THE DESIGN AND IMPLEMENTATION OF CRYPTOGRAPHIC SYSTEM USING CHAOTIC MAPS WITH ABSOLUTE VALUE NONLINEARITY

Sivapong Nilwong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Engineering Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2015

Thesis Title	The Design and Implementation of Cryptographic System using Chaotic Maps with Absolute Value Nonlinearity
By	Sivapong Nilwong
Field of Study	Engineering Technology
Advisor	Asst. Prof. Dr. Wimol San-Um

The Graduate School of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirements for the Master's Degree

..... Dean of Graduate School
(Assoc. Prof. Dr. Pichit Sukchareonpong)
Month……. Date……, Year…….

Thesis Committees

..... Chairperson
(Dr. Surapong Pongyupinpanich)

..... Committee
(Dr. Phaisarn Sudwilai)

..... Committee
(Dr. Pramuk Boonsieng)

..... Advisor
(Asst. Prof. Dr. Wimol San-Um)

SIVAPONG NILWONG. THE DESIGN AND IMPLEMENTATION OF CRYPTOGRAPHIC SYSTEM USING CHAOTIC MAPS WITH ABSOLUTE VALUE NONLINEARITY. ADVISOR: ASST. PROF. DR. WIMOL SAN-UM, 82 PP.

Chaotic systems, especially Chaotic maps have been implemented in various fields of study and applications such as computer programming, electrical circuits, and computer security. Besides, Android has also been operated by a large number of users around the globe. Therefore, two cryptography approaches based on Chaotic Maps with Absolute Value Nonlinearity were proposed in this thesis. The cryptographic approaches proposed in this thesis were implemented on Android mobile platform. Dynamic behaviors of the first case and the second case from Chaotic Maps with Absolute Value Nonlinearity were described in terms of equilibria and Jacobian matrix. Moreover, dynamics of the first and the second cases of Chaotic Maps with Absolute Value Nonlinearity were simulated in terms of Bifurcation diagram, Lyapunov exponents spectrum, chaotic waveforms in time domain, and chaotic waveforms in frequency domain. The encryption process and decryption process of proposed cryptographic approaches, including the key generation, image scramble and unscramble, and XOR operation process were explained. Various images at the size of 256×256 and 512×512 pixels were used to demonstrate the encryption and decryption process of proposed cryptographic approaches. The computation time of proposed cryptographic approaches were measured. Results from the encryption process of each proposed cryptography approach were analyzed qualitatively through intensity histograms and image correlation plots of pixels. The results also analyzed quantitatively through correlation coefficients, Number of Pixels Changing Rate (NPCR), Unified Averaged Changed Density (UACI), and information entropy. Results from Android applications which implemented the proposed cryptographic approaches, in which correct keys and wrong keys were utilized, are also included.

Graduate School Student's signature ……...

Field of Study Engineering Technology Advisor's signature……...

Academic Year 2015

Acknowledgements

The author wishes to express his gratitude and respectfully dedicate his work to his parent and his family for endless encouragements and love. The author is most grateful to his supervisor, Asst. Prof. Dr. Wimol San-Um for his valuable supervision, encouragements, support, and chances throughout the study. Additionally, grateful acknowledgements are made to Dr. Surapong Pongyupinpanich, Dr. Phaisarn Sudwilai, and Dr. Pramuk Boonsieng members of thesis committee, for their valuable suggestions and comments. The author also acknowledges the Intelligent Electronic Systems Research Laboratory and Academic Services Division of Thai-Nichi Institute of Technology for technical and financial supports.

Sivapong Nilwong

Table of Contents

Table of Contents (Continued)

List of Table

List of Figures

List of Figures (Continued)

List of Figures (Continued)

Chapter 1
Introduction

1.1 Introduction

This chapter introduces backgrounds of research approaches, including data encryption systems and chaos-based data encryption approaches. Motivations, statement of problem, research objectives, expected outcomes, and definition of technical terms are also included.

1.2 Backgrounds

Cryptography and data encryption can be traced back to the era of Old Kingdom of Egypt in 1900 B.C. where non-standard hieroglyphs were carved into monuments, or the Scytale of the Spartan army which were a cylinder wounded with a message parchment. Cryptography and data encryption have been improved through time and conflicts, for instance, World War I and World War II. Until the present world where the importance of data security rises as the growth of the internet which rapidly increasing. The internet grants its user limitless sources of data, and methods of storing transmitting data over the internet. However, the convenience comes with risks, especially the risks of data being stolen or destroyed by intruders or hackers who scattered on the internet. Therefore, various data encryption approaches were proposed in many fields of researches and applications such as medication, education, finance, and military. There are existing standards of data encryption algorithms such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), and Rivest-Shamir-Adleman (RSA) cryptosystem. DES was the standard for data encryption that transforms the fixed-length plain text into a cipher text of the same length using the block size of 64 bits and 64-bit key size. Unfortunately, only 56 bits were used as a key, while the remaining 8 bits were checking parity. DES become vulnerable from this small key size, and was be able to cracked in short time, in less than 24 hours. DES was then improved though time until the present, to the current standard for data encryption named AES. AES is based on Rijndael cipher which has a block size of 128 bits. Sizes of AES keys are varied between 128 bits, 192 bits, and 256 bits, depends on the number

of repetition cycles of transformation of the plain text to cipher text. At the time of composing this thesis, AES is secured in which there is no report of breaking AES keys. RSA is a public-key cryptosystem that widely used for secured data transmission, in which the encryption key is public and differs to the decryption key which is private. Recent research in the field of cryptography proposed various alternatives for the encryption of data, including chaos-based cryptography. Chaos-based cryptography implements chaotic systems into the cryptosystem. Properties of chaotic systems are utilized to satisfy requirements needed in the encryption or decryption, including the property of sensitivity to initial conditions and control parameters of chaotic systems. Chaotic systems are implemented in many means of usage in cryptography, such as generating random sequences to be used in encryption process, creating keys for encryption and decryption process, and scrambling the original data which is to be encrypted. Normally, chaos-based cryptography consists of two steps for the encryption of data, which are permutation step and substitution step. Permutation is the step of rearranging aspects of the original data. Substitution is the step of replacing or transforming aspects of the original data, which usually requires keys to be processed with the data. Although various chaos-based cryptographic approaches have been proposed, many chaotic systems implemented to cryptosystems are relatively complex, in which there are too many variables and equations in such chaotic systems. In some cases, the implementation of complex chaotic systems resulted in cryptosystems which are complicated for users who have little no experience in the field of chaotic systems.

This thesis proposes alternative approaches for chaos-based cryptosystem using chaotic maps. Chaotic maps are iterative functions in discrete-time domain which can exhibit chaotic behaviors. There were many chaos-based cryptosystems which implement chaotic maps proposed in recent research. However, most of proposed cryptosystems implement chaotic maps which have a large number of variables and equations, and some approaches implement more than one chaotic map to ensure the security of the cryptosystems. Consequently, the proposed cryptosystems became difficulties for users. The proposed chaos-based cryptographic systems in this thesis aim to implement simple chaotic maps which have only single equation and have a small number of variables. Despite the fact that various chaos-based cryptographic approaches and applications which implement chaotic map were proposed in recent

research, the proposed approaches were focused on enhancing the security which mostly applied on devices or platforms that can be reached by only some users in specified groups such as MATLAB implementation and specified hardware implementation. For ordinary users, mobile platforms are connected to users most of the times, and mostly used in daily activities. Among mobile platforms in the market today, Android is one of the most used mobile platform, in which it was claimed to be used in billions of mobile devices. Android is friendly for users, in which each feature on Android is easily understandable. Moreover, Android is open for developers to develop applications for Android platform, in which the official development environment for Android applications is available for download without any payment. Developers can also sell the developed applications on the application store for Android or the Play Store. From all of the advantages of the Android platform, the proposed cryptographic approaches in this thesis will be implemented on Android. The development of the Android applications can be accomplished using Java programming language, and XML on the official development environment, and in some other development environments HTML or C based programming languages can also be used to develop Android applications. In an early stage of this thesis, the proposed cryptographic approaches were planned to be implemented on FPGA to create the cryptosystem that can be used to encrypt or decrypt any type of file in short time. However, the FPGA approaches were removed from this thesis, due to the fact that the computation power of the FPGA is far lower than the computation power of the mobile devices in the present, which tend to be higher as the technology of computation chip goes further. Particularly, some mobile devices possess the computational power that equals to or, in some cases, more than the computational power of some personal computers. Recent research also pointed to the topic of file types to be encrypted or used to test the cryptosystems. In most of the cases, image file is one of the commonly referred file types. In this thesis, it was initially planned to create cryptographic approaches that are suitable for all types of files, by encrypting and decrypting the bit level of the file. However, the difficulties in accessing files of the types that are not commonly used in mobile devices. Hence, only image files are encrypted, decrypted, and analyzed in this thesis.

1.3 Motivations

Data storage and transmission have recently become more important in many fields of study applications such as medication, finance, education, and military, since the importance of the internet grows rapidly. The data encryption and cryptographic approaches are required to be improved to ensure the security for data transmitting and stored over the internet. Since chaotic maps have proved their performance in random number generation and data encryption, the chaotic maps will be studied and implemented in this thesis as cores for new cryptographic approaches. Due to the openness of Android which leads to various design of systems and applications, and enormous numbers of users of Android, the new cryptographic approaches will be implemented on Android mobile platform, to create cryptography applications which are easy to be used for users on Android mobile platform.

1.4 Statement of Problem and Hypothesis

Chaos-based cryptographic systems are complicated to be used by ordinary users who have little or no knowledge in chaos theory. Additionally, recent research mostly proposed cryptographic approaches which focused on images. Moreover, chaotic maps which implemented in such research are complex, in which each chaotic map has multiple equations or long equations. The complexity of chaotic maps resulted in cryptosystems which require long computation time and computation power. To avoid the problems related to computation times and powers, chaotic maps which implemented to cryptosystems have to be simple, i.e. have only one equation and have small number of variables in chaotic maps. Moreover, some chaotic maps which have a small number of variables, have problems with the dynamics, which will be discussed in chapter 2. Besides, chaos-based cryptosystems have to be simple applications which capable of encrypting and decrypting many file types, to provide convenience in security of computer data for users.

1.5 Objectives

1.5.1 To study Chaotic Maps with Absolute Value Nonlinearity and its implementation on Android.

1.5.2 To create chaos-based cryptographic systems on Android.

1.6 Research Scopes

1.6.1 Chaotic Maps with Absolute Value nonlinearity are studied.

1.6.2 Data type to be experimented is image file.

1.7 Expected Outcomes

1.7.1 Gained knowledge on chaotic maps and how to implement chaotic maps on Android.

1.7.2 Gained the chaos-based cryptographic system on Android.

1.8 Definitions

1.8.1 MATLAB is a computer program for computing mathematical operations from basic mathematics such as adding or subtracting to advanced mathematics such as image processing or artificial intelligence. MATLAB stands for MATrix LABoratory and is originally written for matrix computations [1]. Each of mathematic operations in MATLAB is written in MATLAB commands which is simple mathematic signs, and advanced MATLAB commands are written in human language. Thus, MATLAB commands are easy to understand which makes MATLAB widely used in researches. Apart from mathematical computations, MATLAB can also be used as the computer programming editor using high-level programming language which is the MATLAB commands combined together as a script for the computer program. Unfortunately, MATLAB is not capable of generating the standalone computer program from the MATLAB codes without specified plugin, and the MATLAB codes are runnable only on MATLAB. Consequently, the MATLAB codes from MATLAB have to be written in another programming language again to create the standalone computer program.

1.8.2 NIST stands for National Institute of Standards and Technology. NIST is an agency of the department of commerce of the United States of America. NIST was established in 1901 to remove the industrial competitiveness handicap of the United States at the time.

The major handicap of the industrial competitiveness was the capability of the measurement infrastructure of the United States that lagged behind United Kingdom, German, and other economic rivals. In the present, NIST measurements and standards supports all ranges of technologies, and are implemented in many researches [2].

1.8.3 Cryptosystem is the systems or the methods of cryptography which typically include set of algorithms, keys, and procedures that required to encrypt or decrypt the specified data [3].

1.8.4 Cipher text is the encoded message resulting from the encryption [3].

1.8.5 Plain text is the original unencrypted message, or the result from the successful decryption [3].

Chapter 2
Related Theories and Literature Reviews

2.1 Introduction

This chapter gives information of related theories including chaos theory, bifurcation, Lyapunov exponents, chaotic maps, random number generation, digital images, intensity histograms, Android, cryptography, number of pixel changing rate (NPCR), unified averaged changed density (UACI), and entropy. The literature reviews of related research on chaos-based cryptography, cryptography, and chaotic systems are also included.

