IMAGE ENCRYPTION SCHEME BASED ON CHAOTIC MAPS

SU SU MAUNG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INFORMATION ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2006

ISBN 974-15-2290-8

หัวข้อวิทยานิพนธ์	การเข้ารหัสลับภาพด้วยกระสวนอลวน
นักศึกษา	นางสาวซู ซู หม่อง
รหัสประจำตัวนักศึกษา	47061143
ปริญญา	วิศวกรรมศาสตร์มหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. ปิติเขต สู้รักษา

บทคัดย่อ

วิทยานิพนธ์นี้เสนอวิธีเข้ารหัสลับแบบใหม่โดยอาศัยกล่อง S ขนาด 8x8 ร่วมกับพลศาสตร์ของระบบสัญญาณอลวน (Chaos) เพื่อสร้างลำดับไบท์แบบกึ่งสุ่ม จากนั้นจะทำการเรียงสับเปลี่ยนแล้วแปลงสัญญาณให้อยู่ในรูปดิจิตอลของฟังก์ชันอลวนในสองมิติ ลำดับไบท์แบบกึ่งสุ่มนี้สร้างขึ้นโดยอาศัยกระสวนแบบ Cat ซึ่งใช้เป็นดัชนีของกล่อง S โดยให้เอาท์พุตขนาด 8 บิต (0-255) ซึ่งจะนำไปทำ XOR กับสื่อปกติ (plaintext) ให้กลายเป็นสื่อรหัสลับ (ciphertext) และทำ XOR กับสื่อรหัสลับให้กลับเป็นสื่อปกติตามเดิม จากการทดสอบประสิทธิผลของวิธีการที่นำเสนอด้วยวิธีทางสถิติพบว่าวิธีการเข้ารหัสลับที่นำเสนอนี้สามารถสร้างเลขกึ่งสุ่มที่มีขนาดของกุญแจรหัสยาวเพียงพอแก่การใช้งา

Thesis Title	Image Encryption Scheme Based on Chaotic Maps
Student	Miss. Su Su Maung
Student ID.	47061143
Degree	Master of Engineering
Programme	Information Engineering
Year	2006
Thesis Advisor	Assoc. Prof. Dr. Pitikhate Sooraksa

ABSTRACT

A new encryption scheme using dynamical 8 × 8 S-box, based on the composition of chaotic maps, is proposed. In this method, the dynamical S-box, one first uses a known chaotic dynamical system to generate a sequence of pseudo-random bytes, then applies certain permutations to them, using the discretized version of another two-dimensional chaotic map. A sequence of pseudo-random bytes generated from two-dimensional cat map is used to index the entry of the S-box. The output 8 bits (0-255) of the S-box are XOR-ed with the plaintext to obtain the ciphertext and XOR-ed with the ciphertext to obtain the plaintext. Standard statistical tests of this scheme are performed. We show that this new scheme can generate usable pseudo-random numbers, while maintaining a large enough keyspace for potential use in encryptions.

Acknowledgements

First I would like to express my gratitude to my parents for their compassion, patience and encouragement. I would like to express my sincere thanks to my sister and my brother for their good company and care that made me feel a bit more at home even when I was away. I would like to extend my sincere thanks to Assoc. Prof. Dr. Pitikhate Sooraka and Asst. Prof. Kitdakorn Klomkarn for their supervision, encouragement, useful suggestions and warm feedback throughout my research. I would like to express my great thanks to Assoc. Prof. Dr. Jongkol Ngamwiwit for her kind helps. I would like to extend my gratitude to all my teachers and friends from KMITL who are giving uncountable helps to me. Since I have been supported by JICA (Japan International Cooperation Agency), I would be thankful forever and in return, I promise to be a useful one for our people of Burma. Finally, I would like to acknowledge with gratitude and admiration the help of many people who are not named here.

Contents

Page

Thai Abstract.....I

English Abstract.....II

Acknowledgements.....III

Contents.....IV

List of Tables.....VI

List of Figures.....VII

Chapter 1 Introduction 1

1.1 Motivation..... 1

1.2 Literature Reviews on Image Encryption 2

1.3 The Organization of the Thesis..... 4

Chapter 2 Chaos in Dynamical Systems 5

2.1 Chaotic maps..... 5

2.1.1 One-dimensional map..... 6

2.1.2 Two-dimensional map..... 7

2.1.2.1 The two-dimensional Cat map 8

2.1.2.2 The two-dimensional Baker map 10

Chapter 3 Traditional Cryptographic Ciphers 16

3.1 Introduction to Cryptography 16

3.2 Block Ciphers 17

3.2.1 Blowfish Algorithm..... 19

3.2.2 Secure And Fast Encryption Routine (SAFER K-64) 22

3.2.3 Advanced Encryption Standard (AES) 26

3.2.4 Advantageous and Disadvantageous of each Block Cipher..... 30

3.3 Conclusion 32

Chapter 4 Chaotic based Image Encryption 34

4.1 Introduction to Image Encryption 34

4.2 Basic Features of Chaos 35

Contents (cont.)

page

4.3 Relationship between Chaos and Cryptography 36

4.4 Chaos-Based Block Ciphers for Image Encryption 37

4.5 Chaos-Based Stream Ciphers for Image Encryption 39

Chapter 5 Performance of the Proposed Method 42

5.1 The New Scheme of the Proposed Chaotic Image Encryption 42

5.1.1 Designing dynamical S-box based on chaotic maps 42

5.1.2 Encryption Scheme using dynamical S-box 44

5.2 Simulation Results 46

5.3 Standard statistical tests and analysis 48

5.4 Comparison between the Proposed Scheme and the Others 50

Chapter 6 Conclusions and Further Works 55

6.1 Conclusions 55

References 57

List of Tables

Table page

3.1 Number of rounds *Nr* as a function of the block and key length 27

3.2 Shift offsets for different block lengths 29

3.3 Traditional block ciphers..... 32

3.4 Comparisons of block ciphers..... 33

4.1 Similarities and differences between cryptographic algorithms and chaotic..... 37

5.1 An example of 8×8 S-box 44

5.2 The required interval for runs test in the FIPS PUB 140-2 statistical tests..... 48

5.3 The result of statistical test of the proposed scheme..... 49

5.4 Speeds of the encryption algorithms..... 52

List of Figures

Figure page

2.1 The logistic map 6

2.2 Bifurcation diagram of logistic map 7

2.3 The 2D chaotic cat map 8

2.4 One iteration of the cat map for 8 × 8 pixels image 9

2.5 Original image (a), after applying the cat map once (b), twice (c), nine times 10

2.6 The baker map 10

2.7 Generalized baker map 11

2.8 Discretized baker map 12

2.9 The mapping of the rectangle 8 × 2 to a row of 16 pixels 13

2.10 The permutation induced by the discretized baker map for 6 × 6 pixelsimage 13

2.11 The permutation induced by the discretized baker map for 8 × 8 pixels 14

2.12 Original image (a), after applying the baker map once (b), twice (c), nine 14

2.13 Image consisting of black circle on a white background (a), encrypted 15

2.14 Image consisting of black circle on a white background (a), encrypted 15

3.1 Block cipher 18

3.2 Feistel cipher 19

3.3 Blowfish Encryption 20

3.4 *F* function 21

3.5 Encryption structure of SAFERK-64 23

3.6 Encryption round structure of SAFER K-64 24

3.7 Decryption structure of SAFER K-64 25

3.8 Decryption round structure of SAFER K-64 25

3.9 Key schedule for SAFER K-64 26

3.10 State array and plaintext 28

3.11 Algorithm Overview of AES 28

5.1 Designing dynamical S-box 43

5.2 Encryption and decryption schemes: (a) encryption scheme, (b) decryption 45

5.3 Plain image of Lena (a), cipher image of Lena (b) 45

5.4 Plain image of Shwedagone Pagoda (a), cipher image of Shwedagone 46

List of Figures (cont.)

Figure ..... page

5.5 Histogram of the original image (a), histogram of the encrypted image (b)..... 47

5.6 The 2 D spectrum of the original image (a), the 2 D spectrum of the encrypted..... 47

5.7 Plain image (a) and cipher image (b) of Myanmar national flag..... 50

5.8 Histogram of the original image (a), histogram of the cipher image (b)..... 50

5.9 Encryption using AES: (a) cipher image, (b) histogram of the cipher image 51

5.10 Encryption using Blowfish: (a) cipher image, (b) histogram of the cipher..... 51

5.11 Encryption using SAFER-K64: (a) cipher image, (b) histogram of the cipher 51

5.12 Encryption speed test 52

5.13 Encryption speed tests using variable plaintext sizes 53

5.14 Encryption speed curves 54

Chapter 1

Introduction

1.1 Motivation

In the recent years, internet multimedia applications have become very popular. With the proliferation of the Internet and maturation of the digital signal processing technology, applications of digital imaging are prevalent and still continuously and rapidly increasing today. Many digital services, such as confidential video conferencing, medical and military imaging systems, require reliable security in storage and transmission of digital images/videos. As the rapid progress of the internet in the digital world today, the security of digital images/videos has become more and more important. Valuable multimedia content such as digital images/videos, however, is vulnerable to unauthorized access. In this regard, a direct solution is to use encryption algorithm to mask the image data streams, which has led to the celebrated number-theory-based encryption algorithms such as DES (Data Encryption Standard), AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), RSA (developed by Rivest, Shamir and Adleman). However, these encryption schemes appear not to be ideal for image applications, due to some intrinsic features of images such as bulk capacity, high redundancy, and high correlation among pixels which are troublesome for traditional encryption. Moreover, these encryption schemes require extra operations on compressed image data, thereby demanding long computational time and high computing power. In real-time communications, due to their low encryption and decryption speeds, they may introduce significant latency. Hence, in the thesis a new image encryption scheme based on chaotic maps is proposed. The idea of using chaos for data encryption is certainly not new and can be traced to the classical Shannon's paper [3]. Chaotic maps have been utilized in several different ways in cryptography. Chaotic system is sensitive to the initial condition of the system so that initially nearby points can evolve quickly into very different states. Chaotic systems exhibit irregular and unpredictable behavior. Due to the tight relationship between chaos and cryptography, chaotic cryptography has also been extended to design image encryption schemes. Probably the most obvious application of chaotic maps is to

use one or more one dimensional maps as pseudo-random number generators producing a binary stream which is then XOR-ed with the plaintext to produce the ciphertext. However, these schemes have been shown to produce weak ciphers. In this thesis, a method using cryptographically strong 8×8 S-box based on chaotic maps is proposed. The substitution boxes(S-boxes) have been widely used in almost all traditional cryptographic system, such as DES, AES. RC4 which is a variable-key-size stream cipher also use a 8×8 S-box [2]. The main obstruction in designing image encryption algorithm is that it is rather difficult to shuffle and diffuse data by traditional means of cryptology. Some intrinsic features of images such as bulk data capacity and high redundancy are troublesome for traditional encryption. Since the encryption speed of some traditional ciphers is not sufficiently fast, especially for software implementations, it is difficult to achieve fast and secure real-time encryption simultaneously for large-sized bulky data. Chaos based image encryption has superiority over the conventional encryption methods, particularly in a good combination of speed, security, and flexibility.

1.2 Literature Reviews on Image Encryption

There are some image encryption methods 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 also meant for understanding the image encryption technology in general. Classified with respect to the approach in constructing the scheme, image encryption algorithms 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 [10], 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.

There are some other types of chaotic cryptosystems, most of which transform plaintext directly. And they are often classified into two types: chaotic stream cryptosystems and chaotic block cryptosystems. In chaotic stream cryptosystems, a key stream is produced by a chaotic map, which is used to encrypt a plaintext bit by bit [20, 21]. A chaotic block cryptosystem, on the other hand, transforms a plaintext block by block with some chaotic maps. A cryptosystem based on the chaotic gradient tent map was constructed in [22], and the one based on the modified baker map was suggested in [11]. These cryptosystems apply chaotic maps repeatedly, which guarantees the randomness of the encrypted data. The encryption scheme using dynamical S-box is proposed in [1, 12]. In these schemes, chaotic maps are used to generate cryptographically secure dynamical 8×8 S-box.

In order to speed up the encryption process 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 [3], but their purposes are different. In [16], several partial encryption schemes were provided. It was reported that by a partial encryption, only 13-27% 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. Another fast encryption scheme was proposed in [19], which encrypts the sign bits of the DCT coefficients (i.e., the sign bits of differential DC values for the DC coefficients). Because DC values significantly affect the quality of an image, changing them will render the whole image unreadable.

Since wavelet-based image compression achieves both high compression rate with reasonably high image quality and low computational complexity, many image compression standards (for moving or still pictures) have selected to use wavelets. Integrating an encryption algorithm with wavelet image coding is reasonable and has

great potential usage. In [18], a wavelet-based system combining compression and encryption was recommended.

1.3 The Organization of the Thesis

The organization of this thesis is as follows.

In the chapter 2, 3 and 4, we introduce the theoretical backgrounds, the concept of cryptography, and the relationship between chaos and cryptography. The chapter 5 will describe our proposed new scheme of chaotic image encryption involving the image encryption algorithm, simulation results and analysis of security and performance. Finally, conclusions and future works are remarked in the chapter 6.

Chapter 2

Chaos in Dynamical Systems

2.1 Chaotic maps

Chaos is a dynamical system that is extremely sensitive to its initial conditions. It is a deterministic nonlinear system that has random-like behaviors. The property that stability may depend on initial conditions is characteristic only for nonlinear systems. The study of nonlinear behavior is called nonlinear dynamics. Chaos theory has become a new branch of scientific studies today. Discrete chaotic dynamic systems (i.e., maps) are used in cryptography. This sensitivity property is commonly applied to cryptosystems. If a parameter that describes a linear system is changed, then the quantitative behavior of the system will change, but the qualitative nature of the behavior remains the same. For nonlinear systems, a small change in a parameter can lead to sudden and dramatic changes in both the qualitative and quantitative behavior of the system. For one value the behavior might be periodic, for another value only slightly different from the first, the behavior might be completely aperiodic. Some sudden and dramatic changes in nonlinear systems may give rise to the complex behavior called chaos. The behavior is aperiodic and is apparently random or noisy. All chaotic systems are nonlinear, but not all nonlinear systems are chaotic. In the case of discrete, integer-valued time (with n denoting the time variable, $n = 0,1,2,\ ...$), an example of a dynamical system is a map, which we write in vector form as

$$\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n). \tag{2-1}$$

where $\mathbf{x}_n$ is N-dimensional, $\mathbf{x}_n = (x_n^{(1)}, x_n^{(2)}, ..., x_n^{(N)})$. Given an initial value $\mathbf{x}_0$, we obtain the value at time $n = 1$ by $\mathbf{x}_1 = \mathbf{f}(\mathbf{x}_0)$. Having determined $\mathbf{x}_1$, we can then determine the value at $n = 2$ by $\mathbf{x}_2 = \mathbf{f}(\mathbf{x}_1)$, and so on. Thus given an initial condition $\mathbf{x}_0$, we generate an orbit (or trajectory) of the discrete time system: $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, ...$ If there are two possible values of $\mathbf{x}_n$ for a given $\mathbf{x}_{n+1}$, the map is not invertible. If the map is non invertible, chaos is possible even in one-dimensional maps. If the map is invertible, the dimensionality requirement on map is at least two. Then there can be no chaos unless

$$N \geq 2$$

2.1.1 One-dimensional map

One-dimensional noninvertible maps are the simplest systems capable of chaotic motions. Here we consider the logistic map,

$$x_{n+1} = \mu x_n (1 - x_n). \tag{2-2}$$

The map is illustrated in Figure 2.1. For each initial condition, the map is iterated over and over again. This map receives a real number between 0 and 1, then returns a real number in [0, 1] again. The function is a map of interval to itself. For $\mu = 4$ the logistic map has a maximum value of 1 for $x = 1/2$. The various sequences are generated depending on the parameter μ and the initial value x_0.

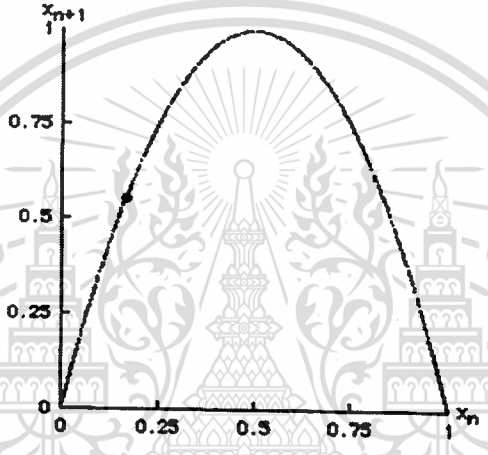
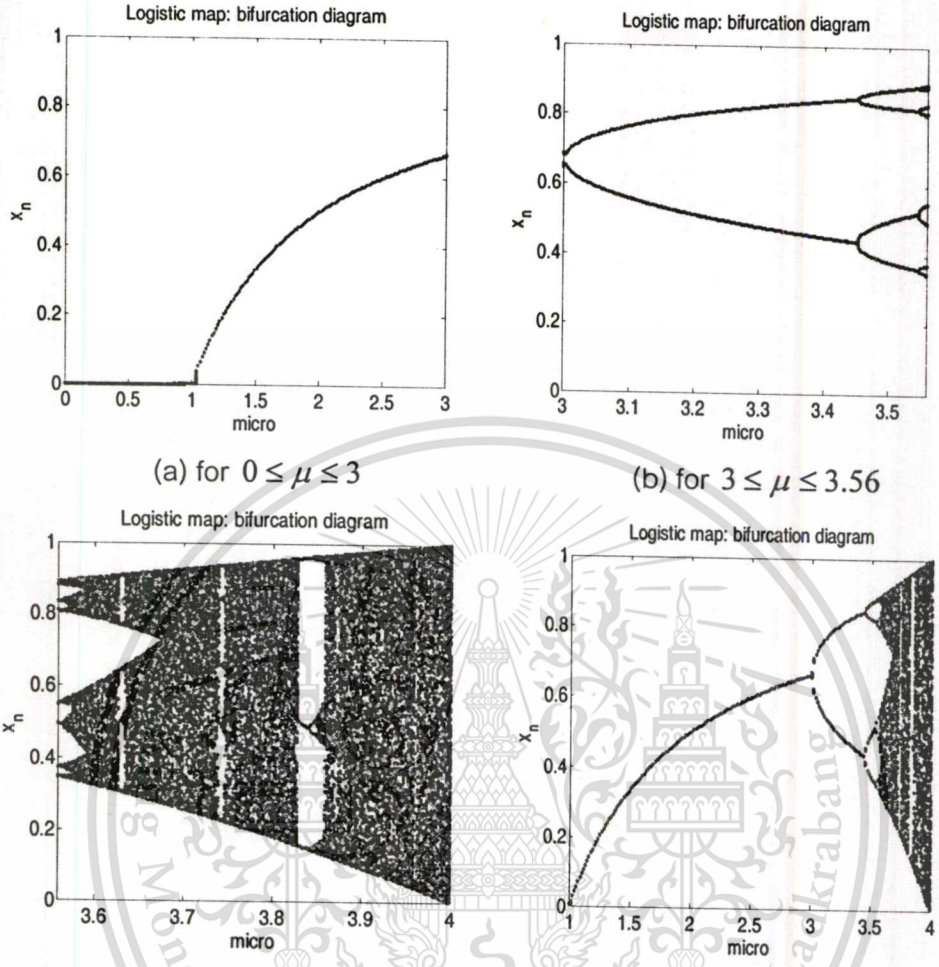


Figure.2.1 The logistic map

When the parameter μ is between 0 and 3 the sequence x_n converges to a fixed point. When the parameter μ exceeds 3, the sequence converges to periodic orbit of period 2. Further increases in μ lead to periodic 4, periodic 8, and so on, occurring at ever smaller and smaller increments of μ. The periods of the periodic orbits are doubled. This is called period doubling cascade, and beyond this cascade, the stable periodic orbit disappears and Chaos appears. For μ just greater than 3.5699..., the sequence values never seem to repeat. The behavior is chaotic. This transition of the orbit structure with the change of parameter is called bifurcation phenomena. The logistic map is very simple mathematical system which involves no derivatives, integrals. But this function exhibits the universal features of the behavior, such as the period-doubling leading to chaos. The trajectories of the logistic map which converge to the fixed point are illustrated by Figure 2.2(a). The phenomenon of period

doubling bifurcation is shown in Figure 2.2(b) and the chaotic behavior occurs as shown in Figure 2.2(c).



