek}(x,y) = \text{mod}((q_k (x,y) + key2), 2n-l)$

$q_{ek}(x,y) = q_{ek}(x,y)$ XOR key2

$q_{ek}(x,y) = \text{mod}((q(x,y) + key2), 2n-1)$

$q_{ek}(x,y) = q_{ek}(x, y)$ XOR key2

Case 0: $q_{ek}(x, y) = \text{mod}\,((q_k (x, y) +key2), 2n-1)$

$q_{ek}(x,y) = q_{ek}(x,y)$ XNOR key2

$q_{ek}(x, y) = \text{mod}\,((q(x, y) +key2), 2n-l)$

$q_{ek}(x, y) = q_{ek}(X,y)$ XNOR key2

j = j + 2;

End;

End;

Step 4: The result $q_{ek}(x,y)$,$q_{ek}(x,y)$ is obtained and stop the algorithm.

The proposed VLSI architectures have two key modules, one for the generation of chaotic bits (CB), and the other for encryption or decryption. The architecture of the chaotic bits (CB) generator is 32. The concept of parallel processing is adopted so that the encryption or decryption of 16 data values can be performed at the same time. Fig.5.1 shows the hardware architecture of the encryption unit (EU). This architecture consists of one 32 bit parallel-in parallel-out register, and 16 encryption processing elements (EPEs). The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input.

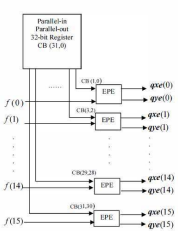

Fig.5.1. Architecture of the encryption unit

The cascade architecture of the encryption processing element (EPE) is shown in Fig.5.2. The architecture of EPE 1 is shown in Fig.5.3. It consists of three multipliers, one adder, two Mod operations and one comparator. The architecture of EPE2 is shown in Fig.5.4. It consists of four data multiplexers, two adders, two XOR gates, two MOD operations, and two inverters, four parallel-to-serial converters, and two serial-to parallel converters.

$q_k(x,y)$= mod(q_k (x,y) - key2),2^{n-1})

f(x, y) = q_k (x,y))$^{-1}$ (q_k(x,y) - 1) mod(p).

j= j+ 2;

End;

End;

Step 4: The result f is obtained and stop the algorithm.

The hardware architecture of decryption unit (DU) is similar to the structure shown in Fig.5.1 except that the EPEs are replaced by decryption processing elements (DPEs) with the encrypted data as the input. The cascade architecture of the decryption processing element (DPE) is shown in Fig.5.5

The architecture of DPE1 is shown in Fig.5.6. The architecture of DPE2 is shown in Fig.5.7. It consists of one subtractor, one Modulo inverse operation, one multiplier and one Mod operation.

5.2.1. ARCHITECTURE OF DECRYPTION PROCESSING UNIT

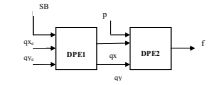

Fig.5.5: Cascade architecture of DPE

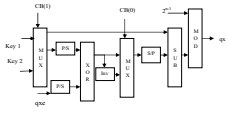

Fig.5.6: Architecture of DPE1

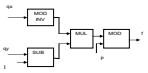

Fig.5.7: Architecture of DPE2

5.1.1. ARCHITECTURE OF ENCRYPTION PROCESSING UNIT

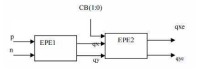

Fig.5.2. Cascade architecture of EPE

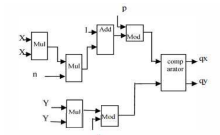

Fig.5.3. Architecture of EPE1

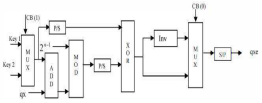

Fig.5.4. Architecture of EPE2

5.2. Proposed Algorithm For Decryption

Steps 1 and 2 are the same as in the above encryption algorithm. Steps 3 and 4 for the decryption are as follows.

Step 3: For x=0 to M-1

For y = 0 to N-1

Switch (2*b(j)) + b(j+1))

Case 3 : q_k (x,y) = q_{ek} (x,y)XOR key1

q_k (x,y) = mod((q_k (x,y) -key1),2^{n-1})

q_k (x,y) = q_{ek} (x,y)XOR key1

q_k(x,y) = mod((q_k(x,y)-key1),2^{n-1})

f(x,y) = (q_k (i))$^{-1}$(q_k(i)-1)mod(p).