2.2 Related Theory

2.2.1 Concept of Chaos Theory

Chaos theory is the study of dynamical systems that are sensitive to initial conditions. Chaos theory also focused on the study of phenomena such as attractors, bifurcations, chaos, fractals, catastrophes, and self-organization which used to describe the systems which changed as time passed. Small differences in initial conditions of the system can be resulted in the divergence of outcomes of chaotic systems. Therefore, the results from chaotic systems are difficult to be predicted. Moreover, the divergence of outcomes or chaotic behaviors can be appeared in the deterministic system, in which results of the system are predictable using the initial conditions, without influences from random elements. In chaotic phenomena, seemingly random events are predictable from simple deterministic equations. Thus, a phenomenon that is locally unpredictable may be globally stable, exhibit clear boundaries, and display sensitivity to initial conditions, or the property that is known as the Butterfly Effect.

2.2.2 Bifurcation

The term bifurcation was first proposed by Henri Poincaré in 1885. The bifurcation theory is the study of changes in topological structures of a family such as solutions of a family of differential equations [4]. Bifurcation theory also concerns about solutions of the system, in which solutions varies with the parameter of the

system. At present, the term bifurcation is used to describe the sudden changes in the dynamics of the system in which the behavior of trajectories in the neighborhood of the fixed point will change when the fixed point changes character. Changes in fixed point character are the result of the changes in control parameter value. For chaos theory, bifurcation theory is usually employed as the bifurcation diagram, for the analysis of chaotic systems. The bifurcation diagram qualitatively indicates possible values from the system, as the parameter of the system changes. Figure 2.1 illustrates and example of the bifurcation diagram which is the bifurcation diagram of the logistic map. Blue dots in figure 2.1 refer to possible values of the state variable of the logistic map. From figure 2.1, area of significant numbers of dots refer to the chaotic region of the system.

2.2.3 Lyapunov Exponents

The Lyapunov exponents or characteristic Lyapunov exponents, provides a measurement of the degree of instability of the system, Lyapunov exponents quantifies the mean rate of divergence of trajectories which start infinitesimally close to their references, in which the distance of two points which initially imperceptibly close to each other may change through time [5]. The difference through time in the distance of two points in the system which start close to each other can be implied as the first derivative of the function of the system. The largest Lyapunov exponent can be calculated from the equation described as

$$\lambda = \lim_{n\to\infty}(\frac{1}{N})\sum_{n=1}^{N}\ln f'(x_n) \tag{2.1}$$

where λ is the Lyapunov exponent, N is the number of iterations, x_n is the state variable, and f'(x_n) is the first derivative of the system. Chaotic behaviors are characterized by the positive Lyapunov exponent, in which two points in the chaotic systems move apart as time passed. Lyapunov exponents can be plotted into spectrums as the example in figure 2.2. The area of positive Lyapunov exponents can be noticed in figure 2.2, which is the same area as the chaotic region in the bifurcation diagram.

2.2.4 Chaotic Maps

Chaotic maps are iterated functions in discrete-time domain which are capable of performing chaotic behaviors. Chaotic maps possess properties of chaotic systems similar to other chaotic systems, including the sensitivity to the initial conditions and control parameters of chaotic maps [6]. Normally, chaotic maps are expressed as iteration functions in the form of

$$x_{n+1} = f(x_n) \tag{2.2}$$

where x_{n+1} is the value of the state variable in the next state, x_n is the value of the state variable in the current state, and $f(x_n)$ is the function of the chaotic map. Chaotic maps can be analyzed through various means of analysis. However, two types of analysis which typically referred in recent research were bifurcation diagram and Lyapunov exponents. Lyapunov exponents provides a measurements of the instability of the system and quantify the mean rate of divergence of trajectories that start imperceptibly close to the reference, as described in section 2.2.3 of this thesis. Since results from Lyapunv Exponents are numbers from equations, it can be concluded that Lyapunov exponent is one of the quantitative analysis for chaotic maps. In contrast, the bifurcation diagram indicates the possible long-terms values of chaotic maps. The term bifurcation is used to describe sudden changes in dynamics of the system, in which the behavior of trajectories near the fixed point of the system will change as the fixed point changes its character, as mentioned in section 2.2.2 of this thesis. For chaotic maps, changes in the value of the control parameter cause fixed point character to be changed. Figure 2.1 shows the bifurcation of the chaotic map called the logistic map. Displayed in figure 2.1, the bifurcation diagram indicates possible values of the logistic map as dots, where the value of the control parameter is varied from zero to four. Moreover, there exist only dots which indicate possible values of the chaotic map in the bifurcation diagram, without any numerical value to quantitatively expresses the possible values. Hence, it can be concluded that the bifurcation diagram is one of the quantitative analysis for chaotic maps. There are various chaotic maps existing and implementing in many fields of study and applications. However, the chaotic map which was frequently referred in recent research is the logistic map.

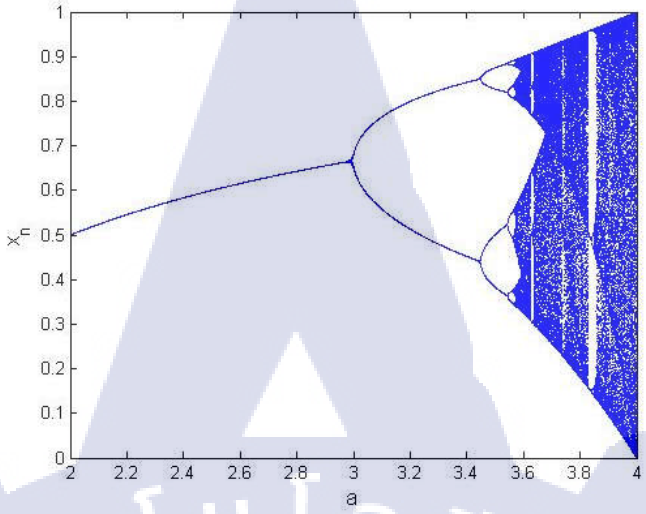


Figure 2.1 Bifurcation diagram of the Logistic map.

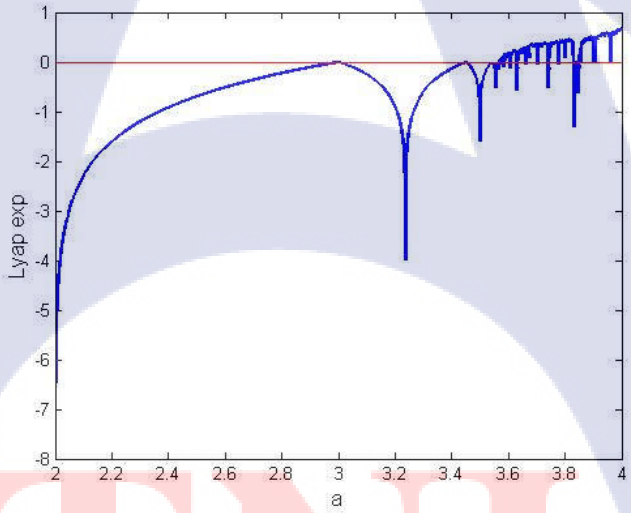


Figure 2.2 Lyapunov exponent spectrums of the Logistic map.

Logistic map is a polynomial recursive function of degree two, and also the chaotic map which frequently used to explain chaotic behaviors from a simple recursive function. Logistic map can be described mathematically as

$$x_{n+1} = rx_n(1-x_n) \tag{2.3}$$

where x_n is the state variable of the logistic map which has the value in the range of [0, 1] and r is the control parameter of the logistic map which has the value of interest in the range of [0, 4). The bifurcation diagram of the logistic map is shown in figure 2.1, and the Lyapunov exponents spectrum of the logistic map is shown in figure 2.2. From the bifurcation diagram in figure 2.1, the number of dots which represent possible values of the state variable of the logistic map rises greatly where the value of the control parameter r has its value in the range between approximately 3.5 and 4. Therefore, it can be concluded from the bifurcation diagram that the logistic map performs chaotic behaviors in this region. However, there are some areas of control parameter value in the chaotic region of the bifurcation diagram, that contain only few possible values of the logistic map, or the white area in the chaotic region. These areas are called the periodic windows, the area which the results from the chaotic map become periodic, without any chaotic behaviors. Lyaponov exponents spectrum in figure 2.2 also gives the same result as the bifurcation diagram, in which the logistic map performs chaotic behaviors where the value of the control parameter is between approximately 3.5 and 4, which can be inspected from the values of the Lyapunov exponents that are more than zero in this region. Moreover, Lyapunov exponents spectrum also shows the periodic windows of the logistic map, in which the values of Lyapunov exponents are less than zero in periodic windows areas.

2.2.5 Chaotic Maps with Absolute Value Nonlinearity

Chaotic Maps with Absolute Value Nonlinearity are chaotic maps which contain absolute value nonlinearity. Initially, Chaotic Maps with Absolut Value Nonlinearity were proposed as four cases of individual chaotic maps which described mathematically as [7]

$$x_{n+1} = |1 - ax_n| \tag{2.4}$$

$$x_{n+1} = |-1 + ax_n| \tag{2.5}$$

$$x_{n+1} = 1 - a|x_n| \tag{2.6}$$

$$x_{n+1} = -1 + a\,|\,x_n\,| \tag{2.7}$$

where x_n is the state variable which has the value of interest in the range of [0, 1], and a is the control parameter which has the value of interest in the range of [0, 2] for all cases. From four cases, the first case and the second case contain the absolute value nonlinearity for the entire function of chaotic maps. On the contrary to the first case and the second case, the third case and the forth case contain the absolute value nonlinearity only at the state variable. Each case of Chaotic Maps with Absolute Value Nonlinearity also has different Jacobian matrix and fixed point. All cases of Chaotic Maps with Absolute Value Nonlinearity have been tested with NIST statistical test suite 800-22rev1a for randomness, by testing random sequences generated from each case of chaotic maps. Test results from the NIST statistical test suite stated that all cases of Chaotic Maps with Absolut Value Nonlinearity passed all of 15 tests in the test suite. The analysis of the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity is enclosed in chapter 4.

2.2.6 Random Number Generation

Random number generation is the process of generating a number or set of numbers randomly. Random numbers are implemented in many fields of applications such as selecting random data in statistics, gambling, and gaming. There are two main courses of generating random numbers using computers, which are pseudo-random number generation, and true random number generation.

The pseudo-random number generation uses algorithms, pre-calculated tables, or mathematical formulae to generate sets of numbers that appear to be random. Despite the fact that numbers generated from pseudo-random number generation process are random, the sets or sequences of numbers from some pseudo-random number generators are periodic, in which the generated sets or sequences of random numbers can repeat themselves. Moreover, sequences or sets of numbers generated from the same pseudo-random number generator yield the same result if the same value of variables are used in pseudo-random number generator. Hence, typical pseudo-random number generators are practically not suitable for encryption.

The true random number generation uses randomness of physical phenomena and introduces the randomness to the computer. Sources of randomness in true random number generation may include radioactive source, lava lamp images, atmospheric noises, and several more physical sources. The results from true random number generation are nondeterministic, and cannot be reproduced. Besides, true random number generation requires longer computation time than the pseudo random.

2.2.7 Digital Images

An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite, or discrete quantities, the image is a digital image [8]. Digital images are composed of a finite number of elements, each of elements has a particular location and value, and elements are denoted by the term of pixel. Spatial resolution is a measure of the smallest discernible detail in an image. While spatial resolution can be measured in various ways and units, one of the most used measurements is the dots per inch (dpi). Dots per inch refers to the number of dots or pixels in one inch, in which the more dpi the image has the more resolution the image will have and more quality of the image will be. There is also the intensity resolution which refers to the smallest discernible change in intensity level. The number of intensity levels is usually the power of two, where the resolution of 8 bits is the common resolution. The 8-bit intensity resolution of the image means there are 2^8 or 256 intensity levels for the image.