(a) for $0 \le \mu \le 3$ (b) for $3 \le \mu \le 3.56$

(c) for $3.56 \le \mu \le 4$ (d) for $1 \le \mu \le 4$

Figure.2.2 Bifurcation diagram of logistic map

2.1.2 Two-dimensional map

The simplest possible case of a multi-dimensional map is a two-dimensional map. The invertible two-dimensional chaotic maps are iterated on a torus or on a square to create new symmetric block encryption schemes. Permutation is iterated and iteration time is a part of secret key. A chaotic map is first generalized by introducing parameters and then discretized to a finite square lattice of points which represent pixels or some other data items. Although the discretized map is a permutation and thus cannot be chaotic, it shares certain properties with its continuous counterpart as long as the number of iterations remains small. The discretized map is further extended to three dimensions and composed with a simple diffusion mechanism. As a result, a symmetric block product encryption scheme is obtained. To encrypt an $N \times N$ image, the

ciphering map is iteratively applied to the image. Examples of some other two-dimensional chaotic maps are given and their suitability for secure encryption is discussed. It is shown that the permutations induced by the invertible two-dimensional chaotic map behave as typical random permutations. Computer simulations indicate that the cipher has good diffusion properties with respect to the plain-text and the key.

2.1.2.1 The two-dimensional Cat map

The cat map is a very well-studied two-dimensional invertible chaotic map introduced by Arnold and Avez. The mathematical formula is:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 . \qquad (2\text{-}3)$$

$$= (x_n + y_n \bmod 1,\ x_n + 2y_n \bmod 1)$$

Where x (mod 1) means the fractional parts of a real number x by subtraction or adding an appropriate integer. The map is known to be chaotic. The unit square is first stretched by the linear transform. After applying the mod operator, the pieces of the image lying in squares other than the unit square are cut and shifted back to the unit square (see Figure 2.3).

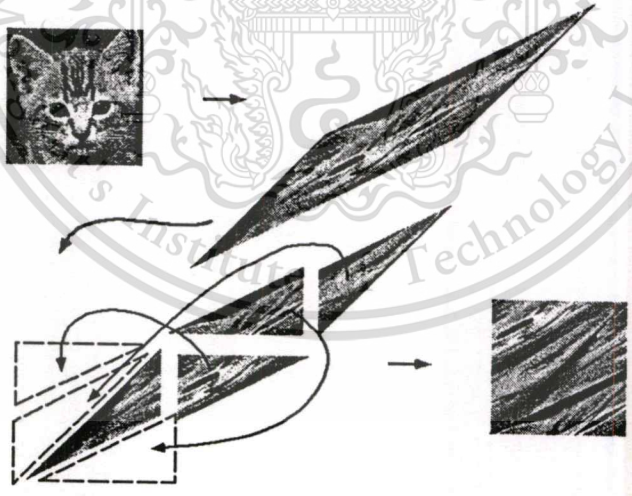


Figure.2.3 The 2D chaotic cat map

The matrix A is used as the parameters for the generalized version of the cat map. A general matrix A,

$$A = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$$

with integer elements will be denoted $A_{(t,u,v,w)}$. In particular, to make sure that the map is one-to-one, the determinant of *A*, $|A| = tw - uv$, has to be equal to 1. We note that the four tuple (t,u,v,w) produces the same cipher as the four tuple (*t* mod *N*, *u* mod *N*, *v* mod *N*, *w* mod *N*).

The discretized version of the cat map is obtained simply by changing the range of (*x*, *y*) from the unit square $I \times I$ to the discrete lattice $N_0^N \times N_0^N$

$$A_{(t,u,v,w)}(r,s)^T \bmod N . \qquad (2\text{-}4)$$

The map $A_{(t,u,v,w)}$ transforms the square lattice of points $N_0^N \times N_0^N$ onto itself in a one-to-one manner. The results of applying the discretized cat map with the matrix $A(1,1,1,2)$ to the 8×8 pixels image is shown in Figure 2.4 and to the 256×256 test image once, twice and nine times are shown in Figure 2.5(b), (c) and (d).

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

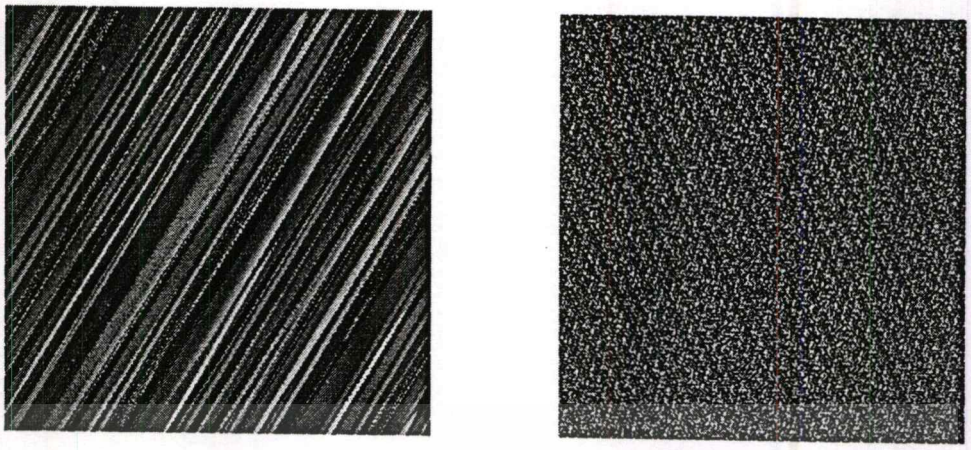
→

1	11	21	31	33	43	53	63
10	20	30	40	42	52	62	8
19	29	39	41	51	61	7	9
28	38	48	50	60	6	16	18
37	47	49	59	5	15	17	27
46	56	58	4	14	24	26	36
55	57	3	13	23	25	35	45
64	2	12	22	32	34	44	54

Figure.2.4 One iteration of the cat map for 8 × 8 pixels image

(a)

(b)



(c) (d)

Figure.2.5 Original image (a), after applying the cat map once (b), twice (c), nine times (d)

2.1.2.2 The two-dimensional Baker map

The baker map, B, is described with the following formulas:

$$B(x,y) = (2x, y/2) \quad \text{when } 0 \le x < 1/2, \tag{2-5}$$

$$B(x,y) = (2x-1, y/2+1/2) \quad \text{when } 1/2 \le x \le 1$$

The map acts on the unit square as shown in Figure 2.6. The left vertical column $[0,1/2)\times[0,1)$ is stretched horizontally and contracted vertically into the rectangle $[0,1)\times[0,1/2)$ and the right vertical column $[1/2,1)\times[0,1)$ is similarly mapped into $[0,1)\times[1/2,1)$. The baker map is a chaotic bijection of the unit square $I \times I$ onto itself.

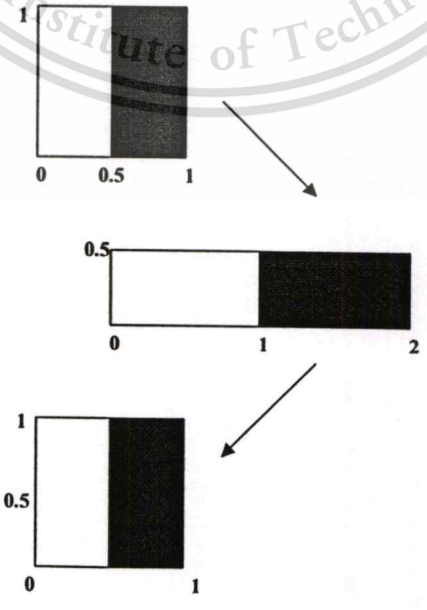


Figure.2.6 The baker map

The map can be generalized in the following way [Pichler & Scharinger, 1994, 1995]. Instead of dividing the square into two rectangles of the same size, the square is divided into *k* vertical rectangles $[F_{i-1}, F_i) \times [0,1), i = 1,...,k$, $F_i = p_1 + ... + p_i$, $F_0 = 0$ such that $p_1 + ... + p_k = 1$ as shown in Figure 2.7. The lower right corner of the *i*th rectangle is located at $F_i = p_1 + ... + p_i$. The generalized baker map stretches each rectangle horizontally by the factor of $1/p_i$. At the same time, the rectangle is contracted vertically by the factor of p_i. Finally, all rectangles are stacked on top of each other as in Figure 2.7. Formally,

$$B(x,y) = \left(\frac{1}{p}(x - F_i), p_i y + F_i \right) \qquad (2\text{-}6)$$

for

$$(x,y) \in [F_i, F_i + p_i) \times [0,1),$$

It is convenient to denote the baker map and its generalized version as $B_{(1/2,1/2)}$ and $B_{(p_1,...,p_k)}$, respectively. The generalized map inherits all important properties of the baker map such as sensitivity to initial conditions and parameters, mixing, and objectiveness.

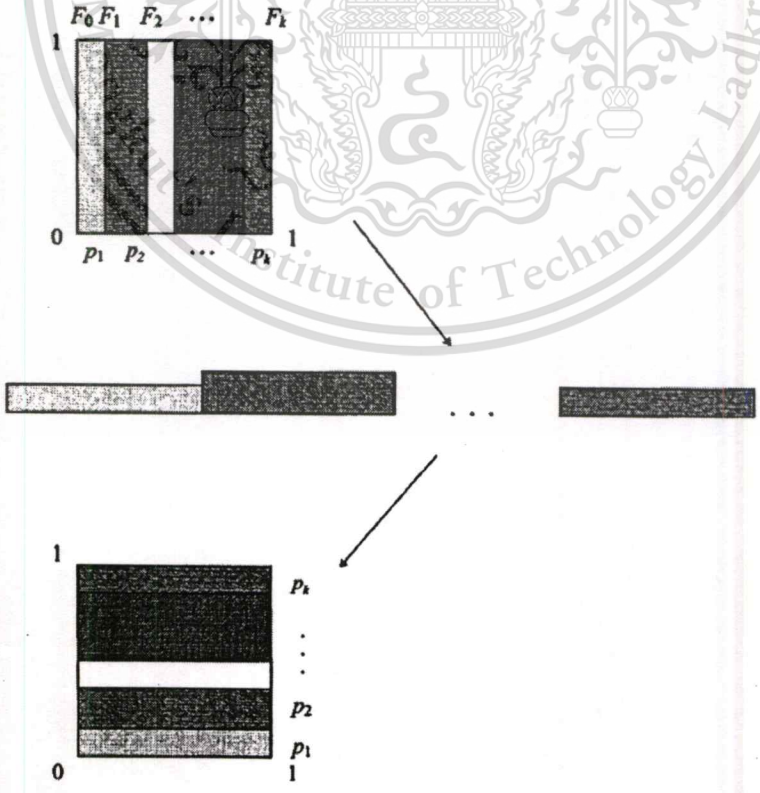


Figure.2.7 Generalized baker map

Since an image is defined on a lattice of finitely many points (pixels), a correspondingly discredited form of the basic map needs to be derived. In particular, the discredited map is required to assign a pixel to another pixel in a bijective manner. Since the discredited map is desired to inherit the property of the continuous basic map, the discredited map should become increasingly close to the basic map as the number of pixels tends to infinity. The discredited generalized baker map will be denoted $B_{(n_1,\ldots,n_k)}$, where the sequence of k integers, $n_1,\ldots,n_k$, is chosen such that each integer n_i, divides *N*, and $n_1+\ldots+n_k=N$. Denoting $N_i=n_1+\ldots+n_i$, the pixel (*r*, *s*), with $N_i \le r < N_i+n_i$, and $0 \le s < N$ is mapped to

$$B_{(n_1,\ldots,n_k)}(r,s)=\left(\frac{N}{n_i}(r-N_i)+s \bmod \frac{N}{n_i},\frac{n_i}{N}(s-s \bmod \frac{N}{n_i})+N_i\right). \quad (2\text{-}7)$$

This formula is based on the following geometrical considerations. A $N \times N$ square is divided into vertical rectangles of height *N* and width n_i. Following the action of the generalized baker map, these vertical rectangles should be stretched in the horizontal direction and contracted in the vertical direction to obtain a horizontal $n_i \times N$ rectangle. To achieve this for the discretized map, each vertical rectangle $N \times n_i$ is divided into n_i boxes $N/n_i \times n_i$ containing exactly *N* points (see Figure 2.8). Each of these boxes is mapped to a row of pixels. Since there are n_i boxes, a horizontal rectangle $n_i \times N$ is obtained, as required.

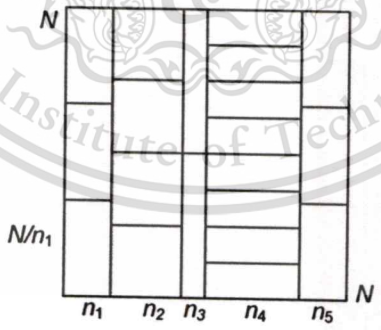


Figure.2.8 Discretized baker map

Now, how the pixels in each box are mapped to a row of pixels need to be specified. Since the original baker map is continuous on each box, the only plausible discretization is to map the box column by column. An example for N = 16, n_i = 2 is shown below. The rectangle $N/n_i \times n_i = 16/2 \times 2 = 8 \times 2$ is mapped to a row of 16 pixels as follows:

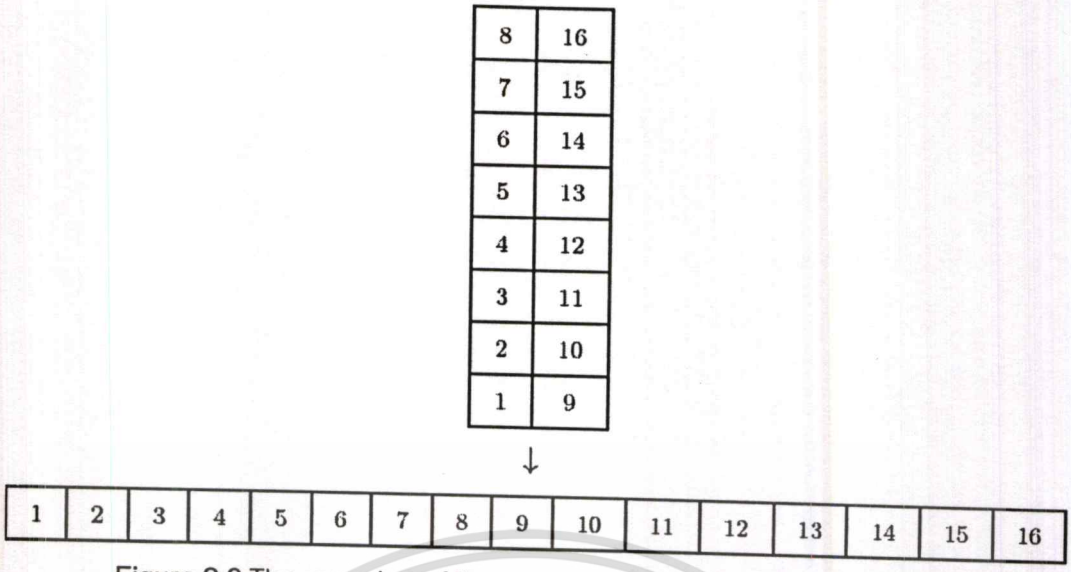


Figure.2.9 The mapping of the rectangle 8 × 2 to a row of 16 pixels

Examples of permutations for a complete 6×6 pixels image and an 8×8 pixels image are worked out in detail in Figure 2.10 and 2.11. For the 6×6 pixels image, a 3-1-2 division is used, while in the 8×8 pixels image case the division is 2-4-2. Equation (2-7) is a symbolic, mathematical description of this geometric procedure. The application of the baker map to a 256×256 test image shown in Figure 2.12(a) produces encrypted images as demonstrated with Figure 2.12(b), (c) and (d). The ciphering key was randomly generated and consists of the following sequence of 6 divisors of 256: (8 8 59 59 4 118). Figure 2.12(a) shows the original image, and Figure 2.12(b), (c), and (d) show the results of applying the generalized discretized baker map once, twice and nine times respectively.