Case 2 : q_k (x,y) = q_{ek} (x,y) XNOR key1

q_k (x,y) = mod((q_k (x,y) -key1),2^{n-1})

q_k(x,y)= q_{ek}(x ,y) XNOR key1

q_k(x,y)= mod((q_k(x,y)-key1),2^{n-1})

ff x,y) = (q_k(i))$^{-1}$(q_k(i)-1)mod(p).

Case 1: q_k (x,y) = q_{ek} (x,y)XOR key2

q_k (x,y) = mod((q_k (x,y) - key2),2^{n-1})

q_k(x,y)= q_{ek} (x,y) XOR key2

q_k(x,y)= mod((q_k (x,y) - key2),2^{n-1})

ff(x, y) = (q_k (i))$^{-1}$ (q_k (i) - 1) mod(p).

Case 0: q_k (x,y) = q_{ek} (x,y)XNOR key2

q_k (x,y) = mod((q_k (x,y) - key2),2^{n-1})

q_k(x,y)= q_{ek} (x,y) XNOR key2

CHAPTER 6

PROPOSED CRYPTOSYSTEM

It describes a cryptosystem based on the use of combined chaos and BB equation in addition to the random sequence generator for providing high security. The block diagram of the proposed cryptosystem for encryption and decryption is shown in fig.6.1. In this cryptosystem, for a given primary key p , the root pair of the BB equation corresponding to each pixel of the image is found. Then, according to a binary sequence generated from a chaotic system, a mod operation is performed on the root pair of the BB equation corresponding to each pixel and then each root is XOR-ed or XNOR-ed bit-by-bit to one of the predetermined keys. Then the encrypted image again processed by pseudorandom sequence generator to provide high encryption. The key is very sensitive to initial conditions.

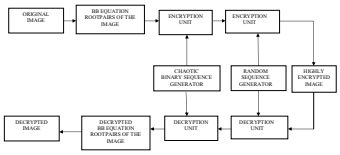

Fig.6.1: Proposed Secure Image Cryptosystem

6.1. GENERATION OF PSEUDO RANDOM SEQUENCE GENERATOR

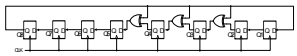

Fig.6.2: 8 bit LFSR structure

The pseudo random sequences are generated by Linear Feedback Shift Register. LFSR is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. And also LFSR is extremely sensitive to initial conditions hence finding the seed is very difficult in this case.

The Fig.8.2 shows an 8-bit LFSR structure used to generate a polynomial of $x^8 + x^6 + x^5 + x^4 + 1$ for 8 – bits. Hence it can generate 256 combinations for a bit single input.

6.1.1. Uses in cryptography

LFSRs have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very uniformly distributed output streams. However, an LFSR is a linear system, leading to fairly easy cryptanalysis. For example, given a stretch of known plaintext and corresponding ciphertext, an attacker can intercept and recover a stretch of LFSR output stream used in the system described, and from that stretch of the output stream can construct an LFSR of minimal size that simulates the intended receiver by using the Berlekamp-Massey algorithm. This LFSR can then be fed the intercepted stretch of output stream to recover the remaining plaintext.

CHAPTER 7

RESULTS & DISCUSSION

The simulation of this project has been done using MODELSIM 7 R2008b and XILINX ISE 9.1i.

Modelsim is a simulation tool for programming {VLSI} {ASIC}s, {FPGA}s, {CPLD}s, and {SoC}s. Modelsim provides a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support is provided for multiple languages including Verilog, SystemVerilog, VHDL and SystemC. The Modelsim conceptual overview is shown below.

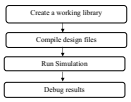

Fig.7.1: Conceptual Overview of Modelsim

In ModelSim, all designs, be they VHDL, Verilog, or some combination thereof, are compiled into a library. We can start a new simulation in ModelSim by creating a working library called "work". "Work" is the library name used by the compiler as the default destination for compiled design units. After creating the working library, we compile our design units into it.