2.2.8 Intensity Histogram

The intensity histograms in this thesis refer to intensity histograms of intensity levels in an image. The image histogram is a bar chart that represents gray levels of an image [9]. An image with 8-bit intensity resolution has 256 shades of gray, or 256 gray levels to describe all shades of tone from black to white. Black has a level of zero, while with has a level of 255. The x axis of the histogram shows the intensity levels in grayscale, while the y axis shows the number of pixels which have intensity of the specified level of gray. Due to the capability of image histograms that display only gray level, multiple image histograms can be used to display levels of intensities of each

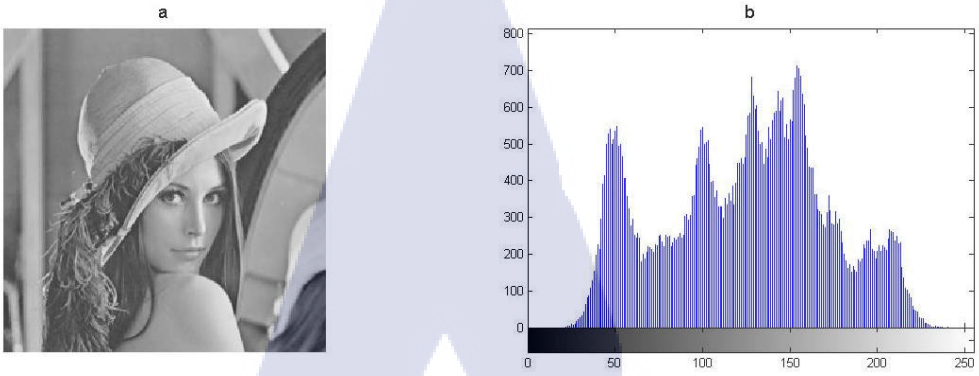


Figure 2.3 (a) LENA image in grayscale, (b) histogram of the LENA image.

color plane, i.e. red, green, and blue, in which three histograms are required for typical colored images in the red, green, and blue (RGB) color system. Figure 2.3 shows the LENA image in grayscale and its histogram.

2.2.9 Android

Android is the open source platform for mobile devices which is popular among millions of users around the globe. Android is open for developers and users, in which there is an open marketplace for Android, the Play Store. Play Store allows Android users to purchase applications or digital contents created by developers [10]. For developers, Android also provide the official integrated development environment (IDE), named Android Studio to developers for free. Android Studio provides developers tools for the development of Android applications, such as editor for the required programming languages and an interface of the developing Android application. Android Studio is also included with the Android virtual device (AVD) which is the set of configurations for the Android device which is to be emulated by the emulator in the Android Studio [11]. The Android emulator allows developers to test the developing applications in the environment that is similar to the real Android device, in which Android emulator has the same capabilities as the actual Android device, except for the phone call. Despite the official development environment provided by Android, there are various development environment that can be used to develop Android applications such as the Eclipse IDE which integrated with the Android development tools.

The development of native Android application typically uses the Java programming language and XML language. XML is used to create the interface of the application and manipulate settings of the application, while Java programming language is used to write the process and actions of the application. The process of the development of the Android application starts with the setup of the development environment and the virtual devices, similar to the development of computer programs The development process is then moved to the development phase, and continued to test and debug phase, then the process ends with the publishing of the application [12]. Despite the fact that the developing application can be tested on virtual devices, the test of the Android application can be accomplished on the actual Android device instead, with risks of device failures that can be caused by the developing application.

2.2.10 Cryptography

Cryptography is the process of creating codes and implementing codes to secure data transmission. Cryptography consists of encryption and decryption. Encryption is the process of hiding the original data by transforming the original data into a form that cannot be read, recognized, or utilized. Decryption is the process of obtaining the original data from the encrypted data be transforming the encrypted data into a form that can be read, recognized, or utilized [3]. Typically, there are two methods of encrypting the original data in cryptography, which are bit stream method and block cipher method. The bit stream method transforms each bit in the plain text or the original data into a cipher bit or the encrypted bit, one bit at the time. The block cipher method separates the original data into blocks, then each block is transformed into an encrypted block, which each encrypted block contains encrypted bits.

For algorithms in cryptography, the cryptographic algorithms can be separated by symmetry of keys, into two groups, symmetric algorithm and asymmetric algorithm. The symmetric algorithms are cryptographic algorithm which utilized the same key for encryption decryption and decryption process. Symmetric algorithms are sometimes called the private key encryption, due to the fact that keys needed to be transferred through private channels. Symmetric algorithms use mathematical operations that can be implemented into fast computing algorithms that can quickly encrypt or decrypt the specified information even by the small computing devices such as mobile devices. The

asymmetric algorithms are different to the symmetric algorithms, in which the asymmetric algorithms utilized two different keys for encryption and decryption. Either of two keys can be used for encryption or decryption. Assumes there are two keys which are key A and key B, if the plain text was encrypted by key A, only key B can decrypt the encrypted text. On the other hand, if the plain text was encrypted using key B, only key A can decrypt the encrypted text. The asymmetric algorithms are most effective when one of two keys is private, while the another key is public.

There is also a topic of the key space in cryptography. The time required to acquire keys of encryption though the mean of brute force increases as the number of bits in the key increases. For example, the encryption keys in this thesis has 128 bits in each key, which will take 5,257,322,061,209,440,000,000 years to crack the key, estimated from a personal computer which performs 8 million guesses per second [3].

Steganography is a technical term that often coined when cryptography is discussed. Steganography is similar to cryptography, in which the intention of steganography is also to conceal the secret information. Despite changing the information as cryptography does, steganography hides parts of the information within other information, such as images, sounds, and documents separately instead.

Apart from cryptography, there is also an art of cryptanalysis. Cryptanalysis is the process of obtaining the original data from the encrypted data or cipher text, without any knowledge on algorithms or keys used in the encryption process of cryptography. There are various attacks in cryptanalysis such as statistical attacks which attackers might know some information of the plain text and use statistical methods to achieve the plain text, and differential attacks which attackers use similar information to determine the differences in the cipher text and may exploit such vulnerabilities in slight differences to recover the secret key, or the algorithms of cryptosystems in some cases.

2.2.11 NPCR and UACI

Number of Pixel Changing Rate (NPCR) and Unified Averaged Changed Intensity (UACI) are measurements of two encrypted images which each of the measured encrypted image has the similar plain image to one another. There are slight differences in two plain images which is to be encrypted in to the images tested with

NPCR and UACI. Normally, differences are only at the pixel level, i.e. one plain image is only one pixel different to another plain image. NPCR and UACI are used to test cryptosystems and encryption systems that related to images, against differential attacks which aim to obtain the plain image, or keys of the encryption by applying similar plain images into cryptosystems and analyze results from the encryption. NPCR measures the changing rate of pixels of two encrypted images which have the similar plain images that are only one pixel different. The equations of NPCR are described as

$$NPCR = \sum_{x,y} \frac{D(x,y)}{T} \times 100\,,\ D(x,y) = \begin{cases} 0, if C_1(x,y) = C_2(x,y) \\ 1, if C_1(x,y) \neq C_2(x,y) \end{cases} \tag{2.8}$$

where $C_1(x, y)$ and $C_2(x, y)$ are intensities of pixels of the first and the second encrypted image, respectively, T is the number of pixels in encrypted images, and (x, y) is the coordinate of pixel. Typically, the ideal value of NPCR is approximately 99 percent, in which pixels of two encrypted images, which have the plain images that are different at only one pixel, are changed almost entirely. UACI measures the changes in the intensities of two encrypted images which have the plain images that are only one pixel different to each other. The equation of UACI is described as

$$UACI = \sum_{x,y} \frac{C_1(x,y) - C_2(x,y)}{T \times F} \times 100 \tag{2.9}$$

where $C_1(x, y)$ and $C_2(x, y)$ are intensities of pixels of the first and the second encrypted image, respectively, T is the number of pixels, and F is the maximum value of intensity of the intensity resolution of the encrypted images which is 255 for 8-bit images. The ideal value of the UACI is approximately 33 percent, according to test results from the recent research [13]. The more UACI specifies more changes in intensities of encrypted images which their plain image is only one pixel different.

2.2.12 Entropy

The term entropy is used in various fields of study. However, the term entropy in this thesis refers to the information entropy, or the Shannon entropy. Shannon

entropy was first proposed in 1948, and named after Claude Shannon. Shannon entropy is a measurement of uncertainty associated random variable. In details, Shannon entropy quantifies the value of information contained within the message, calculated from probability of information [14]. The calculation of entropy is described as

$$H(X) = -\sum_{i=1}^{n} P_i \log_q P_i \tag{2.10}$$

where $H(X)$ is the entropy of random variable X, n is the total number of possible values of the random variable X, i is the number of possible value of the random variable X in the set of possible values, P_i is the probability of the random variable X has the value equal to value in the set of possible values of variable X at i, and q is the base of the logarithms which is usually 2, 10, and natural number e. For digital images which have the intensity resolution of 8 bits. The ideal value of the entropy is 8, since the ideal value of the entropy can be achieved from the message which each of information has same probability in the message. Applying 1 of 256 probability and 256 intensity levels into (2.10), the value of 8 is the result from the calculation of the entropy.

2.3 Literature Reviews on Chaotic Systems and Cryptography

Table 2.1 summarizes researches related to the proposed research approach. According to the earliest research in Table 1, Moni Naor and Adi Shamir proposed the first approach to the visual cryptography [15]. The visual data of the interest in this work are texts and images which are typical visual data at the time. The model of the visual cryptography is the separating of the plain text or image into n shares in which can be decrypted by overlaying all shares together. Since this was the first approach proposed in the field of visual cryptography, there was no test results against any attacks to the data available.

M. Ahmad and M. S. Alam proposed the encryption and decryption approach for images by employing various chaotic maps, including 2D cat map, 2D coupled logistic map, and the logistic map [16]. The plain image in the proposed approach was shuffled using 2D coupled logistic map, in which control parameters were generated using 2D cat map. The shuffled image was processed with the sequence generated from

Table 2.1 Summary of researches related to the proposed approaches.

Author	Year	Proposed Schemes
H. Hsiao and J. Lee [17]	2015	Multiple chaos-based cryptosystem for fingerprint security, using four chaotic systems.
R.F. Martinez-Gonzalez and J.A. Diaz-Mendez [18]	2014	Implementation of Bernoulli's Map-based chaotic stream cipher on FPGA and test with NIST test suite.
G. Savithri and K.L. Sudha [19]	2014	Android application for secret image transferring and reception using Hénon map as a key generator.
H. Liu et al. [20]	2014	Chaos-based image encryption approach using bijection and S-box.
W. San-Um and P. Kettong [7]	2014	Chaotic maps with absolute value nonlinearity which are simple and robust that can be used in various encryption schemes.
Ch.K. Volos et al. [21]	2013	The image encryption scheme which uses the true random bit generator based on the interaction between two nonlinear circuits.
L. Merah et al. [22]	2013	The design and implementation of Lorenz chaotic system on FPGA.
Y. Wu et al. [14]	2013	The measurement of randomness in images through Shannon entropy.
M. Ahmad and M. S. Alam [16]	2010	Image encryption and decryption approach using 2D cat map and 2D coupled logistic map.
M. Naor and A. Shamir [15]	1994	Cryptographic scheme using n shares of the plain image.

the logistic map. The results from the encryption were analyzed through intensity histograms, correlation coefficients, and entropy. The results from the analysis were close to ideal values of the encrypted image.

Y. Wu et al. proposed the measurement of randomness in digital images using the local Shannon entropy. The proposed measurements were able to over weaknesses which utilized global Shannon entropy, including unfair randomness comparisons between images of different sizes, failure to discern image randomness before and after image shuffling, and possible inaccurate scores for synthesized images [14]. The image to be measured was separated into blocks, then each block was computed with the local entropy. The results stated the accuracy that were more than global Shannon entropy measurements, and also revealed the unsecured results from earlier researches which claimed the encryption results were secured.

L. Merah et al. proposed the implementation of Lorenz chaotic system on FPGA [22]. The implementation of the Lorenz chaotic system on FPGA used Xilinx System Generator (XSG) to design the block set which is to be programmed on FPGA. The implementation also represents data on 32 bits which 12 bits are for entire and the remaining 20 bits are for fraction. The designed system block set is then used to generate VHDL code using Xilinx ISE design suite before simulating on simulator and programming on FPGA device. The FPGA which was implemented with Lorenz chaotic system was tested by communicating with a personal computer through RS-232 port. Universal Asynchronous Receiver Transmitter (UART) was used for data transmitting between computer and FPGA in this research with data transfer rate of 9600 bit/s. Test results revealed that chaotic signals generated from Lorenz chaotic system on FPGA can be operated and utilized in other applications which require chaotic system to be implemented.

Ch. K. Volos et al. proposed the image encryption process based on chaotic synchronization phenomena which used two same nonlinear circuits which are connected together and synchronized as the chaotic true random bit generator [21]. The random bit sequences are generated from this random bit generator to be used as the key for the image encryption. The external voltage source which produced pulses is connected to the system of two nonlinear circuits to switch the synchronization state of the system to create random bits of 0 and 1. The generated random bits sequence are

used in XOR operation between the pixel values of the plain image and the random bit sequence. The result satisfied the test from the statistical attack since the pixel values are uniformly distributed in the encrypted image. However, the result did not satisfy the tests from the differential attacks and resulted in the encryption system the required to be operate twice to make the satisfied results.