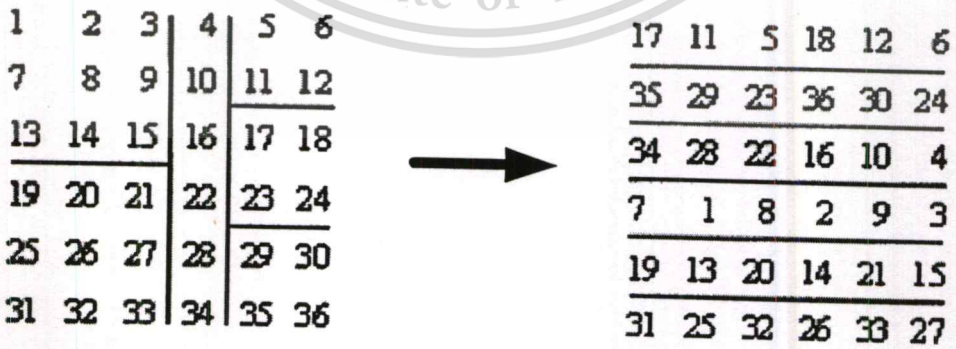


Figure.2.10 The permutation induced by the discretized baker map for 6 × 6 pixelsimage (division 3, 1, 2)

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

→

31	23	15	7	32	24	16	8
63	55	47	39	64	56	48	40
11	3	12	4	13	5	14	6
27	19	28	20	29	21	30	22
43	35	44	36	45	37	46	38
59	51	60	52	61	53	62	54
25	17	9	1	26	18	10	2
57	49	41	33	58	50	42	34

Figure.2.11 The permutation induced by the discretized baker map for 8 × 8 pixels image (division 2, 4, 2)

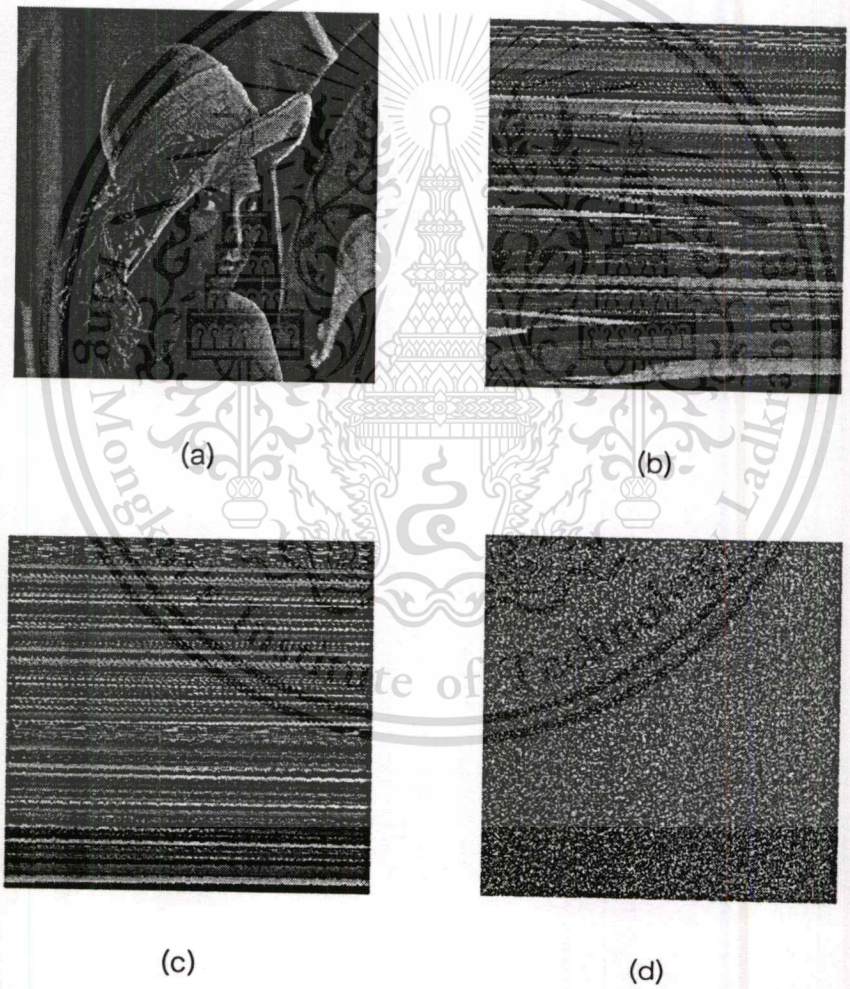
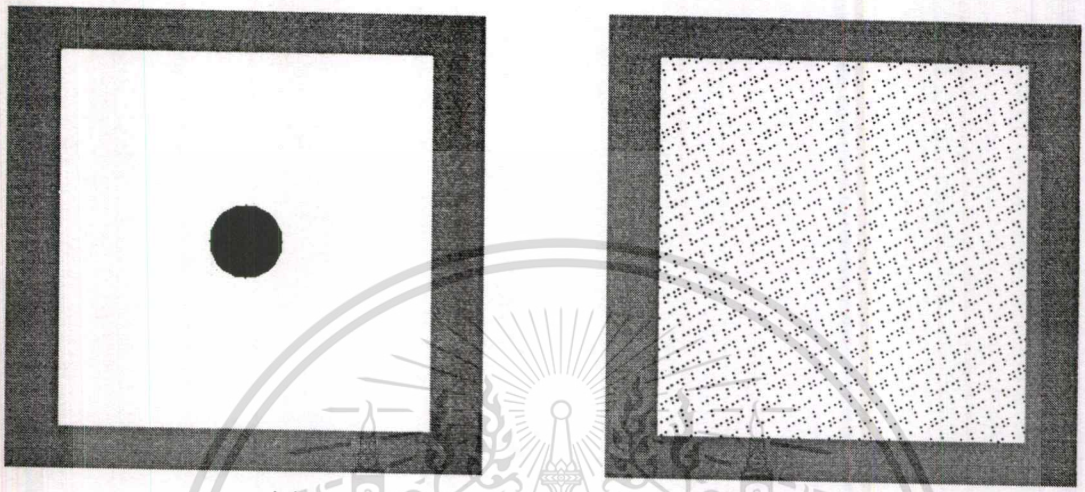


Figure.2.12 Original image (a), after applying the baker map once (b), twice (c), nine times(d)

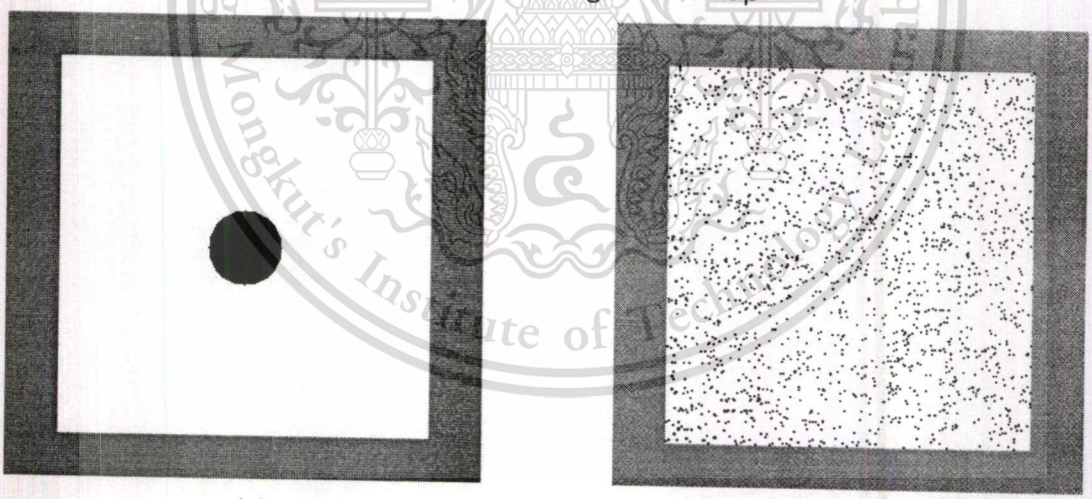
The fact that spatially localized information in the original image becomes nonlocal and uncorrelated in the encrypted image can be illustrated with the following

example. The original image consists of a black circle on a white background of a 256×256 image. A randomly generated ciphering key of baker and the initial conditions of cat map were used to iterate the discretized generalized baker map and cat map nine times. The result is shown in Figure 2.13(b) and Figure 2.14(b). The black pixels are scattered all over the image.



(a) (b)

Figure.2.13 Image consisting of black circle on a white background (a), encrypted image after nine iterations using the cat map



(a) (b)

Figure.2.14 Image consisting of black circle on a white background (a), encrypted image after nine iterations using the baker map

Chapter 3

Traditional Cryptographic Ciphers

3.1 Introduction to Cryptography

In recent years there has been a tremendous increase in the demand for digital imagery. Applications include satellite imaging, medical imaging (digital radiography), video-conferencing, digital broadcasting and scientific visualization. People want to keep the contents of their communications private. Encryption methods, like the symmetric ciphers DES or AES and the asymmetric RSA ciphers, help to achieve this goal. These ciphers and additional protocols to exchange keys or other information create a large toolbox which enables a cryptographically secure environment.

Encryption of digital data, especially multimedia data is wanted by several groups, for example by the following:

- **Individuals.** They want to protect their private data, the images from the last holiday, diary entries, ...
- **Physicians, medical institutions, and their patients.** With the increasing digitalization of inspection methods like X-ray it is convenient and — due to requests by doctors, patients or health care organizations—necessary to store and transmit this data. The emerging wishes of patients and health institutions that experts on different locations cooperate requires a well-founded framework to guarantee that the medical data can only be viewed by the legitimate people.
- **Entertainment industry.** This involves the whole chain from the artists, producers, distributors: they want to protect their business and therefore the content so that everyone who wants to be entertained pays for it.
- **News industry.** this is very similar to the above entertainment industry. The difference lies in the expected life-time of the data that should be protected. The entertainment industry tries to protect their data for a very long time (usually the number lies in the order of multiple decades), on the other hand in the news business the information is outdated very early, in the order of some minutes to days. The basic business model for that industry goes like the following: People who want to be informed prior to others should pay for that advantage. After the

initial publication the information is close to worthless and the only interest lies in archival purposes.

- **Companies with various businesses.** Of course they want to protect their trade secrets from outsiders: the documents with business plans, development sketches, but also records or transmissions of meetings, ...
- **Governments, administration and associated organizations.** They have to keep things secret, sometimes because this supports the privacy of the citizens, sometimes because it is necessary for reasons of national security. It should be noted that selective encryption as it will be presented in this thesis in many cases does not meet prerequisites like the security requirements of such agencies.
- many more . . .

We see that there are many legitimate reasons to protect, sometimes to protect fundamental rights, sometimes because law requires it, sometimes to protect some business,. . . The way to get such protection is to encrypt the data and the usual approach is to encrypt everything. To guarantee confidentiality and authenticity of information cryptographic algorithms and methods are used. Asymmetric and symmetric encryption algorithms can be distinguished. As an exceptional representative of the asymmetric procedures RSA algorithm with advantages in the field of key management and digital signature has to be mentioned. However, the processing speed of corresponding implementations is slow. These algorithms are not suitable for applications like processing of large amounts of data, i.e. hash operations and encryption of bulk data.

In contract to that symmetric algorithms encrypt rapidly large amounts of data. There are many published symmetric block ciphers which have significantly different properties. This chapter compares some symmetric block cipher (BLOWFISH, SAFER K-64, and AES) which offer different levels of security, flexibility, and efficiency.

3.2 Block Ciphers

A block cipher is a function which maps n-bit plaintext block to n-bit ciphertext block, where n is the block length. The function is parameterized by a key $k \in K$, which is assumed to be chosen at random, see Figure 3.1.

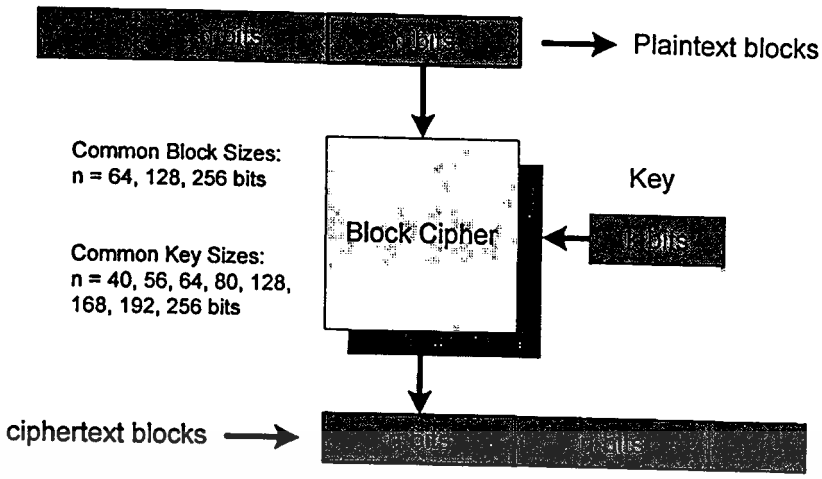


Figure.3.1 Block cipher

Block ciphers can be either symmetric or asymmetric. Many of the cryptographic systems employ symmetric block ciphers that individually provide confidentiality. Because of their versatility they provide a base for the construction of pseudorandom number generators, hash functions and stream ciphers. The algorithm employing block ciphers are typically quite heavy to process. That is because the efficiency has been traded off against security. Block size may cause problems in efficiency because the complexity of implementation of many ciphers grows rapidly with block size. Another weakness is that the whole block is used in the processing even if it contains only few bits of data. The available free memory may be wasted in this scenario. Stream ciphers encrypt individual plaintext characters one at a time and are suitable for software implementation. From the hardware point view, stream ciphers are considered faster than blocks ciphers. Furthermore, block ciphers have a more complex implementation in hardware.

Most symmetric block ciphers are based on a Feistel structure and a round function. A Feistel cipher involves dividing the plaintext into two halves and repeatedly applying a round function to the data for some number of rounds, where in each round using the round function and a key. The left half of the block is fed into the round function, and then the result is XORed with the right half of the block. Finally, after all but the last round, swap the halves of the block. The round function provides a basic encryption mechanism by composing several simple linear and non-linear operations

such as XOR, substitution, permutation, and modular arithmetic [24, 2]. Feistel cipher is shown in Figure 3.2.

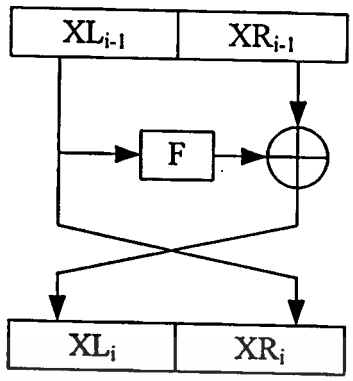


Figure.3.2 Feistel cipher

Different round functions provide different levels of security, efficiency, and flexibility. The strength of a Feistel cipher depends on the degree of diffusion and non-linearity properties provided by the round function. Blowfish algorithm bases their round function on a substitution boxes (S-boxes) as a source of non-linearity. Some ciphers (such as SAFER) use Pseudo-Hadamard Transform (PHT) and few other ciphers (such as IDEA) use multiplication in their round functions for diffusion.

3.2.1 Blowfish Algorithm

Blowfish algorithm, designed by Bruce Scheneier in 1993 [2], is a 64-bit symmetric block cipher. The key length is varied between 32 to 448-bit. Blowfish algorithm consists of two parts: sub-key generation and data encryption.

Encryption

Blowfish encryption is shown in Figure 3.3. It is a 16-round Feistel network. Each round consists of a key dependent permutation, and a key- and data-dependent substitution. All operations are:

⊕ Bitwise XOR of 32-bit words

⊞ Addition of 32-bit words modulo 2^{32}

It uses four S-boxes and P-array to encrypt data. Encryption is as follows:

Divide X into two 32-bit halves: XL, XR.

For i = 1 to 16;

$XL = XL \oplus P_i$

$XR = F(XL) \oplus XR$

Swap *XL* and *XR*

Swap *XL* and *XR* (Undo the last swap)

$XR = XR \oplus P_{17}$

$XL = XL \oplus P_{18}$

Concatenate *XL* and *XR*.

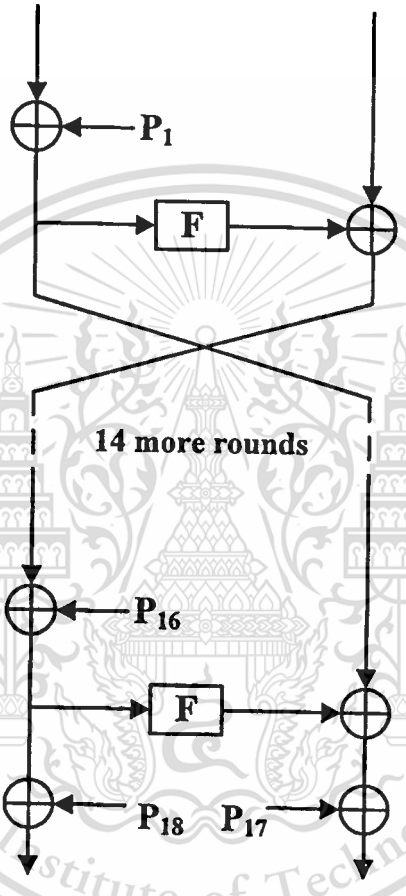


Figure.3.3 Blowfish Encryption

Function *F* is as follows, see Figure 3.4.

Divide *XL* into four 8-bit quarters: *a*, *b*, *c* and *d*.

$$F(XL) = (((S_1[a] + S_2[b]) \oplus S_3[c]) + S_4[d]). \quad (3\text{-}1)$$

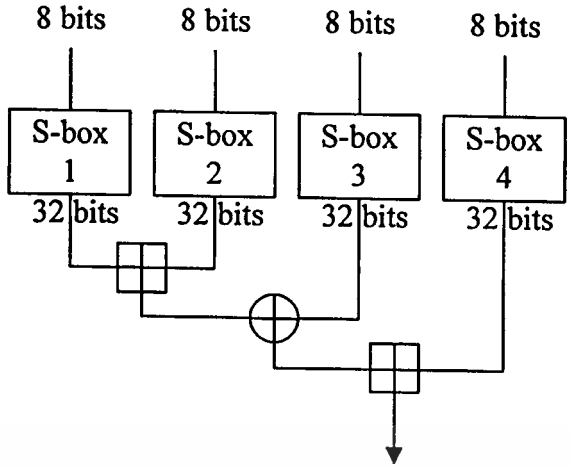


Figure.3.4 *F* function

Decryption is reverse order of encryption.

Sub-key and S-box generation

Blowfish uses a large number of subkeys. These keys must be precomputed before any data encryption or decryption.

1. The P-array consists of 18 subkeys with size of 32-bit.

   $P_1, P_2, \ldots, P_{18}$.

2. There are four 32-bit S-boxes with 256 entries each.

   $S_{1,0}, S_{1,1}, \ldots, S_{1,255}$;

   $S_{2,0}, S_{2,1}, \ldots, S_{2,255}$;

   $S_{3,0}, S_{3,1}, \ldots, S_{3,255}$;

   $S_{4,0}, S_{4,1}, \ldots, S_{4,255}$;

The subkeys are calculated using the blowfish algorithm.

1. First initialize *P*-array and four S-boxes using random number of hexadecimal digit of π (less the initial 3).
2. Perform bitwise XOR *P*-array and the key. That is XOR P_1 with first 32-bit of the key, XOR P_2 with the second 32-bit of the key, and so on for all bits of the key, possibly up to P_{14}. Repeatedly cycle through the key until the entire *P*-array has been XOR-ed with key bits.
3. Encrypt 64-bit block of all zeros using the Blowfish algorithm and the current *P*-array and *S*-array.
4. Replace P_1 and P_2 with the output of encryption in step (3).

5. Encrypt the output of step (3) with modified sub-keys.
6. Replace P_3 and P_4 with the output of step (5).
7. Continue the process to update all elements of P -array and all elements of four S-boxes.

In total, 521 iterations are required to generate all required subkeys.

3.2.2 Secure And Fast Encryption Routine (SAFER K-64)

SAFER K-64(for Secure And Fast Encryption Routine with a Key of length 64 bits), designed by James Massey in 1993, is a secret-key block-enciphering algorithm. The blocklength is 64 bits which is divided into eight byte-length sub-blocks and algorithms are byte-oriented block encryption algorithms, which have the following properties:

- use of a non-orthodox linear transform, called the Pseudo-Hadamard Transform for the desired diffusion
- use of additive constant factors(key biases) in the key expansion for avoidance of weak keys

The encryption algorithm consists of r rounds of identical transformation, followed by an output transformation; $r = 6$ is recommended but larger values of r can be used if desired for even greater security. Each round uses two 8-byte subkeys and output transformation uses one 8-byte subkey. These $2r+1$ subkeys are derived from the 8-byte user secret key. Encryption and decryption structure are slightly different. The encryption structure of SAFER K-64 is shown in Figure 3.5.

The detailed encryption round structure of SAFER K-64 is shown in Figure 3.6. The first step is the Mixed XOR/Byte-Addition of the round input with the subkey K_{2i-1}. The eight bytes of the result are then subjected to one of two nonlinear transformations:

$$y = 45^x \bmod 257 \text{ (If } x = 128, \text{ then } y = 0). \quad (3\text{-}2)$$

$$y = \log_{45} x \text{ (If x = 0, then y = 128).} \quad (3\text{-}3)$$

The mod 257 in Eq.(3-2) is the arithmetic of the finite field GF (257). The element 45 in Eq.(3-3) is a primitive element of this field. These two nonlinear operations are realized with two look-up tables of 256 bytes each.