The ModelSim library format is compatible across all supported platforms. We can simulate our design on any platform without having to recompile your design. With the design compiled, invoke the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the simulation time is set to zero, and enter a run command to begin simulation. If the results are not as expected, use ModelSim's robust debugging environment to track down the cause of the problem.

7.1.2. Simulation Result of Proposed Encryption Algorithm

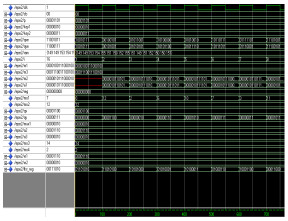

Fig.7.3: Simulation Result of Proposed Encryption Algorithm

In the above figure eb is the combinational register used to select the Encryption Processing Element in the encryption unit, n is the plain text and p is the secret key, key 1 and key 2 are the additional keys used in the encryption process. q_{ek} and q_{ek} are the encrypted output values. All the keys used here are 8 bits. LFSR is the feedback shift register used to provide more encryption.

7.1.3. Simulation Result of Proposed Decryption Algorithm

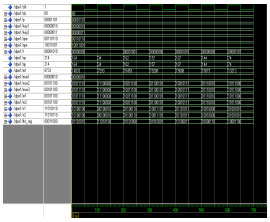

Figure.7.4. Simulation Result of Proposed Decryption Algorithm

In the above figure Encrypted image pixel values can be given as input to the Decryption Unit(DU). Here qxe, qye, primary secret key p, key 1 key 2 are the inputs to the DU. Output pixel values are exactly same as that of input pixel values after decryption process.

Xilinx, Inc. is an American technology company, which designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing The Virtex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores. Xilinx has offered two main FPGA families: the high-performance Virtex series and the high-volume Spartan series. With the introduction of its 28 nm FPGAs in June 2010, Xilinx replaced the high-volume Spartan family with a Kintex family and the low-cost Artix family. The Spartan series targets applications with a low-power footprint, extreme cost sensitivity and high-volume; e.g. displays, set-top boxes, wireless routers and other applications.

The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using Chip Scope Pro tools, and creation of the bit files that are used to configure the chip.

Xilinx is a synthesis tool which converts Schematic/HDL design entry into functionally equivalent logic gates on Xilinx FPGA, with optimized speed & area. So, after specifying behavioral description for HDL, the designer merely has to select the library and specify optimization criteria; and Xilinx synthesis tool determines the net list to meet the specification; which is then converted into bit-file to be loaded onto FPGA-PROM. Also, Xilinx tool generates post-process simulation model after every implementation step, which is used to functionally verify generated net list after processes, like map, place & route.

The synthesis and the simulation results of the proposed and the existing cryptosystems are shown below.

7.1. SIMULATION RESULTS

7.1.1. Simulation Result of Existing IETA Algorithm

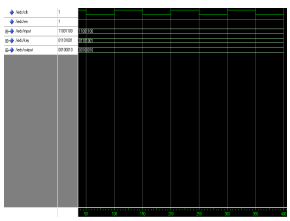

Fig.7.2: Simulation Result of IETA Algorithm

In the above figure input and key are the inputs and output can be produced according to value stored in the transformation table and the value of 8 bit key.

7.2. SYNTHESIS RESULTS

The proposed design has been implemented on Xilinx Spartan 2E device and simulated by Xilinx 9.2i design tool. Therefore the device utilization summary and the timing summary can be obtained from the *Synthesis Report* as shown below.

7.2.1. Power Report

7.2.1.1. Image Encryption using Transformation Algorithm

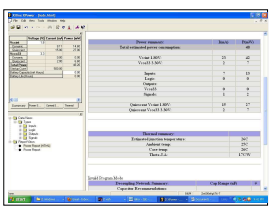

Fig.7.5: Power Report of IETA Algorithm

7.2.1.2. Proposed Cryptosystem based on Chaos & BB Equation

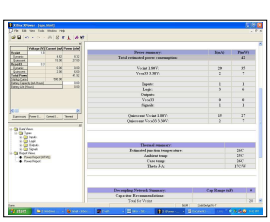

Fig.7.6: Power Report of Proposed Cryptosystem

7.2.2. AREA REPORT

7.2.2.1. Existing IETA Algorithm

Design Summary

Number of errors: 0
Number of warnings: 8

Logic Utilization:
Number of 4 input LUTs: 16 out of 13,824 2%

Logic Distribution:
Number of Slices containing only related logic: 0 out of 0 0%
Number of Slices containing unrelated logic: 0 out of 0 0%