W. San-Um and P. Kettong proposed the simple and robust chaotic maps with absolute value nonlinearity in 4 cases or 4 chaotic maps as described in section 2.2.5 [7]. Chaotic maps with absolute value nonlinearity was generalized in terms of bifurcation diagram, LE spectrum, and Cobweb plot for all cases. The chaotic maps with absolute value nonlinearity are tested with the test suite from the NIST statistical test suite from 800-22 rev1a special publication. The test results which take the chaotic random number sequence from the chaotic maps to the test was satisfied since the chaotic sequences generated from the chaotic maps passed all 15 tests in the NIST statistical test suite which test for their randomness.

G. Savithri and K. L. Sudha proposed the Android application for secret image transmission and reception using chaotic steganography [19]. The proposed Android application in this work used the Henon chaotic map which consist of two equations to generate the chaotic random numbers sequences to be used as a key and used in the pixel scrambling of the plain image. At the sender side of the secret image transmission system, the plain image is scrambled using the number sequence generated from the Henon map and then encrypted and embedded to the cover image through the random pixel insertion. At the receiver side, the cover image embedded with the encrypted image is extracted to get the encrypted image, then the encrypted image is decrypted and unscrambled to get the plain image with the random number sequence generated from the Henon map with the same initial conditions and control parameters. The result from the application satisfied the test on the sensitivity of the password. However, there is no test results against any type of attacks included.

R. F. Martinez-Gonzalez and J. A. Diaz-Mendez proposed the implementation of a stream cipher based on Bernoulli's map on FPGA using VHDL [18]. The implementation consists of two parts: the first part is two pseudo random bit generators based on the Bernoulli's map, and the second part is a XOR gate array. Pseudo random bit generators acquired initial conditions and control parameters to operate. Mechanism

of the proposed pseudo random bit generator consists of multiplexing and feedback loop. Outputs from the random bit generators stored in parallel register and separated into eight 8-bit sequences. Random bit sequences from both pseudo random bit generators are then sent to XOR array to get the keystream used by stream cipher. The result from XOR array was tested by NIST statistical test suite. Test performed in this research were frequency test, block frequency test, run test, cumulative sum test (forward), cumulative sum test (reverse), and FFT test. Test results from six tests was satisfying with obtained P-value closing to one in all six tests.

H. Liu et al. proposed chaos-based encryption approach for color images which designed based on bijection [20]. The plain image of each color plane was separated into blocks of the same size. The S-box was generated using the Chen system. The separated blocks of the plain image were operated with the S-boxes through substitution. The results from the substitution were combined as the cipher image. The results from the proposed encryption approach were analyzed through intensity histograms, correlation coefficients, entropy, NPCR, and UACI. The uniformly distributed of intensity histograms were obtained from all color planes of the encrypted image, while the correlation coefficients were close to zero. Moreover, the entropy value was relatively close to 8, NPCR values were approximately 99 percent, and UACI vales were approximately 33 percent, close to ideal values.

H. Hsiao and J. Lee proposed chaos-based biometric image cryptosystem for fingerprint security. The proposed chaos-based cryptosystem in this thesis utilized four chaotic systems, consists of two 1-D chaotic systems and two 3-D chaotic systems [17]. Therefore, the security of the proposed cryptosystem was strong, due to the large key space. The encrypted results in grayscale from the proposed cryptosystem were analyzed through intensity histograms, pixel correlation plots, NPCR, UACI, and entropy. The results from the analysis were close to ideal values of encrypted images. The proposed cryptosystem also passed tests from NIST SP 800-22a which test the randomness of the cipher image from the proposed cryptosystem.

2.4 Conclusions

This chapter has provided the information of related theories to the proposed cryptographic approaches, including chaos theory, bifurcation, Lyapunov exponents, chaotic maps, random number generation, digital images, intensity histograms, Android, cryptography, number of pixel changing rate (NPCR), unified averaged changed density (UACI), and entropy. The literature reviews of 10 literatures related to the proposed cryptographic approaches were also included.

Chapter 3
Research Methodology

3.1 Introduction

This chapter describes research methodology of this thesis, involving research process, data collection, and research tools.

3.2 Research Process

3.2.1 Study the dynamics of Chaotic Maps with Absolute Value Nonlinearity.

3.2.2 Simulate properties of Chaotic Maps with Absolute Value Nonlinearity on MATLAB and Android, including time-domain waveforms, frequency-domain waveforms, bifurcation diagram, and Lyapunov exponents spectrum.

3.2.3 Design and create cryptographic systems for Android.

3.2.4 Collect and analyze results from designed cryptographic systems.

3.3 Data Collection

The data in this thesis is the set of results from the designed cryptographic systems of both encryption and decryption side. The data will be collected from each designed cryptographic system which uses the same set of plain images for encryption. The collected data from an image section will be analyzed for the performance of the design cryptographic system using pixel density histogram, image correlation, number of pixel change rate, and unified average changing density, and the entropy.

3.4 Research Tools

Research tools in thesis are MATLAB R2013a, Android Studio version 2.0, and Android Virtual Device.

3.5 Conclusions

This chapter has presented research methodology of this thesis, including research process data collection, and research tools.

Chapter 4
Experiment Results

4.1 Introduction

This fourth chapter proposes two cryptographic approaches. Results from each approach were analyzed with specified tools, including pixel density histograms, image correlation, number of pixel change rate, unified average change intensity, and information entropy. The analysis and simulations of the first and the second case of Chaotic Maps with Absolute Value Nonlinearity on MATLAB are described in this chapter. Simulations of the first case of Chaotic Maps with Absolute Value Nonlinearity on Android are also included.

4.2 Dynamics of Chaotic Maps with Absolute Value Nonlinearity

The implementation of any chaotic map into cryptographic systems requires the specified chaotic map to be analyzed and verified on its dynamics, in order to verify if the specified chaotic map is suitable for cryptography. In this thesis, the first case of Chaotic Maps with Absolute Value Nonlinearity is implemented on the first and the third proposed cryptographic approaches, and the second case is implemented on the second approach. The first case of Chaotic Maps with Absolute Value Nonlinearity is described mathematically as [7]

$$x_{n+1} = |1 - ax_n| \tag{2.4}$$

and the second case of Chaotic Maps with Absolute Value Nonlinearity is described as

$$x_{n+1} = |-1 + ax_n| \tag{2.5}$$

where a is the control parameter and x_n is the state variable of both first and second cases of Chaotic Maps with Absolute Value Nonlinearity.

4.2.1 Mathematical Analysis

The mathematical analysis of two chaotic maps which are to be implemented was accomplished using Jacobian matrix and fix points. The Jacobian matrix takes the first order partial derivatives of the chaotic map by state variables of the chaotic map as elements of the Jacobian matrix. The Jacobian matrix of the first case of Chaotic Maps with Absolute Value Nonlinearity was derived as

$$J = [-a \times sign(1 - ax)] \tag{4.1},$$

and the Jacobian matrix of the second case of Chaotic Maps with Absolute Value Nonlinearity was derived as

$$J = [a \times sign(-1 + ax)] \tag{4.2},$$

where J is the Jacobian matrix, a is the control parameter, and x is the state variable of the chaotic maps. For the fix points analysis, fix points of chaotic maps are acquired in the same manner as typical mathematic functions as

$$x = f(x) \tag{4.3},$$

where x is the state variable of the chaotic map and $f(x)$ is the chaotic map. From (4.3) the fix points of the first case of Chaotic Maps with Absolute Value Nonlinearity was acquired and described as

$$x = \frac{1}{(a \pm 1)} \tag{4.4},$$

and the fix points of the second case of Chaotic Maps with Absolute Value Nonlinearity was also acquired and described as

$$x = \frac{1}{(a \pm 1)} \tag{4.5},$$

where x is the state variable and a is the control parameter of the first and the second cases of chaotic maps.

4.2.2 Simulation on MATLAB

The first case and the second case of Chaotic Maps with Absolute Value Nonlinearity were simulated on MATLAB to inspect dynamics of chaotic maps which were going to be implemented to cryptographic systems. The simulation consists of bifurcation diagrams, Lyapunov Exponents (LE) spectrum, chaotic waveforms in time domain, and chaotic waveforms in frequency domain.

4.2.2.1 Bifurcation Diagrams

Bifurcation diagrams indicates possible values of the state variable in time domain for each value of the control parameter of chaotic maps. In this simulation, possible values of the state variable x from the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity were indicated as blue dots, and chaotic maps were computed for 150 iterations for each value of the control parameter. Figure 4.1 and figure 4.2 illustrate the bifurcation diagrams of the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity, respectively. It can be inspected from figure 4.1 and figure 4.2 that bifurcation diagrams of the first case and the second case were almost identically the same to each other. The significant number of blue dots which represents possible values of state variable x in both cases can be visualized where the value of the control parameter a of both cases was in the range of [1, 2]. Since blue dots in figure 4.1 and figure 4.2 represents possible values of the state variables x and the large number of possible values means dense periodic orbits, which is one of properties of chaotic systems, the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity become chaotic where the value of the control parameter a is in the range of [1, 2], in which the large number of blue dots are visible.

4.2.2.2 Lyapunov Exponent Spectrums

In order to measure the instability of chaotic maps quantitatively, the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity have also been simulated with Lyapunov exponent. State variable x of chaotic map (2.4) and

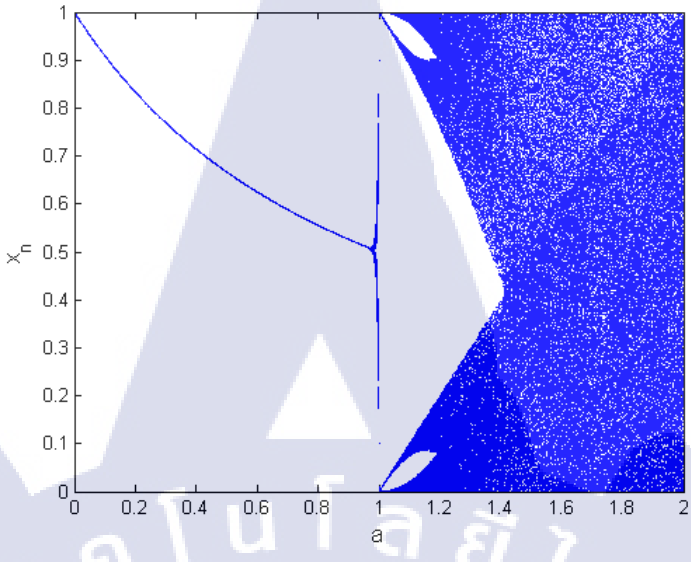


Figure 4.1 Bifurcation diagram of (2.4) on MATLAB.

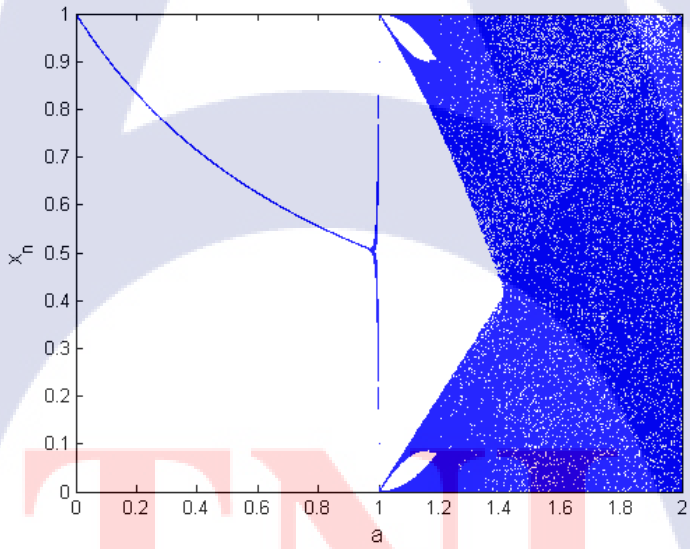


Figure 4.2 Bifurcation diagram of (2.5) on MATLAB.