The output of the eight nonlinear transformations is either XOR-ed or added with bytes of subkeys K_{2i}. The result then passes through three layers of linear operations called Pseudo-Hadamard Transformation (PHT).

$$b_1 = (2a_1 + a_2) \bmod 256 . \quad (3\text{-}4)$$

$$b_2 = (a_1 + a_2) \bmod 256 . \quad (3\text{-}5)$$

Between levels of the linear layer, the decimation-by-2 permutation is applied to achieve the desired diffusion. After r rounds, there is a final output transformation. This is the same as the first step of each round. The result is the 64-bit ciphertext.

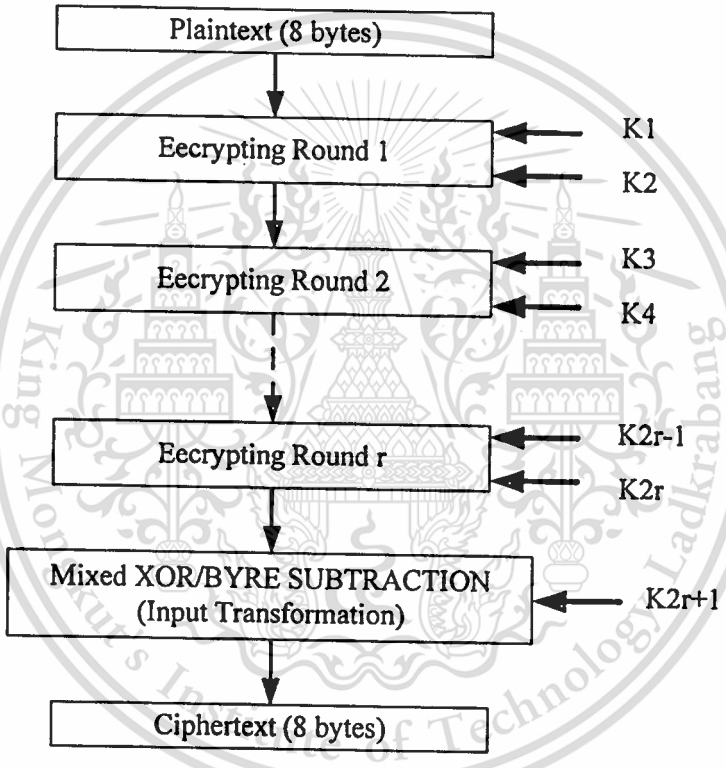


Figure.3.5 Encryption structure of SAFERK-64

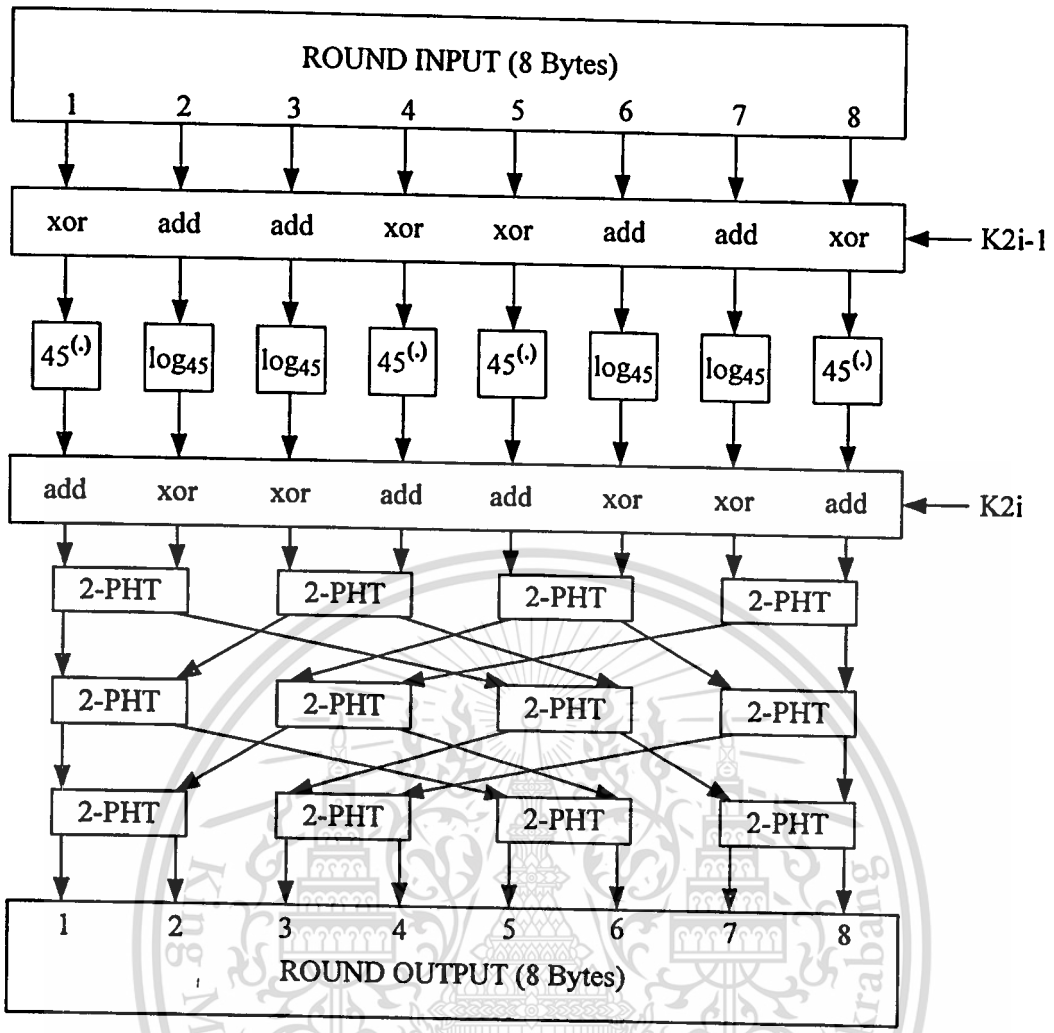


Figure.3.6 Encryption round structure of SAFER K-64

The decryption structure of SAFER K-64 is shown in Figure 3.7. The deciphering algorithm consists of an input transformation (with subtraction instead of addition) that is applied to the ciphertext block, followed by r rounds of identical transformation. The Inverse PHT (IPHT) is:

$$a_1 = (b_1 - b_2) \bmod 256 \quad (3\text{-}6)$$

$$a_2 = (-b_1 + 2b_2) \bmod 256 \quad (3\text{-}7)$$

The IPHT is as simple to compute as the direct PHT. The fan-out-by-two permutation between levels of this inverse linear layer is the inverse of the decimation-by-two permutation used in the linear layer of an encryption round. The detailed decryption round structure of SAFER K-64 is shown in Figure 3.8.

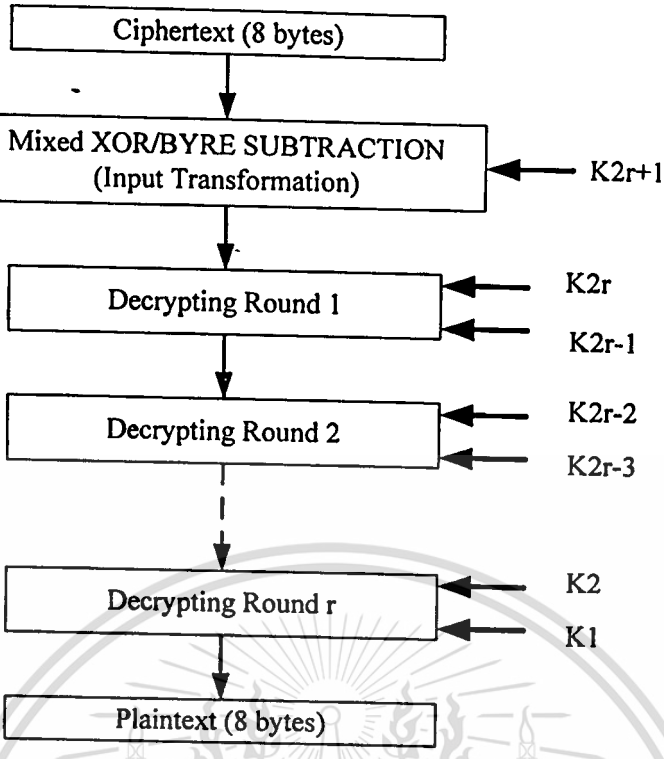


Figure.3.7 Decryption structure of SAFER K-64

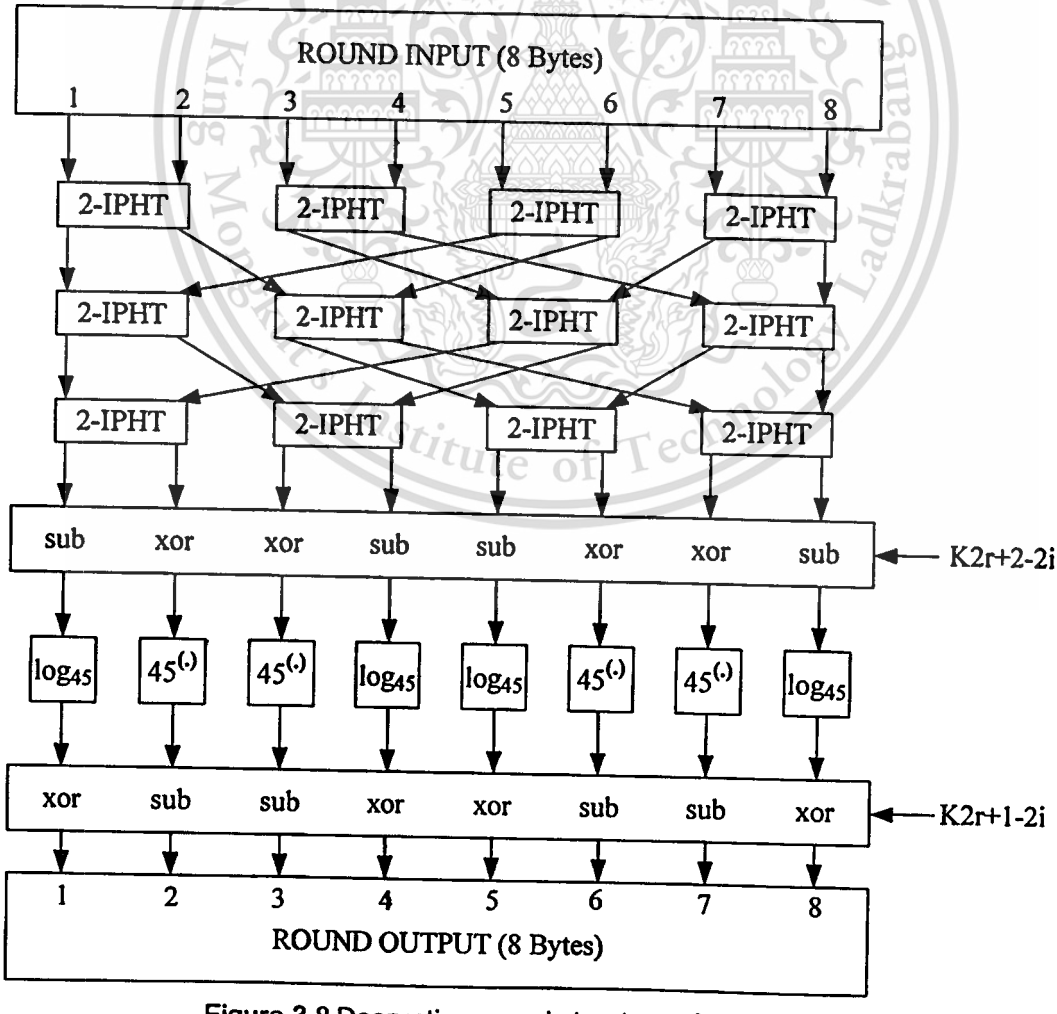


Figure.3.8 Decryption round structure of SAFER K-64

Key Schedule

Figure 3.9 shows the procedure for generating the subkeys $K2$, $K3$, ... , $K2r+1$ from the user-selected subkey $K1$. The constant $B2$, $B3$, ... , $B2r+1$ are the biases that have the purpose of ensuring that the round subkeys appear individually random. $b[i,j]$ is the j^{th} byte of bias B and all of round constants are calculated by the following formula.

$$b[i,j] = 45^{45^{\wedge((9i+j) \bmod 256) \bmod 257}} \bmod 257 \qquad (3\text{-}8)$$

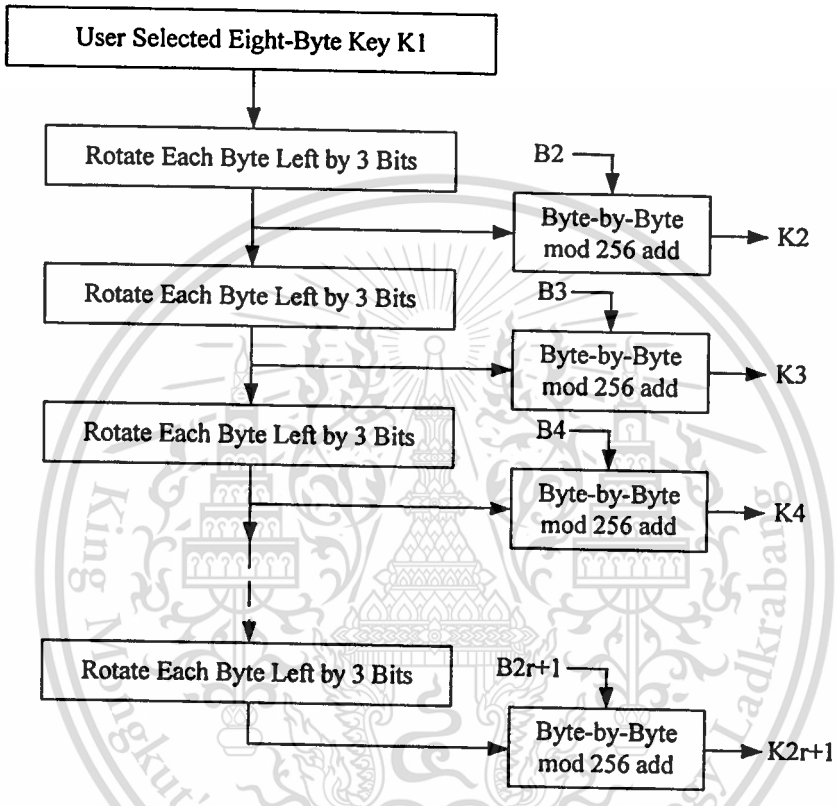


Figure.3.9 Key schedule for SAFER K-64

3.2.3 Advanced Encryption Standard (AES)

In 2000, the NIST chose Rijndael algorithm, designed by Joan Daemen and Vincent Rijmen, as Advanced Encryption Standard (AES) replacing to Data Encryption Algorithm (DES), which has been the standard since 1977. DES was developed by IBM in the early 1970. For many years, DES has been the most important target for cryptanalysis; many new techniques introduced in the last few years have been measured primarily against DES. DES is a balanced Feistel cipher, encrypting 64-bit block with 56-bit key. The cipher consists of 16 rounds. In each round, one 32-bit half of the block is fed into a round function, along with 48 bits of key; the 32-bit result is XOR-ed into the other half. DES' strength derives from the combination of 6-bit to 4-bit

S-boxes and 32-bit permutation. In the past few years, its strength is extremely sensitive to the precise S-box contents and ordering, and the specific bit permutation used. Small changes often have a catastrophic impact on security. At present, DES' short key size makes it unacceptable for most applications.

The structure of AES is very similar in many respects to the Substitution Permutation Network (SPN). It is an iterated block cipher with a simple and elegant structure. It has a variable block length b ($Nb \times 32$, where Nb is the number of 32-bits word with respect to the plaintext) and a variable keylength k ($Nk \times 32$, where Nk is the number of 32-bit words with respect to the cipher key) of 128, 192, or 256 bits. The number of rounds Nr is determined by block length and key length and varies between 10 and 14, as shown in Table 3.1.

Table.3.1 Number of rounds Nr as a function of the block and key length

Nr	$b=4$	$b=6$	$b=8$
$k=4$	10	12	14
$k=6$	12	12	14
$k=8$	14	14	14

AES does not have the Feistel structure. The round transformation is composed of three distinct invertible uniform transformations, called layers and each layer has its own function:

- The linear layer: guarantees high diffusion over multiple rounds.
- The nonlinear layer: the S-box substitution is a nonlinear operation and provides almost ideal protection against differential and linear cryptanalysis.
- The key addition layer: is a simple XOR of the round key to the intermediate State.

The transformation in a round operates on the intermediate result, called the State. The State can be written as a rectangular array of bytes with four rows and as many columns as required by the block length. State of four by four array of bytes is as follows. Initially, State is defined to consist of the 16 bytes of the plaintext X, as follows:

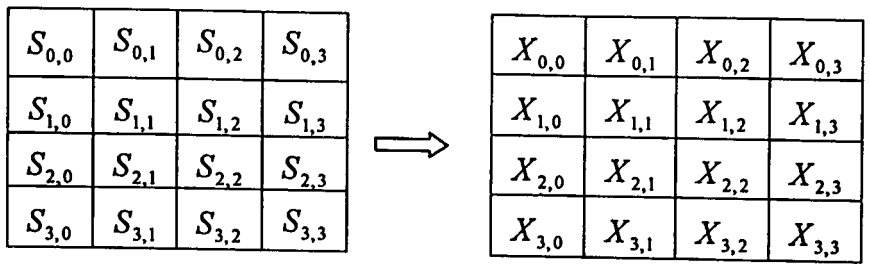


Figure.3.10 State array and plaintext

Figure 3.11 Shows algorithm overview for AES. For each of the first *Nr*-1 rounds, perform substitution operation called ByteSub on State using an S-box; perform a permutation ShiftRow on State; perform an operation MixColumn on State; and AddRoundKey. In the final round the MixColumn step is removed. For decryption, the process is reverse order using the key schedule in reverse order. The individual transformations used in encryption are InvShiftRow, InvByteSub, InvMixColumn and AddRoundKey.

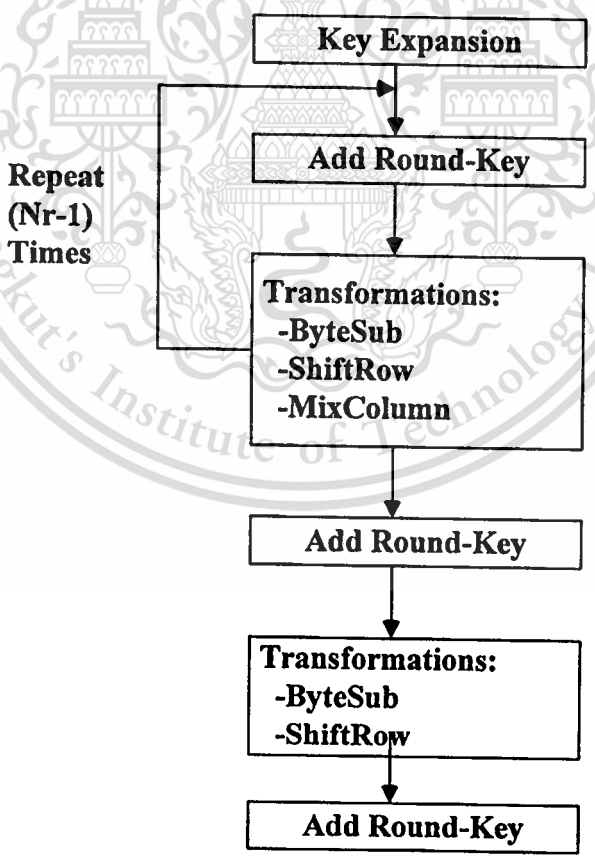


Figure.3.11 Algorithm Overview of AES

ByteSub Transformation

The transformation **ByteSub** is a nonlinear byte substitution operating on each byte of **State** independently using S-box. The entry of S-box is indexed by four leftmost bits as the row and four rightmost bits as the column. S-box is obtained by the composition of two transformations:

1. Each byte is represented as an element of *GF* (2^8) and substituted by its multiplicative inverse in *GF* (2^8). The value 0 remains unchanged.
2. Then an affine transformation over *GF* (2) is calculated as a matrix multiplication and addition of (11000110).

$$\begin{bmatrix} y \\ y \\ y \\ y \\ y \\ y \\ y \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (3\text{-}9)$$

For the inverted S-box, the affine inverse transformation is used, followed by multiplication inversion in *GF* (2^8).