Number of bonded IOBs: 17 out of 510 3%
Number of Block RAMs: 1 out of 72 1%
Number of GCLKs: 1 out of 4 25%
Number of GCLKIOBs: 1 out of 4 25%

Total equivalent gate count for design: 16,384
Additional JTAG gate count for IOBs: 864

7.2.2.2. Proposed Chaos & BB Combined Algorithm

Area report:
Design Summary

Number of errors: 0
Number of warnings: 2

Logic Utilization:

Number of 4 input LUTs: 8 out of 13,824 1%

Logic Distribution:

Number of occupied Slices: 4 out of 6,912 1%
Number of Slices containing only related logic: 4 out of 4 100%
Number of Slices containing unrelated logic: 0 out of 4 0%

Total Number of 4 input LUTs: 8 out of 13,824 1%
Number of bonded IOBs: 34 out of 510 6%

Total equivalent gate count for design: 48
Additional JTAG gate count for IOBs: 1,632

7.2.3. Delay Report

7.2.3.1. Image Encryption Using Transformation Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 7.548ns (Maximum Frequency: 132.481MHz)
Minimum input arrival time before clock: 8.535ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **2.319ns**

7.2.3.2. Proposed Chaos & BB Combined Algorithm

Timing Summary:

Speed Grade: -7

Minimum period: 2.956ns (Maximum Frequency: 338.295MHz)
Minimum input arrival time before clock: 2.948ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Therefore the total delay = **3.268ns**

7.3. COMPARISON

SUMMARY	IETA BASED CRYPTOSYSTEM	PROPOSED CHAOS & BB BASED CRYPTOSYSTEM	% OF RESULTS
POWER	48Mw	42mW	**12.5**
SPEED	2.319ns	3.268ns	**40**
AREA Total Gate Count	16384	48	**99.7**

Table 2.Comparison between IETA based Cryptosystem and proposed Cryptosystem

CHAPTER 8
RESULTS

8.1. IMAGE RESULTS OF EXISTING IETA

INPUT IMAGE

Fig.8.1. lena image

ENCRYPTED IMAGE

Fig.8.2.Encrypted Image

OUTPUT IMAGE

Fig.8.3.Decrypted Image

8.2. IMAGE RESULTS OF PROPOSED CHAOS & BB EQUATION

8.2.1. Results for Grayscale image

INPUT IMAGE

Fig.8.4. lena image

ENCRYPTED IMAGE

Fig.8.5.Encrypted Image

OUTPUT IMAGE

Fig.8.6.Decrypted Image

8.2.2. Results for Color image

INPUT IMAGE

Fig.8.7. lena image

ENCRYPTED IMAGE

Fig.8.8.Encrypted Image

OUTPUT IMAGE

Fig.8.9.Decrypted Image

8.2.3. Results for fingerprint image

INPUT IMAGE

Fig.8.10. fp image

ENCRYPTED IMAGE

Fig.8.11.Encrypted Image

OUTPUT IMAGE

Fig.8.12..Decrypted Image

CHAPTER 9
CONCLUSION & FUTURE WORK

In this project, we proposed a robust and efficient cryptosystem to transmit a digital image in a secure way. A secure cryptosystem based on Chaos & BB equation with pseudorandom sequence is proposed for image encryption & decryption. The new quadratic equation based encryption & decryption algorithm alters the positions of the pixels by XOR and Mod operations without changing its correlation with neighbouring pixels. This method is highly sensitive to initial conditions. The VLSI architecture of the proposed cryptosystem is designed and realized using Xilinx ISE VLSI software for an image. The proposed algorithm is highly promising for high security in real time applications. The proposed algorithm also compared with existing IETA algorithm.

It should be proved that this system is secure against all kinds of attacks. Cryptanalysis of the proposed cryptosystem specifically for cipher text attacks and known plain text attacks is to be provided.