(2.5) were used to calculate the Lyapunov exponents using the equation (2.1) which described mathematically as

$$\lambda = \lim_{n\to\infty}\left(\frac{1}{N}\right)\sum_{n=1}^{N} \ln f'(x_n) \qquad (2.1)$$

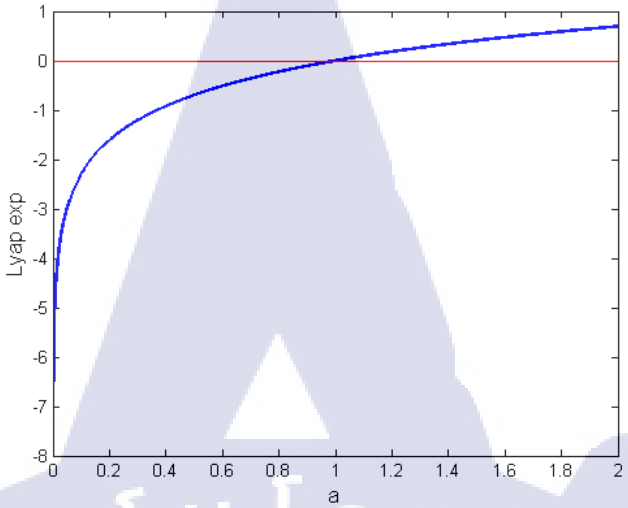


Figure 4.3 Lyapunov exponents spectrum of (2.4) on MATLAB.

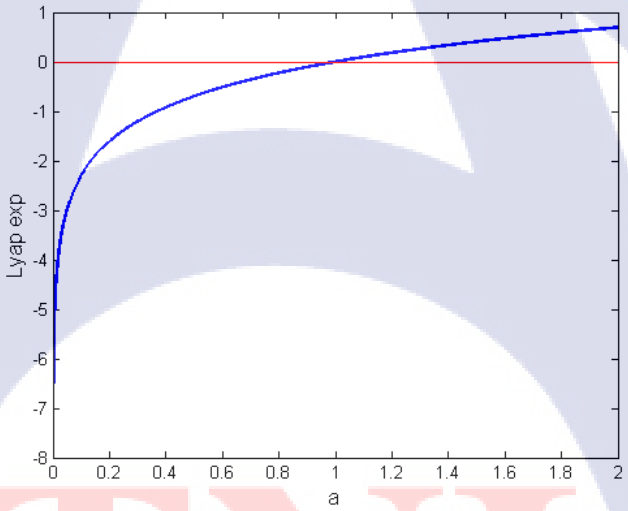


Figure 4.4 Lyapunov exponents spectrum of (2.5) on MATLAB.

where λ is the Lyapunov exponent, N is the number of iterations, x_n is the state variable, and f'(x_n) is the first derivative of the chaotic map. After the calculation, Lyapunov exponents which were results from (2.1) were plotted as spectrums illustrated in figure 4.3 and figure 4.4. From figure 4.3 and figure 4.4, the Lyapunov exponent spectrums of the first case and the second case are the same. Additionally, the value of Lyapunov exponent is more than zero where the value of the control parameter is in the range of [1, 2] in both figure 4.3 and figure 4.4, and tends to be higher as

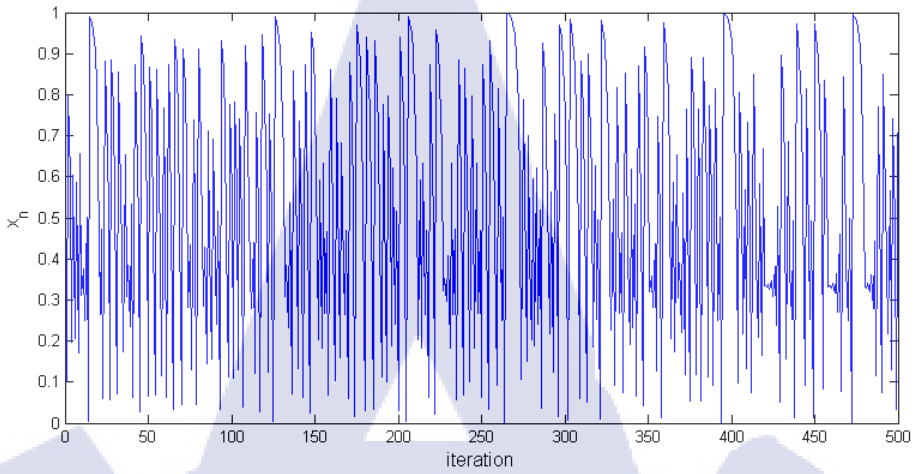


Figure 4.5 Chaotic waveforms in time domain of (2.4), initial condition as 0.1.

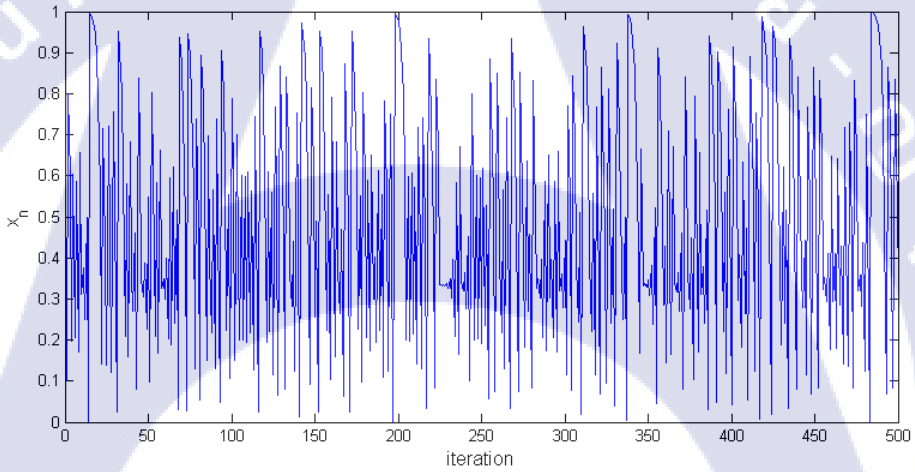


Figure 4.6 Chaotic waveforms in time domain of (2.4), initial condition as 0.100001.

the value of the control parameter is closer to 2. Since the Lyapunov exponent value of the system that more than zero means the system is chaotic, and the more the Lyapunov exponent value is, the more chaotic the system will be, it can be summarized that the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity become chaotic where the value of the control parameter is in the range of [1, 2]. Comparing the Lyapunov exponent spectrums to the bifurcation diagrams of the first case and the second case, it can be seen that the first case and the second case become chaotic when the control parameter value is in the range of [1, 2], as the number

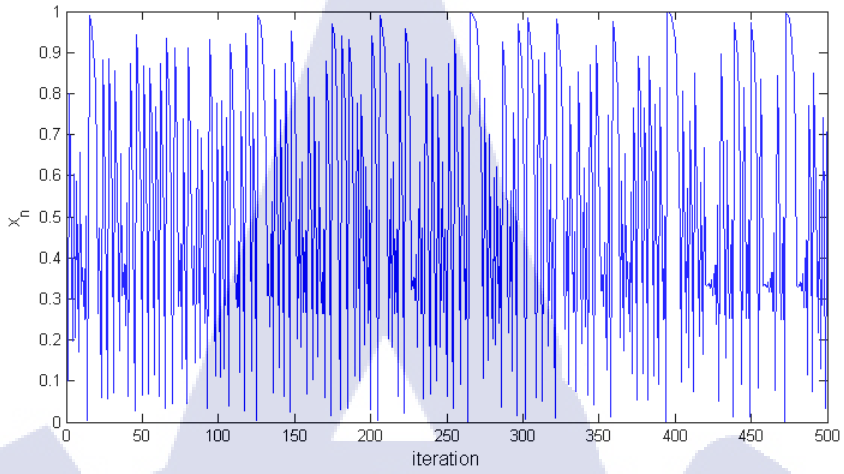


Figure 4.7 Chaotic waveforms in time domain of (2.5), initial condition as 0.1.

of possible values are high in the bifurcation diagrams and the values of the Lyapunov Exponent are positive in this region. Moreover, bifurcation diagrams and Lyapunov exponents of Chaotic Maps with absolute nonlinearity contain no periodic windows, compared to the bifurcation diagrams and Lyapunov exponents of the Logistic map in figure 2.1 and figure 2.2 which contain several periodic windows. Therefore, dynamics of Chaotic Maps with absolute nonlinearity are more robust than Logistic map.

4.2.2.3 Time-Domain Chaotic Waveforms

The time-domain chaotic waveform simulation consists of computing the chaotic maps for the certain iterations, and plotting the computed state variables from each iteration of chaotic maps as waveforms. In this thesis, the chaotic maps were computed for 500 iterations, and the value of the control parameter a is 1.999. Figure 4.5 and figure 4.6 shows the chaotic waveforms in time domain of the first case of Chaotic Maps with Absolute Value Nonlinearity. Figure 4.5 shows the chaotic waveforms in time domain where the initial condition of the state variable x is 0.1, and figure 4.6 shows the chaotic waveforms in time domain where the initial condition of the state variable x is 0.100001. It can be noticed from figure 4.5 and figure 4.6 that, considerable differences in chaotic waveforms in time domain were occurred, despite the small difference in the state variable of the chaotic map. Similar to the first case, the second case of Chaotic Maps with Absolute Value Nonlinearity was also

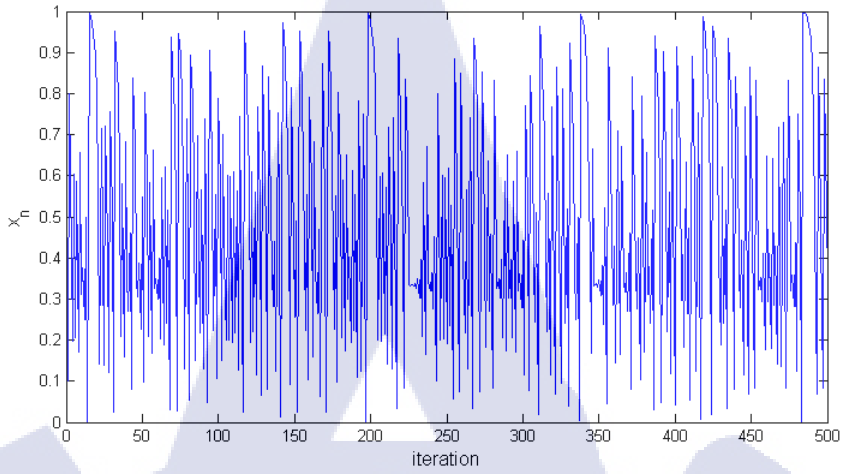


Figure 4.8 Chaotic waveforms in time domain of (2.5), initial condition as 0.100001.

simulated through the time-domain chaotic waveforms. Figure 4.7 shows the chaotic waveforms in time domain of the second case where the state variable *x* is 0.1, and figure 4.8 shows the chaotic waveforms in the domain of the second case where the state variable *x* is 0.100001. As shown in figure 4.7 and figure 4.8, the chaotic waveforms in time domain of the second case also yield the same characteristics as waveforms of the first case, in which the significant differences were visualized despite the slight difference in the initial condition of the state variable *x*.