ShiftRow Transformation

The rows of **State** are cyclically shifted depending on the block length as shown in Table 3.2.

The **InvShiftRow** transformation is a cyclic shift of the rows of **State** the same number of positions, but on the right.

Table.3.2 Shift offsets for different block lengths

Nb	4	6	8
Row0	0	0	0
Row1	1	2	3
Row2	1	2	3
Row3	3	3	4

MixColumn Transformation

The columns of State are considered as polynomials over $GF(2^8)$ and multiplied mod x^4+1 by a fixed polynomial $a(x)$ given by

$$a(x) = '03'x^3 + '01'x^2 + '01'x + '02' \quad (3\text{-}10)$$

where '03', '01', and '02' express hexadecimal values corresponding to x+1, 1 and x, respectively.

In the **InvMixColumn** transformation, every column is transformed by multiplying it with the polynomial $d(x)$ defined by

$$a(x) \otimes d(x) = '01' \quad (3\text{-}11)$$

and given by

$$d(x) = '0B'x^3 + '0D'x^2 + '09'x + '0E' \quad (3\text{-}12)$$

being '0*B*', '0*D*', '09' and '0*E*' the hexadecimal values corresponding to $x^3 + x^2 + x$, respectively.

AddRoundKey Transformation

In this transformation, bytes of input are simply XOR-ed with the expanded round key.

Key Expansion

Expanded key consists of total of *Nb* (*Nr*+1) words. The first four columns are filled with the key, and the others are derived from those in recursive way. Symmetry is eliminated by XORing round constant word array RCon[i] which contains the value given by $[x^{i-1},\{00\},\{00\},\{00\}]$ with x^{i-1} starting *i* at 1. Key expansion is as follows.

```
For i = 0 to Nk -1
W[i] = (K [4i], K [4i+1], K [4i+2], K [4i+3])
For i = Nk to Nb (Nr +1)
temp = W [i-1]
      If (i mod Nk = 0)
            temp = SubByte (ShiftRow (temp)) ⊕ RCon [i/Nk]
       If (Nk = 8 and i mod Nk = 4)
            temp = SubByte (temp)
W[i] = W [i-Nk] ⊕ temp
```

3.2.4 Advantageous and Disadvantageous of each Block Cipher

Blowfish algorithm is not patented. The principal advantageous of Blowfish is simplicity of design (it uses simple operations that are efficient on microprocessor). Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. The key- and data-dependent S-boxes protect against differential and linear cryptanalysis. The

key is long enough to ensure a particular security level. Blowfish with large key length results in much faster performance than IDEA which is protected by patents. Blowfish is a reasonable encryption algorithm, especially for an unpatented one. But Blowfish does not meet all the requirements for a new cryptographic standard. It is only suitable for applications where the key does not change often, like a communications link or an automatic file encryptor.

SAFER K-64 is not patented. It is a byte-oriented iterated block cipher designed for efficient implementation in both software and hardware. Using byte operation, it is useful in application such as smart cards. SAFER K-64 uses Pseudo-Hadamard Transform (PHT) to achieve desired diffusion of small changes in the plaintext or the key and diffusion provided by the PHT appears to be better than that in any other cipher. It also use additive key biases to eliminate the possibility of weak key. Structure of SAFER K-64 is more closely resemble to AES because SAFER K-64 has key-mixing layer in which subkey bytes are combined with data via XOR and addition (XOR/ADD) (AddRoundKey step in AES), one non-linear layer, where two S-boxes are applied alternately (ByteSub step in AES) and a linear transformation layer (PHT) that mixes all bytes in a block (MixColumn step in AES). SAFER K-64 is practically secure against differential cryptanalysis after 8 rounds and is adequately secure against the attack after 6 rounds. After 3 rounds linear cryptanalysis is ineffective against this algorithm. The only substantial weakness in SAFER K-64 is in the key schedule and was observed by Knudsen. Essentially the key schedule is not sufficiently complicated and the same bytes in the users supplied key are used in the same place in every round. While this attack may not impact SAFER's security when used as an encryption algorithm, it greatly reduces its security when used as a one-way hash function. In any case, Knudsen recommends at least 8 rounds. Unfortunately, there are sufficiently many reported observations and partial results available that even though there is no attack on the cipher, many people are still quite wary.

AES resistances against all known attacks. It is flexible for a variety of platforms and operating systems. AES is efficient in implementation of both hardware and software. The round transformation is parallel by design, which is an important advantage in future processors and dedicated hardware. AES can be implemented on a smart card in a small account of code, using a small account of RAM and taking a small

number of cycles. The principal advantageous of AES are the fact that the cipher and its inverse use different components practically eliminates the possibility for weak and semi-weak keys, the nonlinearity of the key expansion practically eliminates the possibility of equivalent keys, and simplicity of design (the cipher does not base its security on obscure and not well understood interactions between arithmetic operations). Traditional block ciphers are summarized as shown in Table 3.3.

Table.3.3 Traditional block ciphers

Cipher	Type	Key size	Block size	Round	Substitution	Permutation
Blowfish	Feistel network	Variable key length	64-bit	16	8×32 S-boxes	P-array
SAFER K-64	Iterated block cipher	64	64-bit	6,8,12,16	$45^x \bmod 257$ and $\log_{45} x$	PHT
AES	SP network	Variable key length	Variable block size	10,12,14	S-box	MixColumn

3.3 Conclusion

In this chapter, the structure of block ciphers has been analyzed, remarking its main advantages, disadvantageous, similarities and dissimilarities. Blowfish is not suitable for application where the keys are frequently changed. For fast speed, applications can store the subkeys rather than execute the derivation process multiple times. With SAFER-K64, even though there is no attack on the cipher, many people are still quite wary about its security. AES resistances against all known attacks. It is flexible for a variety of platforms and operating systems.

On comparison of traditional block cipher is presented in this section. Comparisons between block ciphers are shown in Table 3.4. Multimedia encryption using chaos-based algorithm will be studied and investigated later.

Table.3.4 Comparisons of block ciphers

Ciphers	Suitable	Speed	Security	Memory usage	Hardware Complexity	Patent
Blowfish	Application where the key does not change often	Middle	Middle	high	Low	No
SAFER K-64	Smart cared	Middle	Middle	Low	Low	No
AES	Variety of platform and operating system	High	High	Low	Low	No

Chapter 4

Chaotic based Image Encryption

4.1 Introduction to Image Encryption

Unlike text messages, image data have their special features such as bulk capacity, 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 consistence, and data compression for transmission, etc. Simultaneous fulfillments of these requirements, along with high security and high quality demands, have presented great challenges to real-time imaging practice. One example in point is the case where one needs to manage both encryption and compression. In doing so, if he wants to have an image encrypted after its format is converted, say from a TIFF file to a GIF file, he has to implement encryption before compression. However, for a conventional encrypted image, it has very little compressibility. On the other hand, compression will make a correct and lossless 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 solve.

High redundancy and bulk capacity generally make encrypted image data vulnerable to attacks via cryptanalysis. Based on the bulk capacity, the opponent can gain enough ciphertext samples (even from one picture) for statistical analysis. Meanwhile, since data in image have high redundancy, adjacent pixels likely have similar greyscale values, or image blocks have similar patterns, which usually embed the image with certain, patterns that results in secret leakage.

Image data have strong correlations among adjacent pixels, which makes fast data-shuffling quite difficult. Statistical analysis on large amounts of images shows that averagely adjacent 8 to 16 pixels are correlative in horizontal, vertical, and also diagonal directions for both natural and computer-graphical images. According to Shannon's information theory [3], 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.

Bulk capacity of image data also makes real-time encryption difficult. Compared with texts, image data capacity is horrendously large. For example, a common 24 bits truecolor image of 512-pixel height and 512-pixel width will occupy 512×512×24/8 = 768*K*Bytes in space. Thus, a one-second motion picture will reach up to about 19MBytes. Real-time processing constraints are often required for imaging applications, such as video-conferencing, image surveillance, etc. Vast amount 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.

4.2 Basic Features of Chaos

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

$$x_{k+1} = f(x_k), \qquad f : I \to I, \qquad x_0 \in I \tag{4-1}$$

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:

$$\exists \delta > 0,\ \forall x_0 \in I, \varepsilon > 0\ \exists n \in N,\ y_0 \in I: \quad (4\text{-}2)$$
$$|x_0 - y_0| < \varepsilon \Rightarrow \left|f^n(x_0) - f^n(y_0)\right| > \delta$$

2. Topological transitivity:

$$\forall I_1, I_2 \subset I\ \exists x_0 \in I_1,\ n \in N:\ f^n(x_0) \in I_2 \quad (4\text{-}3)$$

3. Density of periodic points in I:

Let $P = \{p \in I \mid \exists n \in N: f^n(p) = p\}$ be the set of periodic points of f.

Then P is dense in $I: \overline{P} = I$.

The sensitivity to initial conditions of chaos is often illustrated as the butterfly effect, which rooted in Lorenz's original wording "Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?" This sensitivity property is commonly utilized for the keys of cryptosystems. The topological transitivity property ensures the ergodicity of a chaotic map, which means that if partitioning the state space into a finite number of regions, no matter how many, any orbit of the map will pass through all these regions. This property is linked to the diffusion feature of cryptosystems. In chaotic cryptology, the above two properties are often used to construct stream ciphers and also block ciphers.

4.3 Relationship between Chaos and Cryptography

It is very natural to apply the discrete chaos theory to cryptography. The property of sensitive dependence of orbits on initial conditions makes the encryption nature very complicated. This prevents the system from any brute-force attack. The ergodicity implies that the state space cannot be nontrivially divided into several subspaces. So, if some orbit starts from an arbitrary point *x*, it will then never be restricted within a small region. This property indicates that if a chaotic map is used to compose encryption then the plaintext space will not be restricted to a small subspace. Thus, for ciphertext *C*, to search for the corresponding plaintext P one must go over the entire state space *X*. 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. The following excerpt from Shannon's masterpiece [3] demonstrates that cryptographic algorithms have unconsciously used the mixing property of chaos.

> Good mixing transformations are often formed by repeated products of two simple non-commuting operations. Hopf has shown, for example, that pastry dough can be mixed by such a sequence of operations. The dough is first rolled out into a thin slab, then folded over, then rolled, and the folded again, etc. . . . In a good mixing transformation . . .functions are complicated, involving all variables in a sensitive way. A small variation of any one (variable) changes (the output) considerably.

The similarities and differences between chaos and cryptography can be listed in Table 4.1. Chaotic maps and cryptographic algorithms have some similar properties: both are sensitive to a tiny change in initial conditions and parameters; both have random-like behaviors; 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, on real numbers.

Table.4.1 Similarities and differences between cryptographic algorithms and chaotic systems

Cryptographic Algorithms	Chaotic Systems
Phase space: finite set of integers	Phase space: set of real numbers
Algebraic methods	Analytic methods
Rounds	Iterations
Key (Boolean) - Discrete keyspace	Parameters (real) – Continuous keyspace
Diffusion	Sensitivity to a change in initial condition/parameters
Digital realizations by integer arithmetic	Digital realization by non integer arithmetic which approximates continuous-value systems

4.4 Chaos-Based Block Ciphers for Image Encryption

Chaos-based block ciphers have excellent flexibility. Applying block ciphering on a whole picture can achieve very fast shuffling. Meanwhile, if a huge-sized image is needed to be encrypted by a device with limited memory or computational power, say a

single chip or a mobile phone, the image can be split into several small blocks which are then encrypted in serial. Unlike stream ciphers, block ciphers are suitable for parallel processing. Moreover, using a block cipher it is easy to balance the requirements of encryption intensity and cipher speed by simply controlling the cipher rounds.

Integrating a chaotic map into a block cipher is to utilize chaos properties to rapidly scramble and diffuse data. Two general principles that guide the design of block ciphers are diffusion and confusion. Diffusion means spreading out the influence of a single plaintext digit over many ciphertext digits, so that the statistical structure of the plaintext becomes unclear. Confusion, on the other hand, means to use transformations that complicate the dependence of the statistics of the ciphertext on the statistics of the plaintext [23]. These two principles are closely related to the mixing and ergodicity properties of chaotic maps.

A general approach to chaos-based block ciphers design was provided, which consists of four steps:

1. Choosing a chaotic map: one should consider maps with good mixing property, robust chaos, and a large parameter set.
2. Introducing the parameters.
3. Discretization.
4. Cryptanalysis and key scheduling.

This framework is recapitulative and can be used to direct block ciphers design. However, to design a fast block cipher applicable to real-time imaging, more considerations are in order. For example, since images are highly correlated, it is better to choose a higher-dimensional chaotic map to speed up the permutation process. A block image cipher design framework, based on higher-dimensional chaotic maps, is recommended in [25]:

1. Choose a higher-dimensional chaotic map and generalize it by introducing a large number of parameters. The chaotic map chosen should have a large parameter set and good mixing property. In addition, the map should be a measure-preserved map, to ensure one-to-one mapping after discretization, which is needed for decryption. Good examples of such maps include the generalized cat map, generalized baker map, etc.

2. Discretize the map. Despite the fact that in theory a discretized map is only defined over a finite field therefore can never to truly chaotic, one should keep a certain strong features of chaos, such as mixing and sensitivity to parameters, while keeping the data shuffling speed fast. For practical use, this oftentimes proves sufficient.
3. Compose a diffusion process. Although pixels' positions of an image have been scrambled in the last step, generally the distribution of grey-scales of the image is still unchanged, i.e., the histogram of the plain-image is about the same as that of the cipher-image. This leaves a door widely open for statistical attack and chosen-plaintext attack. Thus, a diffusion process is necessary, to make the influence of each single pixel spreading over all of the image. The diffusion process may simply use an one-dimensional chaotic map to accomplish. In addition, one may also introduce an additional substitution procedure to speed up the diffusion process.
4. Perform security evaluation. Many cryptanalysis methods that widely used in traditional cryptography should be applied to analyze the performance of a proposed chaos-based cipher. These methods include: keyspace analysis, statistical attack, differential and linear attacks, know-plaintext attack and chosen-plaintext attack, etc.
5. Other performance evaluation. Apart from security analysis, for image encryption, other issues should also be considered. These include such as cipher speed, cipher-image size, decipher-image quality (if data compression is combined), and computational overhead, etc.

4.5 Chaos-Based Stream Ciphers for Image Encryption

Compared with a block cipher, the main advantage of a chaotic stream cipher is that it can be designed to accommodate image compression. Elaborately designing a stream cipher only introduces a small computational overhead in image coding. Moreover, if the image compression algorithm is an embedded one, i.e., the decoding procedure is from coarse to fine progressively, the cipherimage stream can also be truncated at any desired point without influencing the decoding process.

Encryption by a stream cipher is to use a sequence of random numbers to mask a sequence of plaintext of the same length, bit by bit. Although using random numbers to mask a plaintext can achieve theoretical security [3], its practical implementation is impossible. The fact is that it is practically very difficult, if not impossible, to generate a truly random number sequence with a deterministic algorithms. In practice, pseudo-random numbers are used instead. Then, the main problem is to generate pseudo-random with "good" properties to meet the need of a key stream. A commonly used pseudo-random generator (PRG) is the linear congruential generator (LCG). Since chaotic systems can generate orbits that prove to be non-distinguishable from truly random orbits (e.g., they both have broad power spectra, and they are both extremely sensitive to small changes of initial conditions), recently chaotic pseudo-random number generators (CPRNG) have attracted more and more attention.

Both compressed and uncompressed image data are treated as bit streams in stream ciphers. It is more interesting to construct pseudo-random bit sequences. Traditionally, linear feedback shift registers (LFSR) are popular generators of pseudo-random bit sequences like the m-sequence. By using a chaotic map, CPRNG is easy to construct. Assume that a dynamical system, denoted (X,ϕ), has a normalized invariant measure μ. Divide the state space X into two disjointed parts, X_0 and X_1, such that $\mu(X_0)=\mu(X_1)=1/2$. Take an initial value $x_0 \in X$ as seed, and start to revolute the system governed by ϕ and x_0. Suppose that after n iterations, a value x_n is obtained. The nth bit b_n of the sequence is then determined by the following formula:

$$b_n = \begin{cases} 0 & if \quad x_n \in X_0 \\ 1 & if \quad x_n \in X_1 \end{cases} \tag{4-4}$$

Thus, one obtains a bit sequence, $\{b_1, b_2, ..., b_n, ...\}$. Owing to the intrinsic properties of chaos, like ergodicity and mixing, the CPRNG has many good features: unique dependence of the sequence on the seed, equiprobable occurrence of "0" and "1," and asymptotic statistical independence of bits, etc.

To calculate quantities over one complete period of the generator, the following three conditions should be satisfied:

- the number of "0" bits should differ from the number of "1" bits by at most one;

- among all the runs, a half should be of length 1, a quarter should be of length 2, an eighth should be of length 3, and so on, and for each of these lengths there should be equally many runs of "0" bits and runs of "1" bits;
- the value of the autocorrelation function is equivalent to the period of the generator when the offset is 0; otherwise, the value is equal to a certain constant integer.

A practical and widely used test standard for stream cipher is specified by National Institute of Standards and Technology (NIST) in the United States, called FIPS 140-2. It consists of 4 tests on a total of 16 aspects. They are monobit test, poker test, runs test, and long run test.

Chapter 5

Performance of the Proposed Method

5.1 The New Scheme of the Proposed Chaotic Image Encryption

Chaotic maps have been utilized in several different ways in cryptography. Probably the most obvious application of chaotic maps is to use one or more one dimensional maps as pseudo-random number generators producing a binary stream which is then XOR-ed with the plaintext to produce the ciphertext. However, these schemes have been shown to produce weak ciphers. In this chapter, a method using cryptographically strong 8×8 S-box based on chaotic maps is proposed. The substitution boxes(S-boxes) have been widely used in almost all traditional cryptographic system, such as DES, AES. RC4 which is a variable-key-size stream cipher also use a 8×8 S-box [2]. The entries are a permutation of the numbers 0 through 255, and the permutation is a function of the variable-length key. To obtain dynamical 8×8 S-box, using chaotic maps is the best approach [1]. A method for obtaining dynamical S-box is obtained by iterating chaotic maps. According to perfect properties of sensitive dependent on initial condition and system parameter of the chaotic system, it is easy and convenient to obtain a class of good S-box with changing the initial condition or system parameter slightly. The method is composed of two steps: First, by iterating chaotic map, a sequence of random variables is generated and turns it to a decimal integer on the range of $0\text{-}2^n$, then an integer table can be obtained. Second, a key-dependent permuting is used to shuffle the table nonlinear by a map to achieve a more secure encryption. The shuffle table is the desirable S-box. The digitized value generated from two dimensional cat map is used to index the entry of the S-box which is XOR-ed with the values of the plain image pixel.