REFERENCES :

1. K.Deergha Rao and Ch. Gangadhar, "VLSI realization of a secure cryptosystem for image encryption and decryption" in Proceedings of IEEE Int. Conf. on Communication and signal processing, 2011
2. K.Deergha Rao, K.Praveen Kumar,and P. V.Muralikrishna,"A New and Secure Cryptosystem for Image Encryption and decryption", appear in IETE Journal of Research, March-Apr. 2011
3. Mohammad Ali Bani Younes and Aman Jantan, "Image Encryption Using Block-Based Transformation Algorithm" Proc. IAENG International Journal of Computer Science, Feb 2008, 35:1, IJCS_35_1_03
4. G.Alvarez L.H.Encinas, and M.Masque, "Known-Plaintext Attack to Two Cryptosystems Based on the BB Equation", IEEE Transactions on Circuits and Systems II: Express Briefs Volume 55, Issue 5, May 2008
5. R.Rhouma, S.Meherzi, S.Belghith, "CML-based colour image encryption" in International Journal of Chaos, Solitons and Fractales 2007
6. Alvarez, G. and Shujun L. "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystem" in Int. Journal of Bifurcation and Chaos, 2006
7. N.Rama Murthy, M.N.S.Swamy, "Cryptographic Applications of Brahmagupta-Bhaskara Equation", IEEE Transactions on circuits -I, 2006
8. G.Zhi-Hong, H.Fangjun, and G.Wenjie,"Chaos - based image encryption algorithm,"Department of Electrical and computer Engineering,University of Waterloo, ON N2L 3G1, Canada.Published by: Elsevier, 2005, pp. 153-157.

9. I.Ozturk, I.Sogukpinar, "Analysis and comparison of image encryption algorithm,"Journal of transactions on engineering, computingand technology December, vol. 3, 2004, p.38.
10. B.Furht and D.Socek, "Multimedia security encryption techniques" IEC Int. Engg. Consortium, Chicago, IL pages 335-349, 2004
11. Sinha, K. Singh, "A technique for image encryption using digi tal signature," Source:OpticsCommunications,vol.218,no..2003
12. G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalyzing a discrete-time chaos synchronization secure communication system,"Chaos, Solitons &fractals, 2003,pp. 689-694.
13. S.l.Li and X.Zheng , "On the secusity of an image encryption method," Proc. IEEE International Conference on Image Processing (ICIP 2002), vol.2, pp. 925-928, 2002.
14. S.J.Li and X. Zheng, "Cryptanalysis of a Chaotic Image Encryption Method", IEEE International Symposium on Circuits and Systems (ISCAS 2002)
15. M.I.Sobhy, and A.R.Shehata, "Methods of attacking chaotic encryption and countermeasures," Proc. IEEE Inernational Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)
16. H.Cheng and X.Li, "Partial encryption of compressed images and videos" in IEEE Transactions on Signal Processing, volume 48, pages 2439-2451, 2000
17. Dachselt F, Schwarz W (2001) Chaos and cryptography. IEEE Trans Circuits and Systems-I 48(12):1498 – 1509

18. Aloha Sinha, Kehar Singh, "A technique for image encryption using digital signature", Optics Communications, ARTICLE IN PRESS, 2003,1-6,
19. S.S.Maniccam, N.G. Bourbakis, "Lossless image compression and encryption using SCAN", Pattern Recognition 34 (2001), 1229-1245
20. Chin-Chen Chang, Min-Shian Hwang, Chen,"A new encryption algorithm for image cryptosystems", The Journal of Systems and Software 58 (2001), 83-91
21. Jiun-In Guo, Jui-Cheng Yen, "A new mirror-like image encryption algorithm and its VLSI architecture", Department of Electronics Engineering National Lien-Ho College of Technology and Commerce,Miaoli,(2001) Taiwan, Republic of China
22. Jui-Cheng Yen, Jiun-In Guo, "A new chaotic image encryption algorithm",Department of Electronics Engineering National Lien-Ho College of Technology and Commerce, Miaoli, Taiwan,(2001) Republic of China
23. Shuqun Zhang and A. Karim, "Color image encryption using double random phase encoding",Microwave and Optical Technology, Vol.21, No.5, June 5 1999
24. Young-Chang Hou, "Visual cryptography for color images", Pattern Recognition 36 (2003), www.elsevier.com/locate/patcog, 1619-1629
25. M Kwan, ―The Design of the ICE Encryption Algorith, in proc. Of Fast software Encryption workshop, 1997.