4.2.2.4 Frequency-Domain Chaotic Waveforms

It is advisable to simulate on the frequency domain after the chaotic waveforms in time domain were acquired, due to the fact that, even the waveforms in time domain have their appearances as chaotic waveforms, it is possible to be simple periodic signals if there are significant frequencies appeared in the frequency domain. In this thesis, the frequency-domain chaotic waveforms were obtained from the chaotic waveforms in time domain, which have their chaotic maps computed for 500 iterations, using 0.1 as the initial condition of the state variable, and have 1.999 as value of the control parameter. Additionally, the method of transforming the chaotic waveforms in time domain into the frequency domain is the Fast Fourier Transforms (FFT), which use the sampling frequency value as 2048. Figure 4.9 shows the chaotic waveforms in frequency domain of the first case, and figure 4.10 shows the chaotic waveforms in

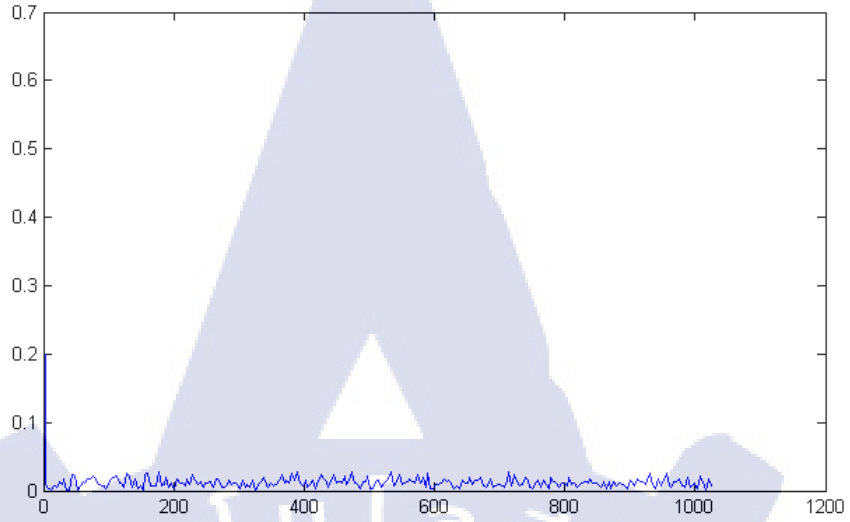


Figure 4.9 Chaotic waveforms in frequency domain of (2.4) on MATLAB

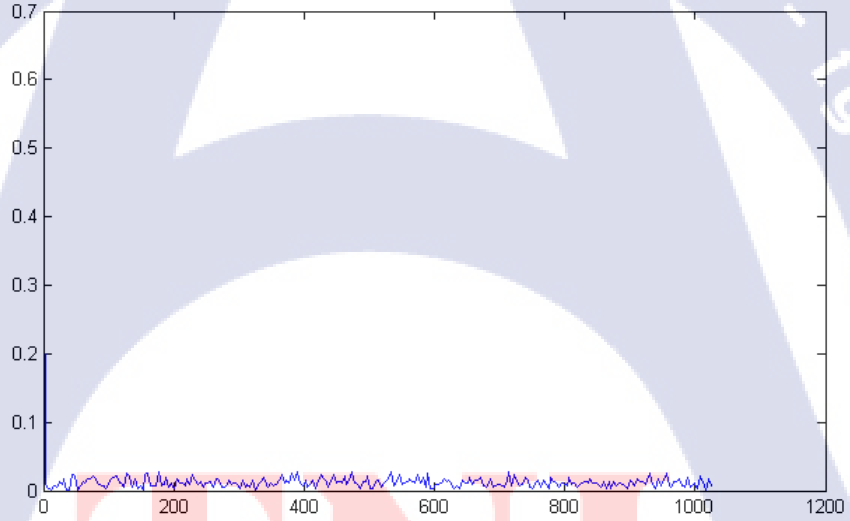


Figure 4.10 Chaotic waveforms in frequency domain of (2.5) on MATLAB

frequency domain of the second case of Chaotic Maps with Absolute Value Nonlinearity. From figure 4. And figure 4.10, there is no significant frequency appeared in the waveforms. However, the frequency numbers on the x axis of the waveforms are completely depend on the value of sampling frequency, due to the fact that chaotic maps are iterative functions, in which calculations of chaotic waveforms of chaotic maps in time domain do not involve frequencies.

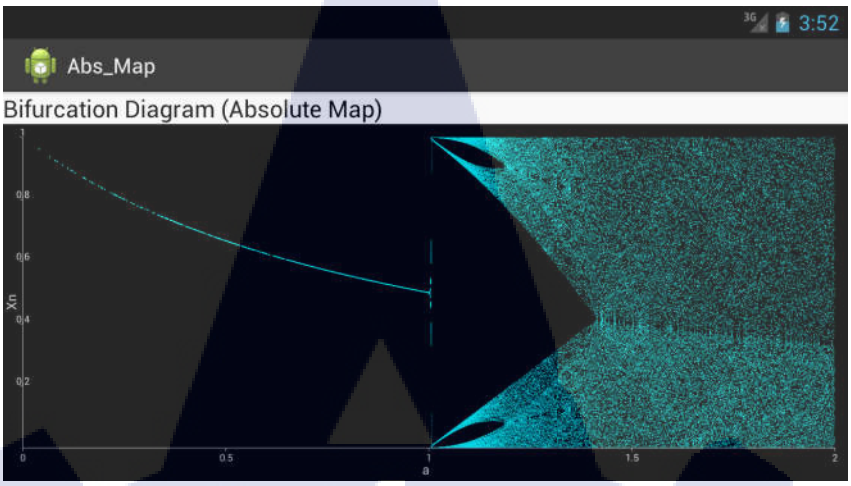


Figure 4.11 Bifurcation diagram of (2.4) on Android.

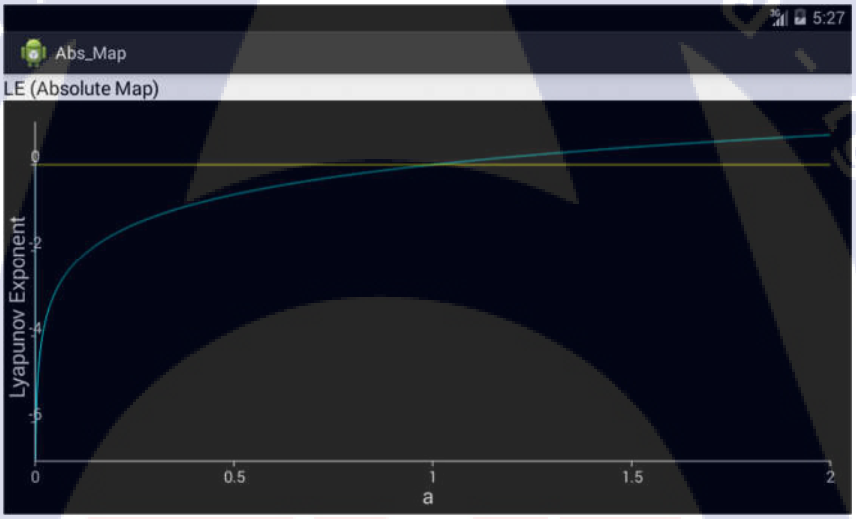


Figure 4.12 Lyapunov exponents spectrum of (2.4) on Android.

4.3 Simulations of Chaotic Maps with Absolute Value Nonlinearity on Android

After the simulations on MATLAB have been completed, the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity were simulated on Android. The reason behind the simulations on Android is that the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity have to be implemented on cryptographic systems on Android. The simulations consist of time-domain chaotic

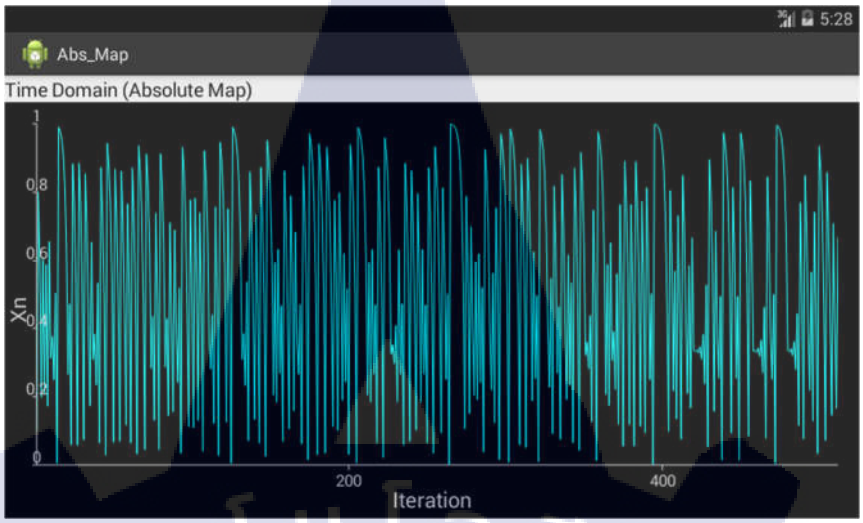


Figure 4.13 Chaotic waveforms in time domain of (2.4) on Android, using the initial condition as 0.1

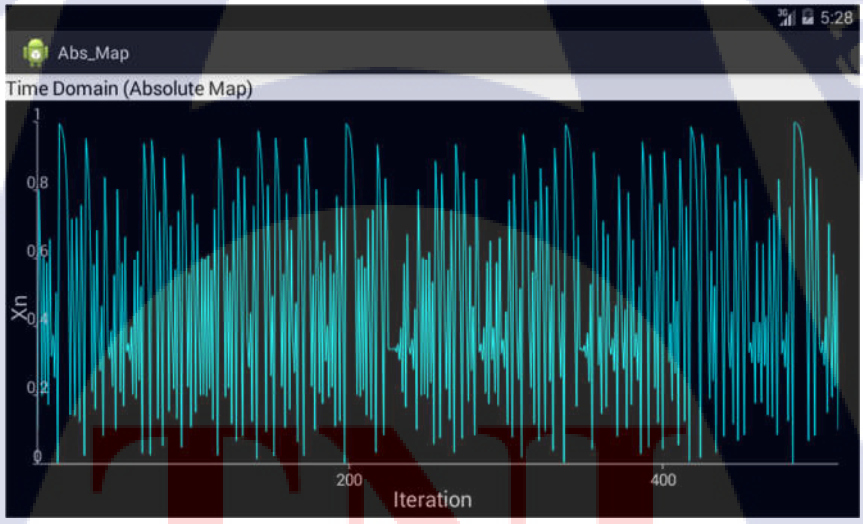


Figure 4.14 Chaotic waveforms in time domain of (2.4) on Android, using the initial condition as 0.100001

waveforms, frequency-domain chaotic waveforms, bifurcation diagram, and Lyapunov exponents on emulated Android device. However, only the first case of Chaotic Maps with Absolute Value Nonlinearity was simulated in this thesis, due to the results from the simulations on MATLAB, in which the first case and the second case yield almost identically the same results in all methods of simulations completed on MATLAB.

Figure 4.15 Chaotic waveforms in frequency domain of (2.4) on Android

4.3.1 Bifurcation Diagrams on Android

Figure 4.11 shows the bifurcation diagram of the first case of Chaotic Maps with Absolute Value Nonlinearity on Android. Similar to the bifurcation diagram simulated on MATLAB shown in figure 4.1, the bifurcation diagram on Android shown in figure 4.11 also had dense possible values where the value of the control parameter a is in the range of [1, 2], since the large number of dots which represent can also be visible in this region. From the bifurcation diagram on figure 4.11, it can be concluded that the first case of Chaotic Maps with Absolute Value Nonlinearity also become chaotic where the value of the control parameter a is in the range of [1, 2] on Android.

4.3.2 Lyapunov Exponents Spectrums on Android

The Lyapunov exponents spectrum of the first case of Chaotic Maps with Absolute Value Nonlinearity on Android is shown in figure 4.12. From the Lyapunov exponents spectrum in figure 4.12, the Lyapunov exponent on Android gave the same results as the Lyapunov exponent simulated on MATLAB shown in figure 4.3, in which the values of Lyapunov exponent are more than zero where the value of the control parameter is in the range of [1, 2]. Since the value of the Lyapunov exponents that more than zero refer to chaotic behaviors in the system, it can be concluded from the

Lyapunov exponents spectrum that, the first case of Chaotic Maps with Absolute Value Nonlinearity performs chaotic behaviors on Android where the value of the control parameter is in the range of [1, 2].

4.3.3 Time-Domain Chaotic Waveforms

The first case of Chaotic Maps with Absolute Value Nonlinearity was also simulated through time-domain chaotic waveforms generation on Android. However, the chaotic waveforms simulation on time domain on Android was altered from the simulation on MATLAB, in which the iteration number was increased from 500 iterations to 512 iterations, while initial conditions of the state variable and the value of the control parameter were the same as the simulation on MATLAB. Figure 4.13 and figure 4.14 shows the chaotic waveforms in time-domain of the first case of Chaotic Maps with Absolute Value Nonlinearity on Android, where the initial condition of the state variable is 0.1 and 0.100001, respectively. From figure 4.13 and figure 4.14, there are significant differences from the chaotic waveforms which generated from the first case of the chaotic maps that used initial conditions which are differed at only 0.000001. Thus, this first case of the chaotic maps is highly sensitive to the initial condition.

4.3.4 Frequency-Domain Chaotic Waveforms

The chaotic waveforms in frequency domain simulation of the first case of Chaotic Maps consists of the generation of the chaotic waveforms in time domain and the transformation of the generated chaotic waveforms in to the frequency domain, similar to the simulation on MATLAB. However, there is no native function on Android or Java to handle the Fast Fourier Transform, which is required for the transformation of chaotic waveforms from time domain into frequency domain. Therefore, JTransforms, an external java library has been included to the simulation to handle the transformation into frequency domain. The first case of the chaotic maps was operated for 512 iterations using the initial condition as 0.1, and the control parameter as 1.999, to generate chaotic waveforms in time domain which is to be transforms into frequency domain. Figure 4.15 illustrates the chaotic waveforms in frequency of the first case of the chaotic maps on Android. From the chaotic waveforms in frequency domain, there is no significant frequency to be perceptible, similar to the simulation on MATLAB.