5.1.1 Designing dynamical S-box based on chaotic maps

First, choose two numbers: one is an initial value x which is a float number in (0,1), another a control parameter μ where $0 \le \mu \le 4$. Then use these values to compute the Logistic map.

$$x_{n+1} = \mu x_n (1 - x_n). \tag{5-1}$$

Here, we choose $x_0 = 0.1$ and the value of μ in Eq.(5-1) is selected as 3.9999. The value obtained from the logistic map is digitized by multiplying with proper scale and then the lower bits are extracted to obtain a decimal integer on the range of $0\text{-}2^n$. In this way, an integer table on the range of $0\text{-}2^n$ can be obtained. Secondly, a key-dependent permuting is used to shuffle the table nonlinearly by applying the same transformation the baker map several times. To take advantage of the diffusion, the baker map is first generalized by introducing parameters and then discretized to a finite square lattice of points. After applying the discretized baker map with a sequences of 6 divisors of 16 (2 2 4 4 2 2) to the integer table for further permutation, the required dynamical 8×8 S-box is obtained.

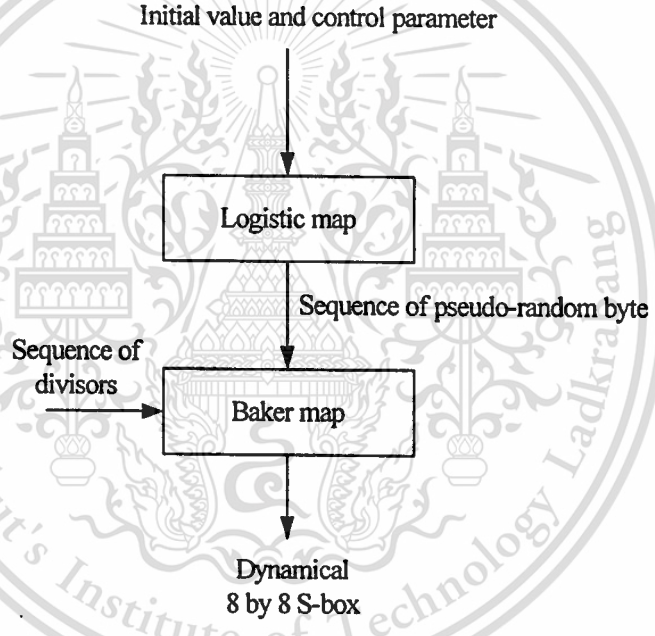


Figure.5.1 Designing dynamical S-box

Table.5.1 An example of 8×8 S-box

105	208	122	96	123	3	246	144	98	126	102	224	236	9	52	41
1	64	32	163	52	73	175	6	241	215	28	156	132	202	11	41
40	238	54	117	175	174	235	206	172	29	127	182	149	99	205	122
178	34	79	106	36	58	225	13	120	99	135	203	39	177	116	251
230	89	220	199	195	193	232	224	49	16	245	33	145	112	139	31
190	88	202	129	81	99	200	90	151	219	6	166	134	122	63	111
1	62	230	230	135	67	200	61	166	114	19	85	228	129	61	240
118	38	166	226	198	39	120	92	224	79	36	183	119	130	242	36
70	124	142	31	9	223	73	25	125	108	246	62	148	36	151	200
62	200	88	166	177	117	33	3	177	237	63	117	101	100	181	78
149	222	47	109	104	48	132	58	17	253	236	27	28	160	48	132
197	55	220	198	107	143	144	60	83	39	20	105	22	17	191	204
153	58	245	186	200	3	64	113	124	106	30	175	254	189	228	32
184	68	35	79	13	45	230	83	76	146	247	218	39	123	184	156
200	24	66	44	104	54	26	81	244	129	151	183	155	211	41	85
192	154	155	218	251	202	151	163	72	203	235	93	148	3	30	47

5.1.2 Encryption Scheme using dynamical S-box

Encryption is byte by byte operation operating on each byte. The entry of S-box is indexed by pseudo-random bytes generated from two dimensional cat map. The resultant 8 bits value of S-box is simply XOR-ed with the plaintext to produce the ciphertext and XOR-ed with the ciphertext to produce the plaintext. The fixed points of the cat map are associated with initial points whose coordinates are rational fractions. Initial points whose coordinates are irrational numbers (between 0 and 1) lead to chaotic trajectories. Here we choose the golden mean value $((\sqrt{5}-1)/2)$ which is the most irrational of all irrational numbers as the initial values of cat map. Figure 5.2 and 5.3 (b) show this encryption scheme and the enciphered image using this encryption scheme.

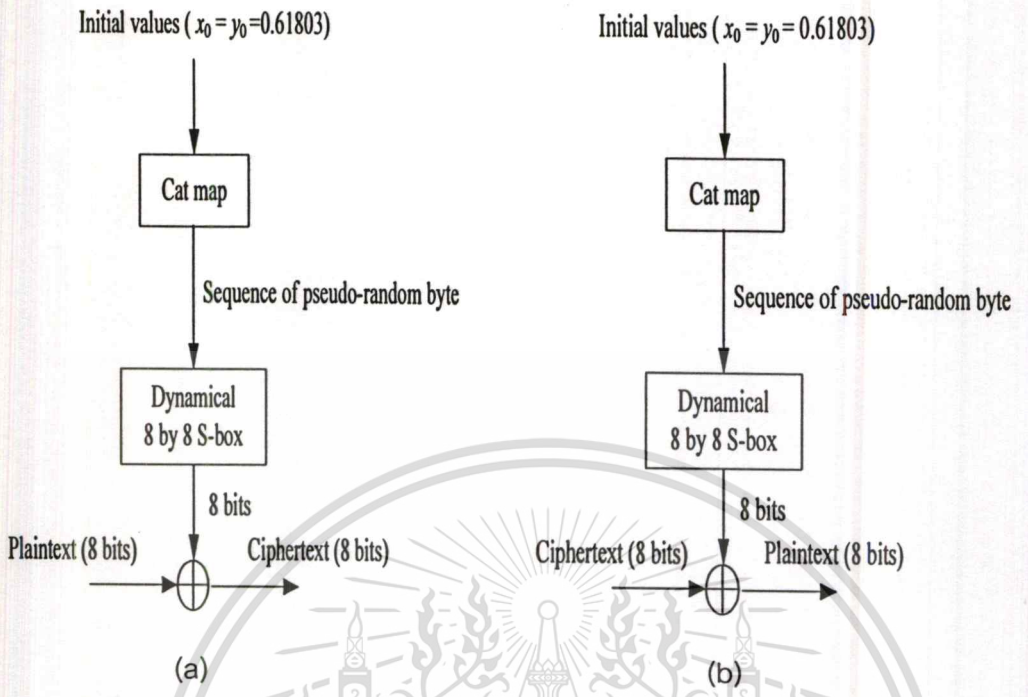
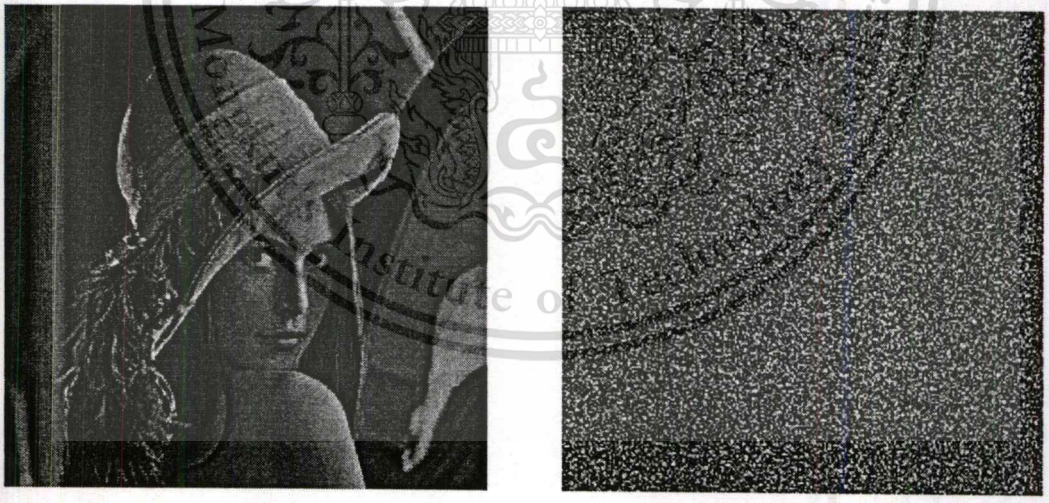
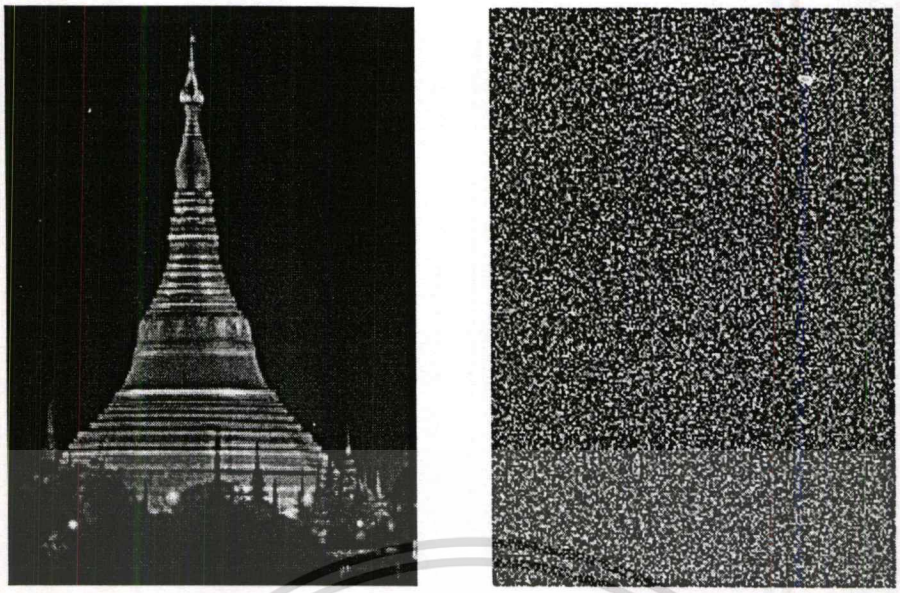


Figure.5.2 Encryption and decryption schemes: (a) encryption scheme, (b) decryption scheme



(a) (b)

Figure.5.3 Plain image of Lena (a), cipher image of Lena (b)



(a) (b)

Figure.5.4 Plain image of Shwedagone Pagoda (a), cipher image of Shwedagone Pagoda(b)

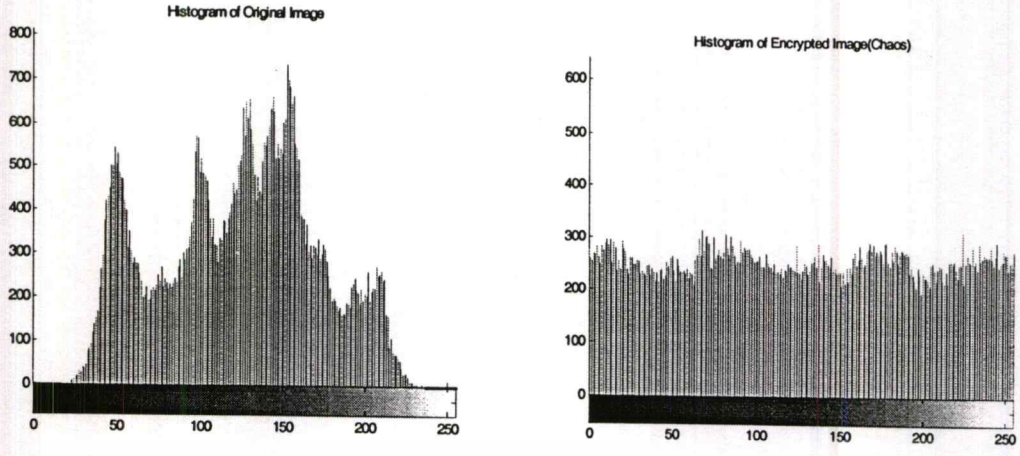
5.2 Simulation Results

We utilize the MATLAB software and use 256×256 color Lena image to perform the simulations which are:

1. Display a histogram of image data for plaintext and ciphertext.
2. Two-dimensional discrete Fourier transform and shift zero-frequency component of discrete Fourier transform to center of spectrum by plotting 2 D diagram.
3. The correlation coefficients between plain image and cipher image.
4. The correlation coefficients among different cipher images obtained by different user keys.

The simulation results are described as follows:

The histograms of the original image and cipher image are shown in Figure 5.4. The histogram of cipher image is uniform, i.e., the confusion property of cipher image is better.

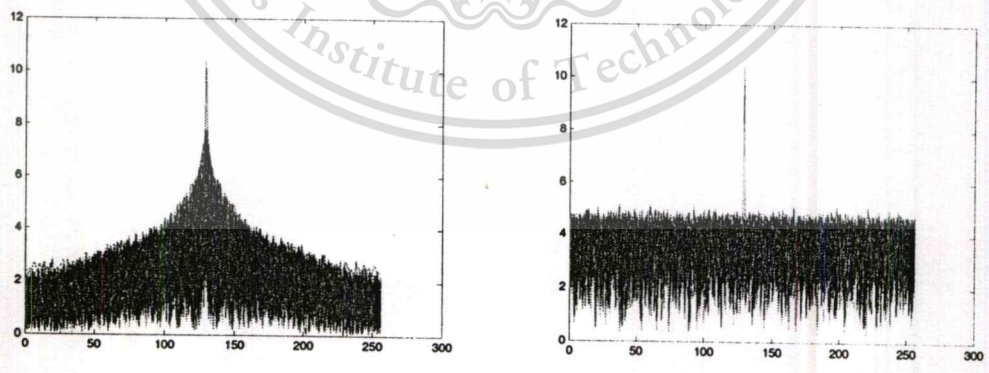


(a) (b)

Figure.5.5 Histogram of the original image (a), histogram of the encrypted image (b)

Two-dimensional discrete Fourier transform and shift zero-frequency component of discrete Fourier transform to center of spectrum by plotting 2 D diagram (see Figure 5.5) represents average distribution of spectrum energy. The result indicates the nice diffusion characteristics.

The correlation coefficient between source image and cipher image is 0.00041818. The value is nearly equal to zero and it expresses that source image and cipher image is almost independence. The correlation coefficient among different cipher images obtained by different user keys is 0.00374. The value closes zero, i.e., different cipher images are independence.



(a) (b)

Figure.5.6 The 2 D spectrum of the original image (a), the 2 D spectrum of the encrypted image (b)

5.3 Standard statistical tests and analysis

In order to test the method, we have performed certain statistical tests for various chaotic systems. The tests we use are the standard criteria specified in FIPS PUB 140-2 tests [4], which consist of four tests, totaling a number of 16 items. In the FIPS PUB 140-2 statistical tests of random numbers one considers a single bit stream of 20,000 consecutive bits output from the generator. The bits are then subjected to each of the tests below. Failure to meet any of the specified criteria means that the sequence must be rejected. The four tests, termed the monobit test, the poker test, the runs test, and the long run test, are briefly described below for completeness.

(1). Monobit Test

Count the number of ones in the 20,000 bit stream. Denote this quantity by X. The test is passed if $9{,}725 < X < 10{,}275$.

(2). Poker Test

Divide the 20,000 bit stream into 5,000 contiguous 4-bit segments. Count and store the number of occurrences of each of the 16 possible 4-bit values. Denote f (i) as the number of each 4-bit value i where $0 \le i \le 15$. Evaluate the following:

$$X = \frac{16}{5000}\left(\sum_{i=0}^{15}[f(i)]^2\right) - 5000. \qquad (5\text{-}2)$$

The test is passed if $2.16 < X < 46.17$.

Table.5.2 The required interval for runs test in the FIPS PUB 140-2 statistical tests

Length of Run	Required Interval
1	2315-2685
2	1114-1386
3	527-723
4	240-384
5	103-209
6+	103-209

(3). Runs Test

A run is defined as the maximal sequence of consecutive bits of either all ones or all zeros, which is part of a 20,000 bit sample stream. The incidences of all runs (for both Consecutive zeros and consecutive ones) of all lengths (≥ 1) in the sample stream

should be counted and stored. The test is passed if the number of runs that occur (of lengths 1 through 6) of each type is within the corresponding interval specified in Table 5.2. This must hold for both the zeros and ones; that is, all 12 counts must lie in the specified interval. For the purpose of this test, runs of greater than 6 are considered to be of length 6.

(4). Long Run Test

A long run test is defined to be a run of length 26 or more (of either zeros or ones). For the sample of 20,000 bits, the test is passed if there are no long runs. To test the quality of the random bits generated, we will have to check a total of sixteen items (one for the monobit test, one for the poker test, twelve for the runs test, and two for the long run test).

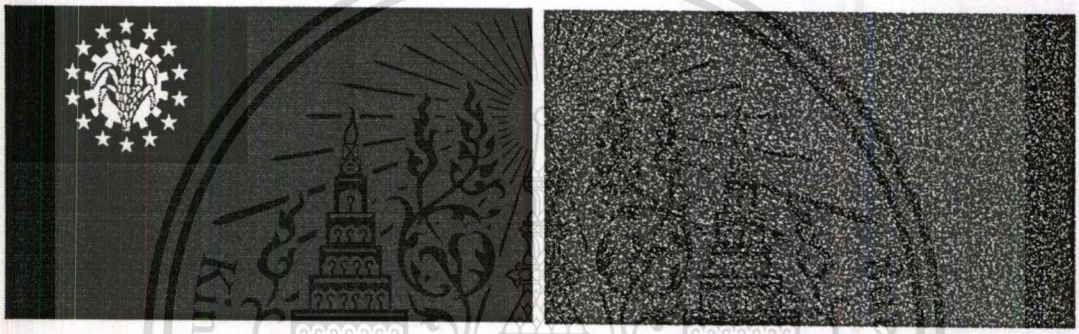
According to the property of high sensitive dependent on the initial condition and system parameter of the chaotic map, a different initial value and control parameter will result in a different S-box. In this thesis, we employ the logistic map to generate the chaotic integer table, which initial value x_0=0.1 is chosen and the value of μ in Eq.(5-1) is selected as 3.9996. By applying the baker map nine times, we can obtain the dynamical 8×8 S-box. The result of statistical test of the proposed scheme is shown in Table 5. 3.