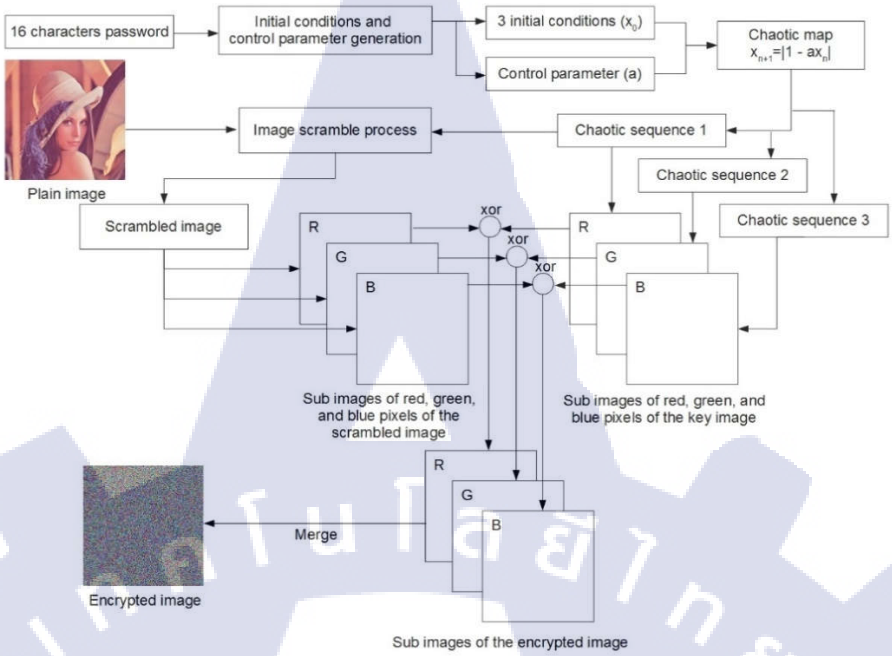


Figure 4.16 Block diagram of the encryption process of the first proposed cryptographic approach.

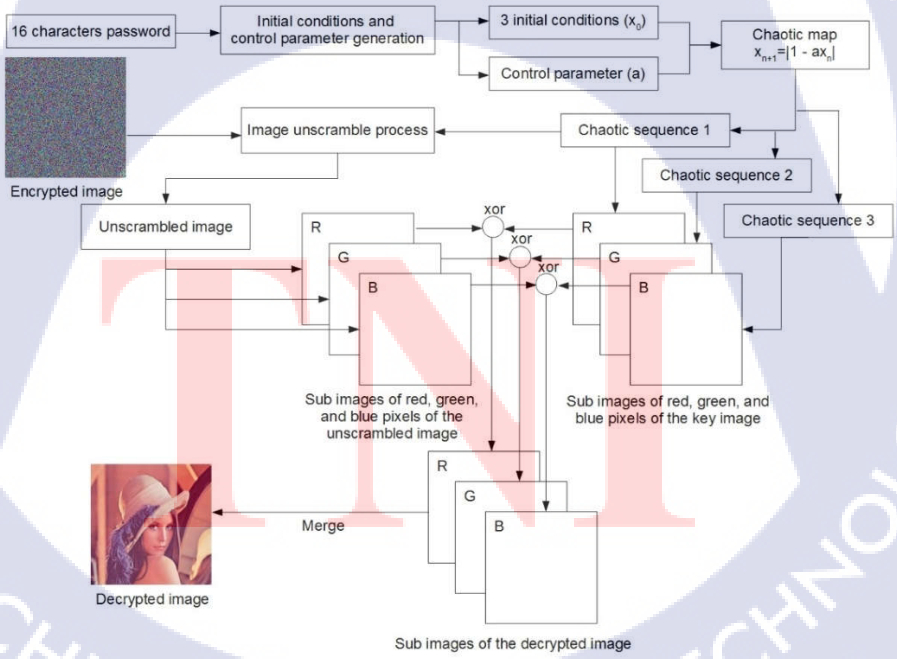


Figure 4.17 Block diagram of the decryption process of the first proposed cryptographic approach.

4.4 Proposed Cryptographic Approach I

The first proposed cryptographic approach is the cryptographic approach for images on Android. This first proposed cryptographic approach utilized the first case of Chaotic Maps with Absolute Value Nonlinearity for both encryption and decryption process of cryptography. Figure 4.16 illustrates the block diagram of the encryption process of the first proposed cryptographic approach, and figure 4.17 illustrates the block diagram of the decryption process of the proposed cryptographic approach.

4.4.1 Encryption and Decryption Process of the First Approach

The encryption process as described in figure 4.16 starts by the acquisition of the plain image and the password which is the combination of 16 alphanumeric characters. The password of 16 characters is separated into 3 sets of 6 characters, then the separated sets of characters are used to generate 3 initial conditions and a control parameter value to be used in the chaotic map, as shown in the block diagram in figure 4.18. Generated initial conditions and control parameter are utilized in the first case of the chaotic maps to generate three chaotic sequences. Each chaotic sequence generated from the first case of Chaotic Maps with Absolute Value Nonlinearity using different initial conditions, i.e. the first sequence used the first generated initial condition, second sequence used the second initial condition, and so on. However, the same value of the control parameter and the number of iterations, which equals to the number of pixels are used in the generation of each chaotic sequence. The generated chaotic sequences are used to create an image of the same size as the plain image, which will be used as a key, by set each pixel in each color plane with numbers in chaotic sequence multiplied by 255, due to the fact that digital images typically have 255 as maximum intensities. The first sequence is used to generate red color plane, the second sequence is used to generate green color plane, and the third sequence is used to generate blue color plane of the image key. In the same time, the first generated chaotic sequence is also used to scramble the plain image by swapping the intensity of the pixel positioned at pixel number p to the pixel designated at value of the element p in the chaotic sequence multiplied by the number of pixel of the plain image, starting from the first pixel on top-leftmost of the image. After the plain image was scrambled, the scrambled image is separated into 3 sub images, which each sub image contains pixels of one color plane,

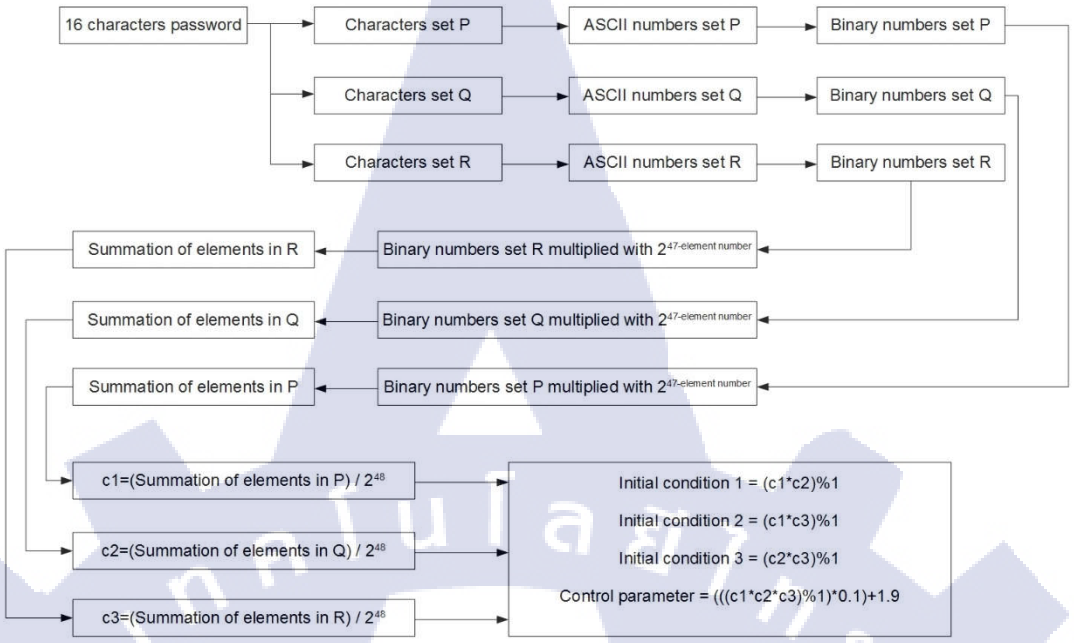


Figure 4.18 Block diagram of the initial condition and the control parameter generation process of the first proposed cryptographic approach.

i.e. red, green, and blue color plane. Each sub image has its pixels operated with pixels of sub image of the image key of the same color plane using XOR operation, where sub image in red color plane has its pixel intensities XOR with the intensities of the image key at the same position, i.e. intensity of the pixel on the top-leftmost of the scrambled image XOR with the intensity of the pixel on the top-leftmost of the image key. After XOR operation was completed, sub images that passed through the XOR operation are merged into the encrypted image.

The decryption process of the first proposed approach has similar process to the encryption process. From figure 4.17, each block of the decryption process is similar to blocks of the encryption process in figure 4.16. However, there are differences that observable. The first difference in the decryption process is the input of the process, where the decryption process acquires the encrypted image as its input. The second difference is the process of scrambling which is similar to the scramble process in the encryption process, in which pixels swap their intensities, though unscrambling process has different starting point of swapping from the scrambling process, by starting at the last pixel on bottom-rightmost of the image, instead of the top-leftmost pixel.

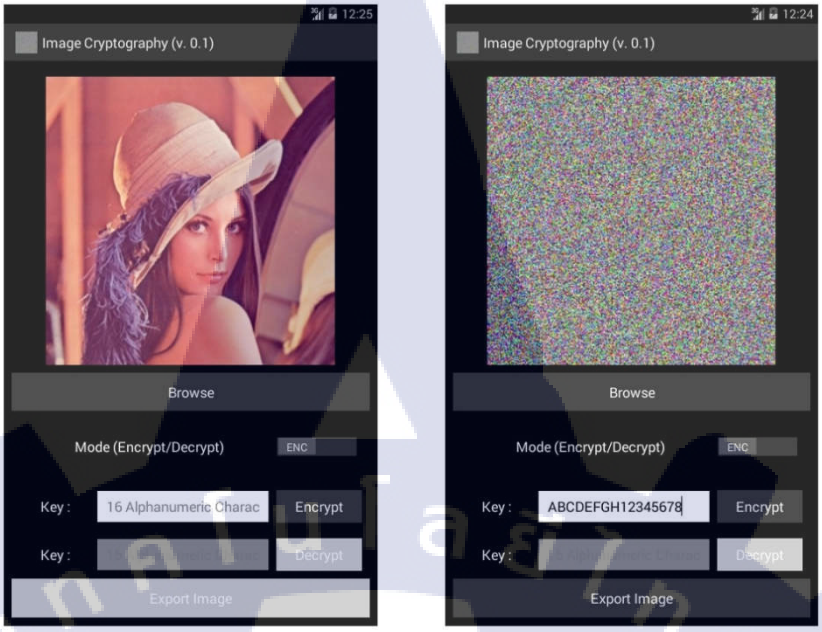


Figure 4.19 Results from the encryption process of the first approach, (a) the plain image before encryption, and (b) the encrypted image.

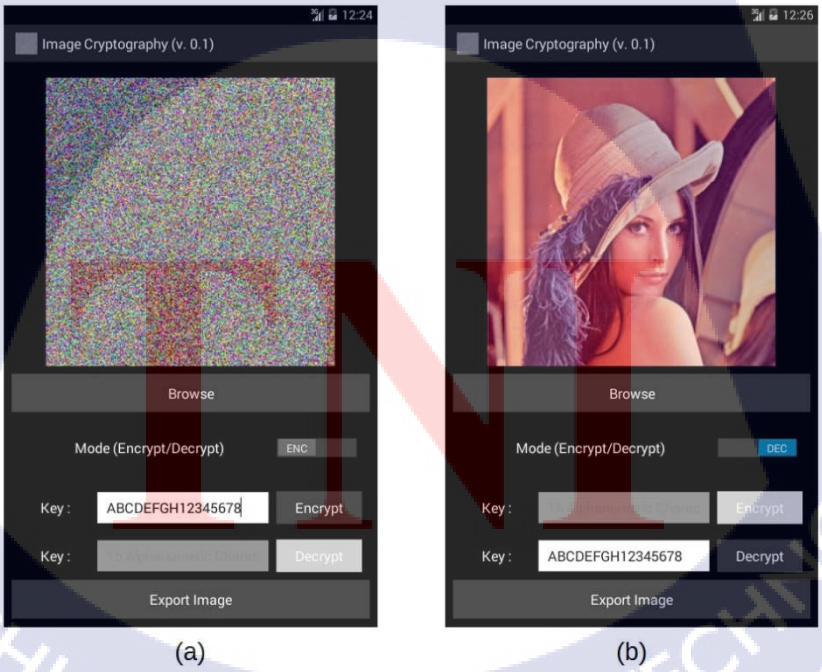


Figure 4.20 Results from the decryption process using the correct password of the first approach, (a) the encrypted image, and (b) the decrypted image.