Table.5.3 The result of statistical test of the proposed scheme

Monobit Test	9932
Poker Test	20.577
Run Test of Run 1	2321
Run Test of Run 2	1288
Run Test of Run 3	587
Run Test of Run 4	348
Run Test of Run 5	167
Run Test of Run 6	150
Long Run Test	0

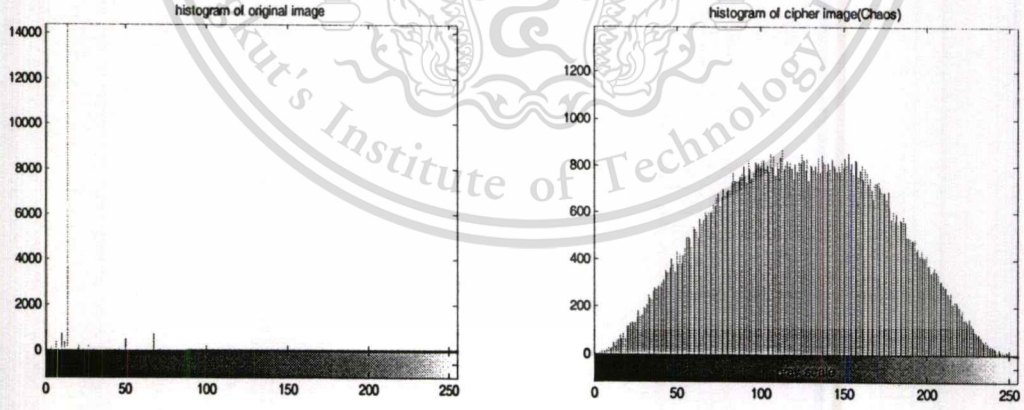
5.4 Comparison between the Proposed Scheme and the Others

The image contains an area with a fixed color, which means large redundancy. The edge of this area will be approximately preserved after encryption. There exists high redundancy in images, which may make block ciphers running in ECB (Electronic Code Book) mode fail to conceal all visible information in some plain-images. This is because those consecutive identical pixels lead to the same repeated patterns when a block cipher is used in ECB mode. See Figure 5.8 ~ Figure 5.10 for real examples of this phenomenon. Figure 5.6(b) shows the enciphered image using the proposed scheme and the image is hardly recognizable.



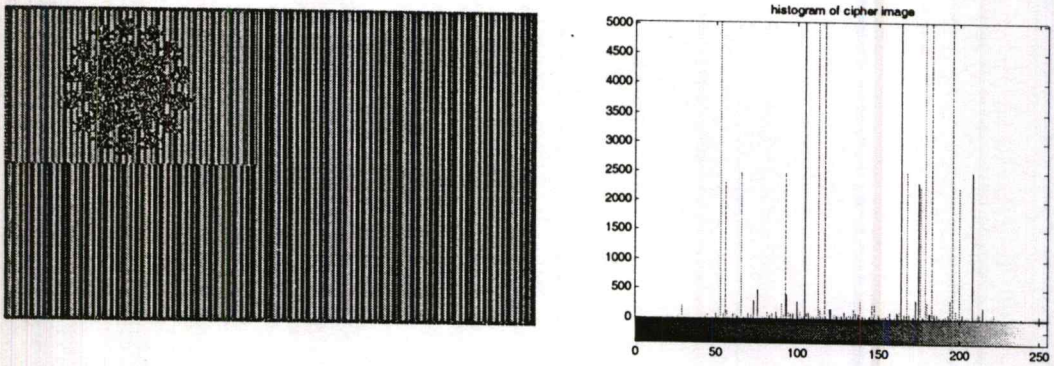
(a) (b)

Figure.5.7 Plain image (a) and cipher image (b) of Myanmar national flag



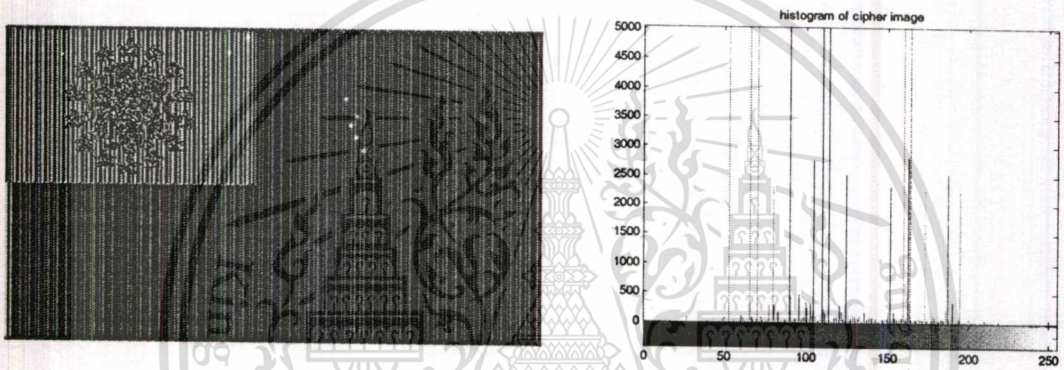
(a) (b)

Figure.5.8 Histogram of the original image (a), histogram of the cipher image (b)



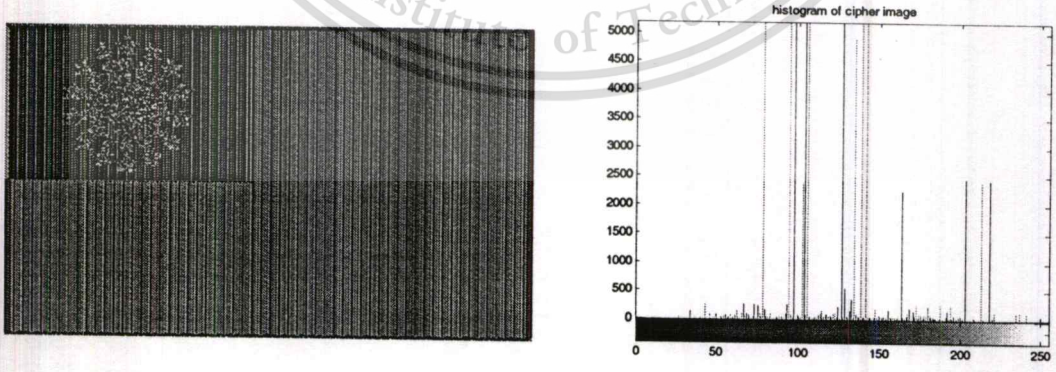
(a) (b)

Figure.5.9 Encryption using AES: (a) cipher image, (b) histogram of the cipher image



(a) (b)

Figure.5.10 Encryption using Blowfish: (a) cipher image, (b) histogram of the cipher image



(a) (b)

Figure.5.11 Encryption using SAFER-K64: (a) cipher image, (b) histogram of the cipher image

In particular, chaotic maps are easy to be implemented by microprocessors and personal computers. Therefore, chaotic cryptosystems generally have high speed with low cost, which makes them better candidates than many traditional ciphers for multimedia data encryption. Here, tests are performed on the encryption speed of the proposed chaotic cryptosystem. Given a 256×256 color image using PC which is Pentium IV 2.80GHz CPU with 704 MB of RAM, the encryption speed is as follow:

Table.5.4 Speeds of the encryption algorithms

Cipher	MB/Sec
AES	0.51032
Blowfish	0.2527
Safer-K64	0.2545
Chaos	3.4872

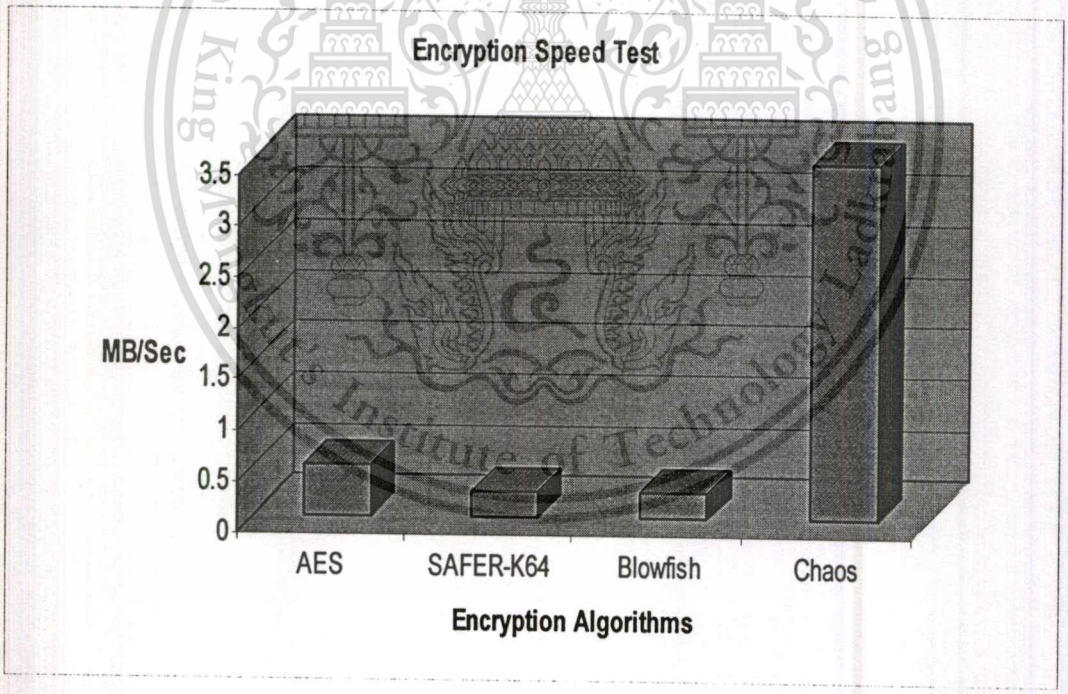


Figure.5.12 Encryption speed test

Compared with traditional block ciphers, the proposed chaos-based cryptosystem has high encryption speed. This advantage makes it suitable for large-volume data encryption such as image. In addition, the encryption process and decryption process are symmetric, and easy to be realized, which makes it suitable for multimedia encryption. The curves shown in Figure 5.13 show the relationship between

the encryption speed and the plaintext size. Seen from the figure, all curves are increasing as the plaintext size increases, that is, the encryption speed increases with the plaintext size. However, the curve of the chaotic cryptosystem rises much slower than the others. Moreover, with the increase of the block size *N*, the time difference becomes larger and larger. Therefore, for large-volume data, the chaotic cryptosystem proposed here is better overall.

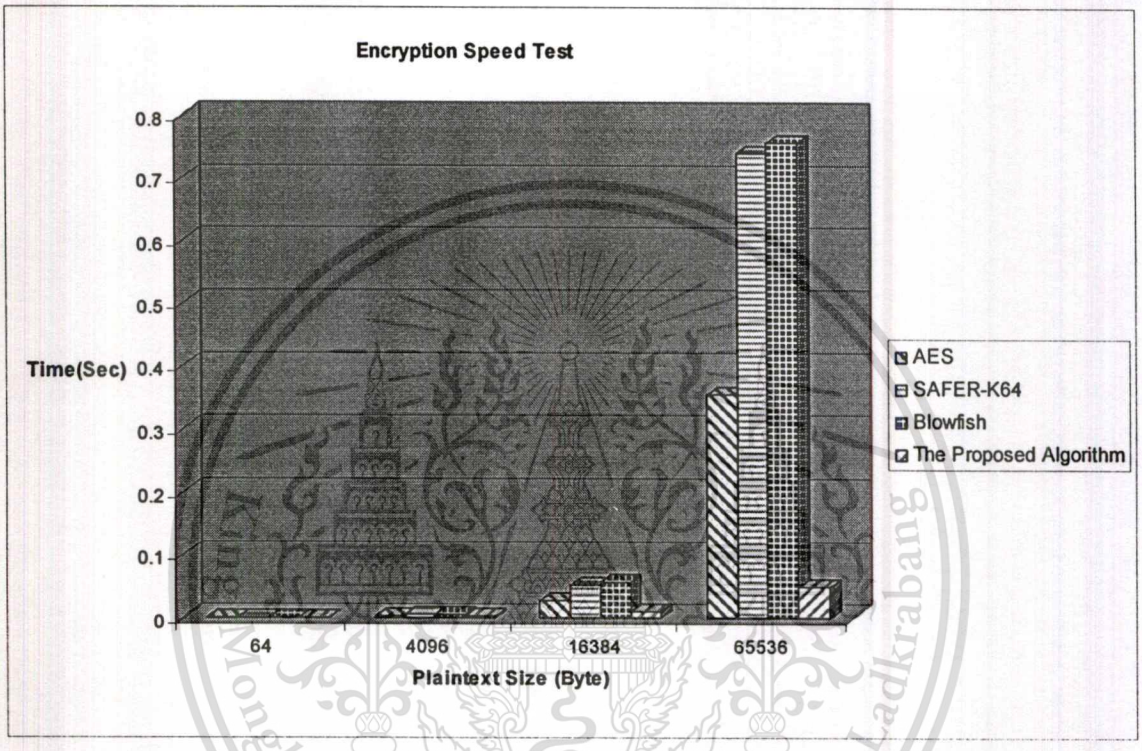


Figure.5.13 Encryption speed tests using variable plaintext sizes

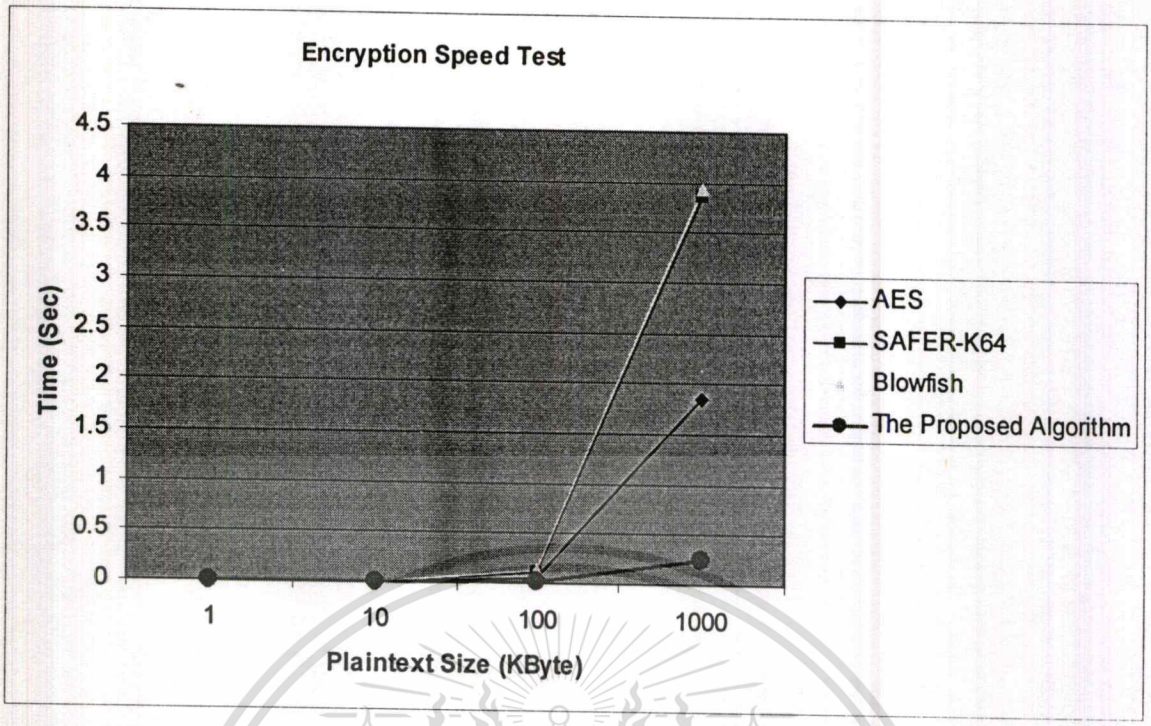


Figure.5.14 Encryption speed curves

Chapter 6

Conclusions and Further Works

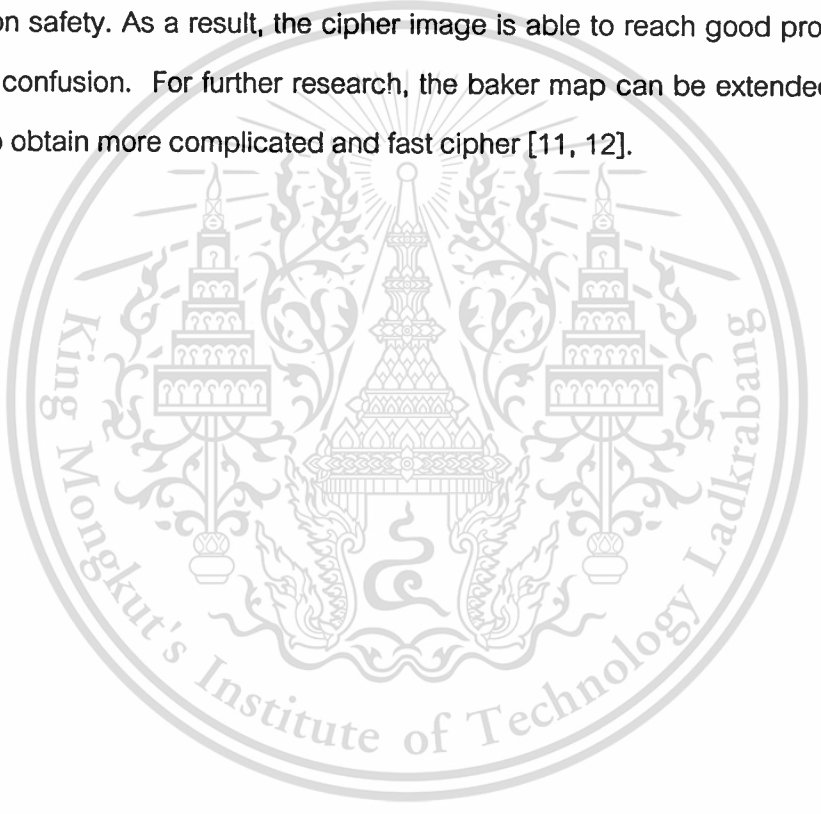
6.1 Conclusions

Image and video encryption plays a more and more important role in today's multimedia world. Although many encryption schemes have been proposed to provide security for digital images and videos, some of them are too weak to resist various attacks designed by cryptanalysts. As an emerging tool for the design of digital ciphers, chaos theory has been widely investigated especially to develop image and video encryption algorithms. The simplicity of many discrete chaotic maps and the well-established chaos theory make it possible to approach practically good solutions to image and video encryption. In this thesis, the chaotic image encryption is studied by using properties of chaos including deterministic dynamics, unpredictable behavior and non-linear transform. Our proposed new scheme of chaotic image encryption can be briefly described as follows. At first, we utilize the one-dimensional logistic map to generate the integer table, and then scramble the integer table using two-dimensional baker map. This table is the desirable S-box. Then, the value generated from two-dimensional cat is used to index the value from S-box randomly. The source image (i.e., plaintext) is performed XOR operations with the value form the S-box.

Our contribution of the proposed new chaotic image encryption scheme is that, by MATLAB simulation, the cipher images have 6 good characteristics as follows. (1)The correlation between source image and cipher image is very small, almost equal to zero. That is, the source image and cipher image are mostly independent. In addition, the correlation among the cipher images with different keys is almost zero. This represents the cipher images are mostly independent, too. (2)The histogram of cipher image pixels is uniform nearly. Namely, our cipher image holds a good confusion property on the statistics. (3)From the simulation of the two-dimensional discrete Fourier transform and shift zero-frequency component of discrete Fourier transform to center of spectrum, we can discovery the energy at the spectrum is consistent, i.e., our cipher image possesses good diffusion property on the statistics. (4)The encryption speed is faster than the other traditional ciphers and the encryption speed is increased with the plaintext size. The

larger the plaintext size, the faster the encryption speed is than the other traditional ciphers are. (5)The proposed encryption scheme passes test for FIPS PUB 140-2, Security Requirements for Cryptographic Modules of National Institute of Standard and Technology. That is, this new encryption scheme can generate usable pseudo-random numbers for the key stream. (6)In addition, the encryption scheme maintain a large enough key space.

To sum up, for our proposed new scheme of the chaotic image encryption, the cipher image characteristics of statistics on confusion and diffusion are excellent. That is, our new chaotic image encryption method can acquire very fine the ciphertext and communication safety. As a result, the cipher image is able to reach good properties of diffusion and confusion. For further research, the baker map can be extended to three dimensions to obtain more complicated and fast cipher [11, 12].



References

[1] G. Tang, X. Lieu and Y. Chen. "A novel method for designing S-boxes based on chaotic maps." *Chaos, Solitons and Fractals 23, pp. 413-419, April 2004.*

[2] B. Scheneier. "Applied Cryptography, Protocols, Algorithms, and Source Code in C." *John Wiley & Sons, 1994.*

[3] C. E. Shannon. "Communication theory of secret systems." *Bell Syst. Tech. J., vol. 28, no. 4, pp. 656-715, 1949.*

[4] National Institute of Standard and Technology and Communication Security Establishment, Derived Test Requirement (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules.

[5] G. Tang and X. Liao. "A method for designing dynamical S-boxes based on discredited chaotic map." *Chaos, Solitons and Fractals 23, pp. 1901-1909, July 2004.*

[6] G. Chen, Y. Mao and Charles K. Chui. "A symmetric image encryption scheme based on 3D chaotic cat maps." *Chaos, Solitons and Fractals 23, pp. 749-761, Dec. 2003.*

[7] G. Jakimoski and L. Kocarev. "Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps." *IEEE Trans. Circuit&Syst. I, vol. 48, pp. 163-169, Feb. 2001.*

[8] N. Masuda and K. Aihara. "Cryptosystems with Discretized Chaotic Maps." *IEEE Trans. Circuit&Syst. I, vol. 49, pp. 28-40, Jan. 2002.*

[9] L. Kocarev and G. Jakimoski. "Logistic map as a block encryption algorithm." *Physic Letters A 289, pp. 199-206, Sep. 2001.*

[10] J. Fridrich. "Symmetric Cipher Based on Two-dimensional Chaotic Maps." *International Journal of Bifurcation and Chaos, Vol. 8, No. 6, pp. 1259-1284, 1998.*

[11] G. Chen, Y. Mao and S. Lian. "A novel fast image encryption scheme based on 3D chaotic baker maps." *International Journal of Bifurcation and Chaos, Vol. 14, No. 10, pp. 3613-3624, 2004.*

[12] G. Chen, Y. Chen, X. Liao. "An extended method for obtaining S-boxes based on three-dimensional chaotic baker maps." *Chaos, Solitons and Fractals, Oct. 2005.*

[13] Robert C. Hilborn. "Chaos and Nonlinear Dynamics, An Introduction for Scientists and Engineers." *Oxford university press, 2000.*

[14] T. Kapitaniak. "Chaos for Engineers, Theory, Applications, and Control." *Springer-Verlag Berlin Heidelberg, 2000.*

[15] O. Edward. "Chaos in Dynamical Systems." *Cambridge University Press, 2002.*

[16] H. Cheng and XB. Li. "Partial encryption of compressed Images and videos." *IEEE Trans Signal Processing, vol. 48, pp. 2439-2451, 2000.*

[17] B. Furht and D. Kirovski. "Multimedia Security Handbook." *Chapter 4, Feb. 2004.*

[18] T. Uehara, R. Safavi-Naini and P. Ogunbona. "Securing wavelet compression with random permutations." *IEEE Pacific Rim Conference on Multimedia, pp. 332-335, 2000.*

[19] C. Shi and B. Bhargava. "A fast MPEG video encryption algorithm." *Proc 6th ACM Int Multimedia Conference, pp. 81-88, 1998.*

[20] T. Kohda and A. Tsuneda. "Stream cipher systems based on chaotic binary sequences." *SCIS96-11C, Jan. 1996.*

[21] Lu HP, Wang SH and Hu G. "Pseudo-random number generator based on coupled map lattices." *Int J Modern Phys, pp. 2409–2414, 2004.*

[22] T. Habutsu, Y. Nishio, I. Sasase and S. Mori. "A secret key cryptosystem by iterating chaotic map." *Lect Notes Comput Sci, 547, pp. 127–140, 1991.*

[23] L. Kocarev. "Chaos-based cryptography: a brief overview." *IEEE Circ Syst, pp.6–21, 2001.*

[24] Menezes, P. van Oorschot, and S. Vanstone. "Handbook of Applied Cryptography." *CRC Press, 1996.*

[25] Y. Mao and G. Chen. ""handbook of computational geometry for pattern recognition, computer vision, neural computing robotics." *Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2003.*

Multimedia Encryption using Tools in System Engineering

Su Su Maung[1], Kitdakorn Klomkarn[1], Noriyuki Komine[2], and Pitikhate Sooraksa[1]
[1]Department of Information Engineering
Faculty of Engineering, King Mongkut Institute of Technology Ladkrabang
Chalongkrung Rd., Bangkok, Thailand 10520
s7061143@kmitl.ac.th, kkkitdak@kmitl.ac.th, kspitikh@kmitl.ac.th
[2]Department of Applied Computer Engineering
School of Information and Electronics, Tokai University
1117 Kitakaname, Hiratsuka-Shi, Kanagawa-ken, Japan
komine@keyaki.cc.u-tokai.ac.jp

Abstract. *This paper proposes an image encryption scheme based on chaotic maps. In this method, the dynamical S-box obtained by iterating chaotic maps is used. A sequence of pseudo-random bytes generated from two dimensional cat map is used to index the entry of the S-box. The output 8 bits (0-255) of the S-box are XOR-ed with the plaintext to produce the ciphertext and XOR-ed with the ciphertext to produce the plaintext. Standard statistical tests of this scheme are performed.*
Keywords: encryption, multimedia, it-mechatronics, system engineering

1 Introduction

In recent years, cryptography has been used to send secure message over an unsecured channel. For secure communication, cryptographically secure pseudo-random bits which are used as a key stream for a stream cipher are needed. The idea of using chaos for data encryption is certainly not new and can be traced to the classical Shannon's paper [3]. Secure communication method based on chaotic maps has been utilized. One or more one dimensional maps are used as pseudo-random number generators producing a key stream which is then XOR-ed with the plaintext to produce the ciphertext. According to its own properties of sensitive dependence on initial condition and system parameter of the chaotic system, it is easy and convenient to obtain cryptographically secure pseudo-random bits with changing the initial condition or system parameter slightly. In this paper, a method using dynamical 8×8 S-box based on chaotic maps is proposed. The substitution boxes(S-boxes) have been widely used in almost all traditional cryptographic system, such as DES, AES. RC4 which is a variable-key-size stream cipher also uses a 8×8 S-box [2]. The entries are a permutation of the numbers 0 through 255, and the permutation is a function of the variable-length key. To obtain dynamical 8×8 S-box, using chaotic maps is the best approach [1]. A different initial value and control parameter will results in a different S-box. For more randomness, the values of S-box are randomly chosen by another chaotic map.
This paper is organized as follows. In Section 2, the descriptions of chaotic maps are introduced. The design of dynamical 8×8 S-box and the encryption scheme is described in Section 3 and 4, respectively while statistical tests and analysis are made in Section 5. Finally, conclusion is drawn in Section 6.

2 Descriptions of chaotic maps

The Cat map: two-dimensional invertible chaotic map introduced by Arnold and Avez. The mathematical formula is:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 \qquad (1)$$
$$= (x_n + y_n \bmod 1,\ x_n + 2y_n \bmod 1)$$

where x (mod 1) means the fractional parts of a real number x by subtraction or adding an appropriate integer. The map is known to be chaotic. The unit square is first stretched by the linear transform and then folded by the modulo operation.

The Baker map: the baker map, B, is described with the following formulas

$$B(x,y) = (2x, y/2) \qquad 0 \le x \le 1/2 \qquad (2)$$
$$B(x,y) = (2x-1, y/2+1/2) \qquad 1/2 \le x \le 1$$

The map acts on the unit square. The left vertical column $[0, 1/2) \times [0, 1)$ is stretched horizontally and contracted vertically into the rectangle $[0, 1) \times [0, 1/2)$, and the right vertical column $[1/2, 1) \times [0,1)$ is similarly mapped onto $[0, 1) \times [1/2, 1)$. The Baker map is a chaotic bijection of the unit square $I \times I$ onto itself.

The map can be generalized. Instead of dividing the square into two rectangles of the same size, the square is divided into k vertical rectangles $[F_{i-1}, F) \times [0, 1)$, $i = 1, ..., k$, $F_i = p_1 + ... + p_i$, $F_0 = 0$ such that $p_1 + \ldots + p_k = 1$. The lower right corner of the i-th rectangle is located at F_i. The generalized baker map stretches each rectangle horizontally by the factor of $1/p_i$. At the same time, the rectangle is contracted vertically by the factor of p_i. Finally, all rectangles are stacked on top of each other. Formally,

$$B(x,y) = (1/p_i(x-F_i), p_i y + F_i) \qquad (3)$$

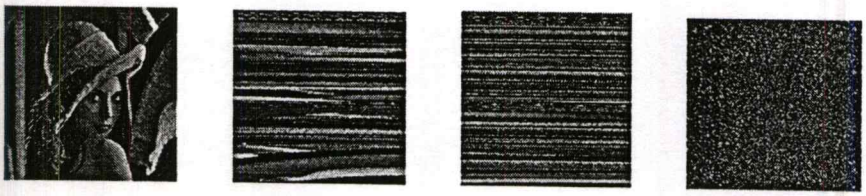
for

$$(x, y) \in [F_i, F_i + p_i) \times [0,1),$$

Since an image is defined on a lattice of finitely many points (pixels), a correspondingly discretized form of the generalized baker map needs to be derived. In particular, the discretized map is required to assign a pixel to another pixel in a bijective manner. Since the discretized map is desired to inherit the properties of the continuous basic map, the discretized map should become increasingly close to the basic map as the number of pixels tends to infinity. We define a sequence of k integers, $n_1, \ldots, n_k$ such that each integer n_i divides N, and $n_1 + ... + n_k = N$. Denoting $N_i = n_1 + ... + n_i$, $N_0 = 0$, the pixel (r, s), with $N_i \le r < N_i + n_i$ and $0 \le s < N$ is mapped to

$$B(n_1,\ldots,n_k)(r,s) = \left(\frac{N}{n_i}(r - N_i) + s \bmod \frac{N}{n_i}, \frac{n_i}{N}(s - s \bmod \frac{N}{n_i}) + N_i \right) \qquad (4)$$

The results of applying the discretized baker map to the test image after 1, 2, and 9 iterations are shown in Fig.1.



(a) (b) (c) (d)

Fig.1: Original image (a), after applying the Baker map one times (b), two times (c), nine times (d).

The Logistic map: one of the simplest forms of one-dimensional chaotic maps and mathematically its equation can be written as:

$$x_{n+1} = \mu x_n(1-x_n) \quad (5)$$

3 dynamical S-box based on chaotic maps

First, choose two numbers: one is a initial value x which is a float number in (0,1), another a control parameter μ where $0 \le \mu \le 4$. Then use these values to compute the logistic map. The value obtained from the logistic map is digitized by multiplying with proper scale and then the 8 least significant bits are extracted to easily place them in a byte array. In this way, an integer table on the range of $0\text{-}2^n$ can be obtained. Secondly, a key-dependent permuting is used to shuffle the table nonlinearly by applying the same transformation of the baker map several times. To take advantage of the diffusion, the baker map is first generalized by introducing parameters and then discretized to a finite square lattice of points. After applying the discretized baker map to the integer table for further permutation, the required dynamical 8×8 S-box is obtained.

Table 1: An example 8×8 S-box

92	34	10	145	74	248	167	72	235	47	81	139	210	245	248	69
250	149	37	7	65	7	30	33	19	152	149	254	194	231	231	34
74	154	126	253	149	253	106	158	210	213	248	61	248	253	87	244
70	138	117	160	185	7	99	58	202	88	45	9	202	9	230	188
149	27	0	202	28	210	1	101	8	253	5	255	99	206	20	198
41	139	28	196	90	92	101	1	243	235	244	224	86	115	42	195
47	239	101	244	230	255	244	119	100	0	41	253	38	96	138	2
167	2	91	94	29	240	235	2	233	17	75	184	230	10	212	132
243	185	74	237	117	203	95	250	45	165	19	125	89	233	47	180
28	209	26	119	167	152	253	186	28	246	9	73	139	36	251	216
253	2	149	168	221	9	7	61	255	36	27	20	231	124	1	216
244	38	100	255	248	129	238	12	253	255	253	202	228	0	63	26
42	82	28	211	243	16	66	154	250	35	240	146	9	172	178	244
133	222	55	168	144	255	44	92	28	1	57	229	73	4	177	236
128	115	229	250	154	0	5	44	248	3	189	20	240	14	4	145
92	34	10	145	74	248	167	72	235	47	81	139	210	245	248	69

4 Encryption scheme using dynamical S-box

Encryption is byte by byte operation operating on each byte. The entry of S-box is randomly indexed by pseudo-random bytes generated from two dimensional cat map. According to the property of high sensitive dependence on the initial condition and system parameter of the chaotic map, a different initial value and control parameter will results in a different value. These values could not be easily predicted by an adversary without knowing the initial condition. In this way, we get cryptographically secure pseudo-random bits. The resultant 8 bits value of S-box is simply XOR-ed with the plaintext to produce the ciphertext and XOR-ed with the ciphertext to produce the plaintext. Fig. 3 (b) shows the enciphered image using this encryption scheme and the image is hardly recognizable.

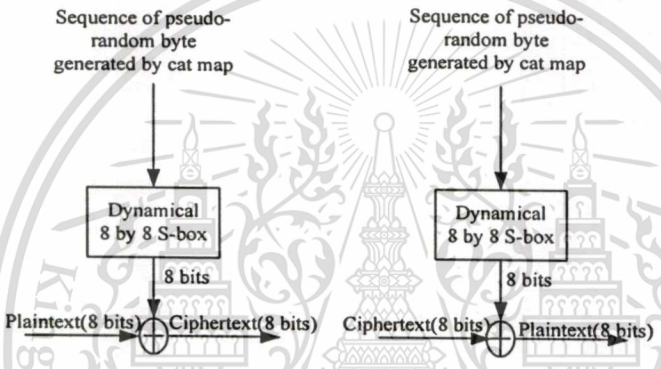


Fig. 2: Encryption scheme

(a) (b)

Fig. 3: Plain image (a), cipher image (b)

5 Standard statistical tests and analysis

In order to test the method, we have performed certain statistical tests for various chaotic systems. The tests we use are the standard criteria specified in FIPS PUB 140-2 tests [4], which consist of four tests, totaling a number of 16 items. In the FIPS PUB 140-2 statistical tests of random numbers one considers a single bit

stream of 20,000 consecutive bits output from the generator. The bits are then subjected to each of the tests below. Failure to meet any of the specified criteria means that the sequence must be rejected. The four tests are described below.

Monobit Test: Count the number of ones in the 20,000 bit stream. Denote this quantity by X. Thetest is passed if $9{,}725 < X < 10{,}275$.

Poker Test: Divide the 20,000 bit stream into 5,000 contiguous 4-bit segments. Count and store the number of occurrences of each of the 16 possible 4-bit values. Denote $f(i)$ as the number of each 4-bit value i where $0 \leq i \leq 15$. Evaluate the following:

$$X = \frac{16}{5000}\left(\sum_{i=0}^{15}[f(i)]^2\right) - 5000 \qquad (6)$$

The test is passed if $2.16 < X < 46.17$.

Table 2: The required interval for runs test

Length of Run	Required Interval
1	2315-2685
2	1114-1386
3	527-723
4	240-384
5	103-209
6+	103-209

Runs Test: A run is defined as the maximal sequence of consecutive bits of either all ones or all zeros, which is part of a 20,000 bit sample stream. The incidences of all runs (for both consecutive zeros and consecutive ones) of all lengths(≥ 1) in the sample stream should be counted and stored. The test is passed if the number of runs that occur (of lengths 1 through 6) of each type is within the corresponding interval specified in Table 1. This must hold for both the zeros and ones; that is, all 12 counts must lie in the specified interval. For the purpose of this test, runs of greater than 6 are considered to be of length 6.

Long Run Test: A long run test is defined to be a run of length 26 or more (of either zeros or ones). For the sample of 20,000 bits, the test is passed if there are no long runs. To test the quality of the random bits generated, we will have to check a total of sixteen items (one for the monobit test, one for the poker test, twelve for the runs test, and two for the long run test).

According to the property of high sensitive dependent on the initial condition and system parameter of the chaotic map, a different initial value and control parameter will result in a different S-box. In this paper, we employ the Logistic map to generate the chaotic integer table, which initial value x_0=0.1 is chosen and the value of μ in Eq.(5) is selected as 3.9996. By applying the Baker map nine times,

we can obtain the dynamical 8×8 S-box. . We generate a sequence of pseudo-random numbers by using the Cat map with parameters x_0=0.1 and y_0=0.1. The generated random bits are used to index the entries of S-box. For the 16 total number of testing items, the test result is about 75%.

6 Conclusion

In this paper, new encryption scheme using dynamical S-box and chaotic maps has been proposed. Standard statistical tests of this scheme are performed. We show that this new scheme can generate a high percentage of usable pseudo-random numbers, while maintaining a large enough key space.

Acknowledgment

This paper is supported by JICA project for AUN-SeedNet under CR Program 2005, and is supported in part by The Thailand Research Funds under grant RSA4680007.

References

1. G. Tang, X. Lieu and Y. Chen, "A novel method for designing S-boxes based on chaotic maps," *Chaos, Solitons and Fractals* 23, pp. 413-419, April 2004.
2. B. Schneier, "Applied Cryptography, Protocols, Algorithms, and Source Code in C," John Wiley & Sons, 1994.
3. C. E. Shanoon, "Communication theory of secret systems," Bell Syst. Tech. J., vol. 28, no. 4, pp. 656-715, 1949.
4. National Institute of Standard and Technology and Communication Security Establishment, *Derived Test Requirement (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules.*
5. G. Tang and X. Liao, "A method for designing dynamical S-boxes based on discretized chaotic map," *Chaos, Solitons and Fractals* 23, pp. 1901-1909, July 2004.
6. G. Chen, Y. Mao and Charles K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals* 23, pp. 749-761, Dec. 2003.
7. G. Jakimoski and L. Kocarev, "Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps," *IEEE Trans. Circuit&Syst. I*, vol. 48, pp. 163-169, Feb. 2001.
8. N. Masuda and K. Aihara, "Cryptosystems With Discretized Chaotic Maps," *IEEE Trans. Circuit&Syst. I*, vol. 49, pp. 28-40,Jan. 2002.
9. L. Kocarev and G. Jakimoski, "Logistic map as a block encryption algorithm," *Physic Letters* A 289, pp. 199-206, Sep. 2001.

Author Biography

Su Su Maung graduated the Degree of Bachelor of Engineering in Information Technology from Electronics Engineering and Information Technology Department of Mandalay Technological University (MTU) in 2003.

The author has been a lecturer at the Electronics Engineering and Information Technology Department of Yangon Technological University (YTU) before she applied to the Department of Information Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang for the Master Degree. The author's research interests include information security and cryptography.