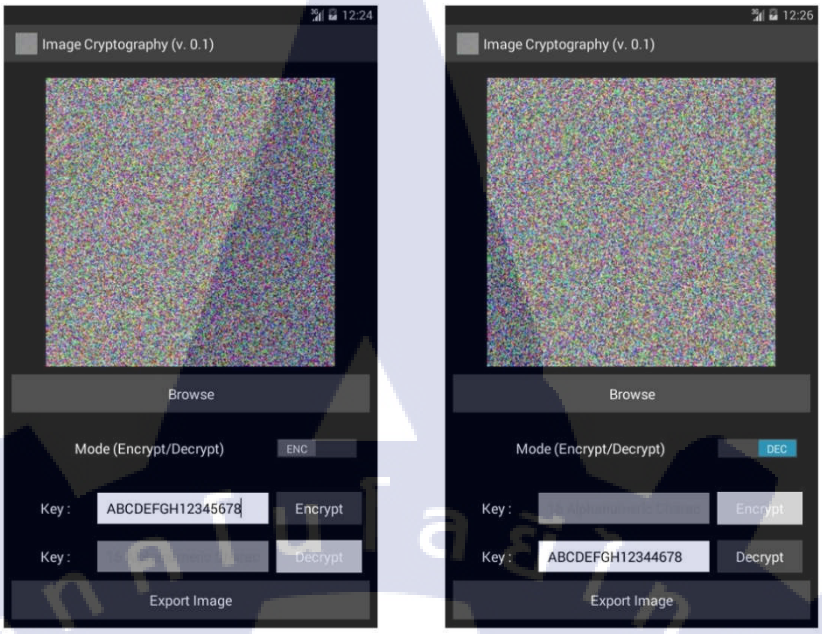


Figure 4.21 Results from the decryption process using the wrong password of the first approach, (a) the encrypted image, and (b) the decrypted image.

4.4.2 Results of the First Approach from Android Emulator

Results from the Android application which implemented the first proposed cryptographic approach are displayed in figure 4.19, figure 4.20, and figure 4.21. Figure 4.19 illustrates the results of the encryption process, where figure 4.20 (a) shows the plain image before the encryption, and figure 4.19 (b) shows the encrypted image. Figure 4.20 illustrates the results from the decryption process which used the same password as the encryption process, where figure 4.20 (a) shows the encrypted image, and figure 4.20 (b) shows the decrypted image which is the same as the plain image in figure 4.19 (a). Figure 4.21 illustrates the results from the decryption process which used the password which is only one character different from the key used in the encryption process, where figure 4.21 (a) shows the encrypted image, and figure 4.21 (b) shows the decrypted image which is scrambled and contains no evidence of the plain image in figure 4.19 (a). Table 4.1 depicts the average computation time of the encryption process and the decryption process on emulated Android device, using the LENA image at the size of 256×256 and 512×512 as test images. From table 4.1, the

Table 4.1 Average computation time of encryption process and decryption process of the first proposed cryptographic approach.

Image size	Encryption time (ms.)	Decryption time (ms.)
256×256	272	271.8
512×512	884	870.8

computation time is less than half of a second for the image at the size of 256×256, and the computation time rises as the image size increased, to approximately 1 second of the image at the size of 512×512, while the computation time of the decryption process is slightly lower than the encryption process.

4.4.3 Analysis of Results from Encryption Process of the First Approach

The results from the encryption process of the first proposed cryptographic approach were analyzed on MATLAB. There are two types of the analysis in this thesis. The first type is the qualitative analysis which has no numeric result from the analysis. On the other hand, the second type is the quantitative analysis which results from the analysis are numeric. Two images at the size of 256×256 were used as the plain images in the analysis. The first image for the experiments is the LENA image as shown in figure 4.22, and the second image is the Monkey image as shown in figure 4.23.

4.4.3.1 Qualitative Analysis of Proposed Cryptographic Approach I

The qualitative analysis of the results from the first proposed cryptographic approach consists of the intensity histogram of the image and the correlation plot of pixels in the image. Figure 4.24 shows the intensity histogram of the plain image compared to the encrypted image, using the LENA image as the plain image, where figure 4.24 (b), (c), and (d) show intensity histograms of the plain image in red, green, and blue color plane, respectively. Furthermore, figure 4.24 (f), (g), and (h) show intensity histogram of the encrypted image in red, green, and blue color plane, respectively. Figure 4.25 shows the intensity histogram, using the Monkey image as the plain image, where figure 4.25 (b), (c), and (d) show intensity histograms of the plain image in red, green, and blue color plane, respectively. Furthermore, figure 4.25 (f),

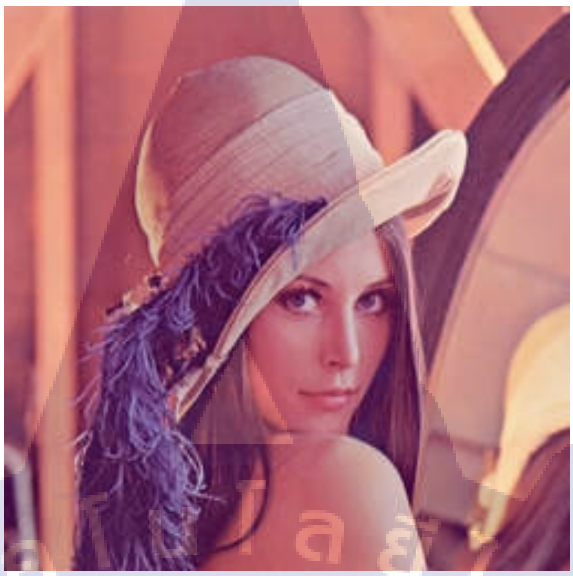


Figure 4.22 The LENA image.

Figure 4.23 The Monkey image.

(g), and (h) show intensity histogram of the encrypted image in red, green, and blue color plane, respectively. From figure 4.24 and figure 4.25, encrypted images have significant differences in the intensity histograms of the images, compared to the intensity histograms of plain images. For intensity histograms of all color planes,

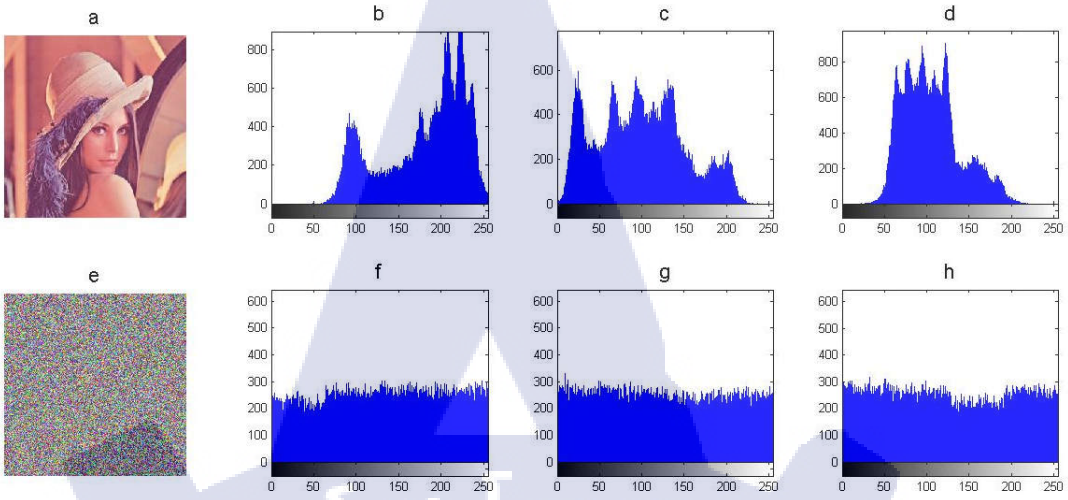


Figure 4.24 Intensity histogram of the LENA image and its encrypted image from the first proposed cryptographic approach.

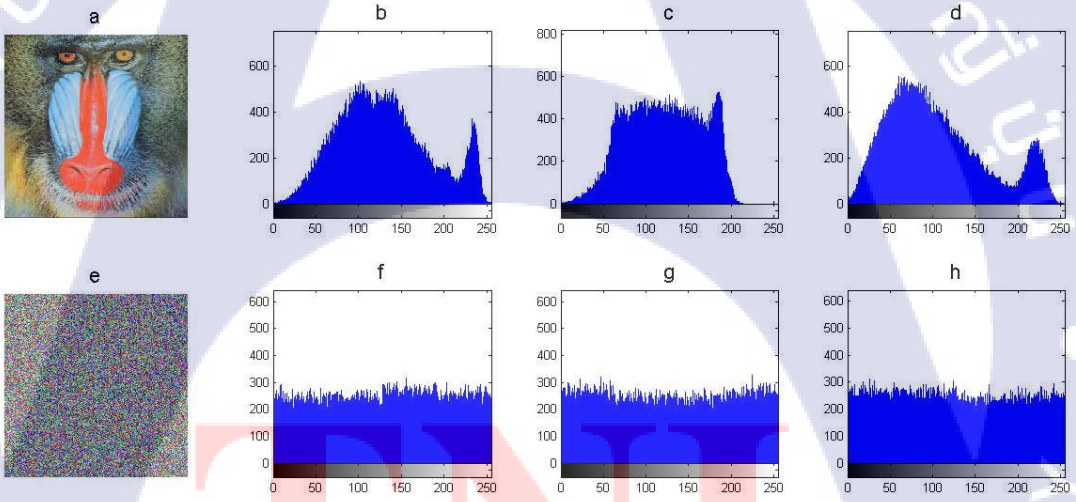


Figure 4.25 Intensity histogram of the Monkey image and its encrypted image from the first proposed cryptographic approach.

histograms of encrypted images are uniformly distributed, which means all intensities of the encrypted images have almost the same probability to be appeared in encrypted images. The uniformly distributed intensity histograms of the image also mean the image contains no significant characteristics of intensities of the image, which can be used to determine the plain image or the encryption key in some cases. Figure 4.26 and

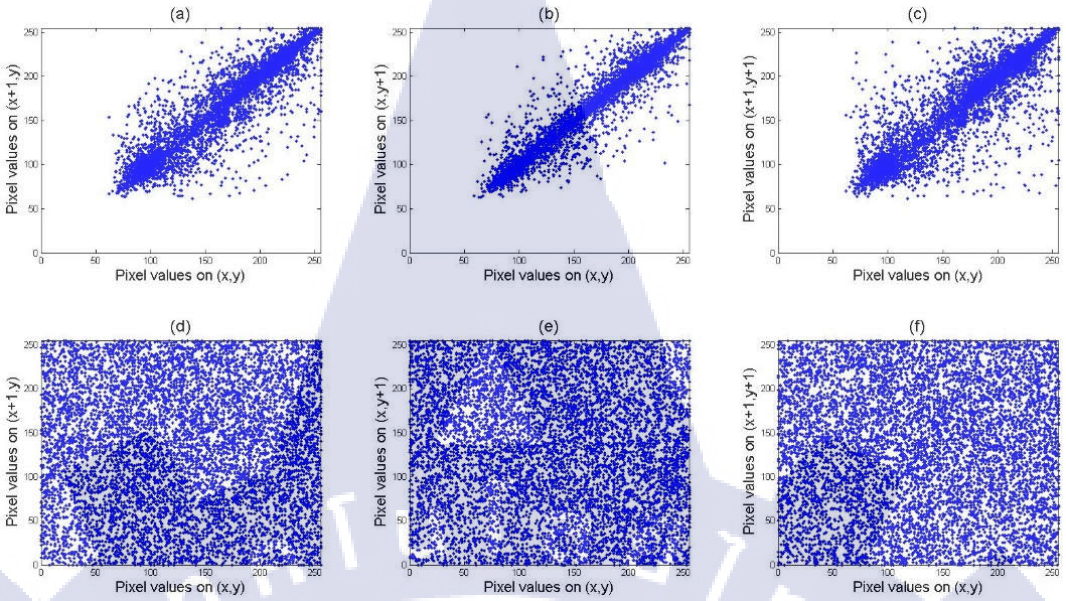


Figure 4.26 Correlation plots of pixels in plain image and encrypted image in red color plane from the first approach, using the LENA image as the plain image.

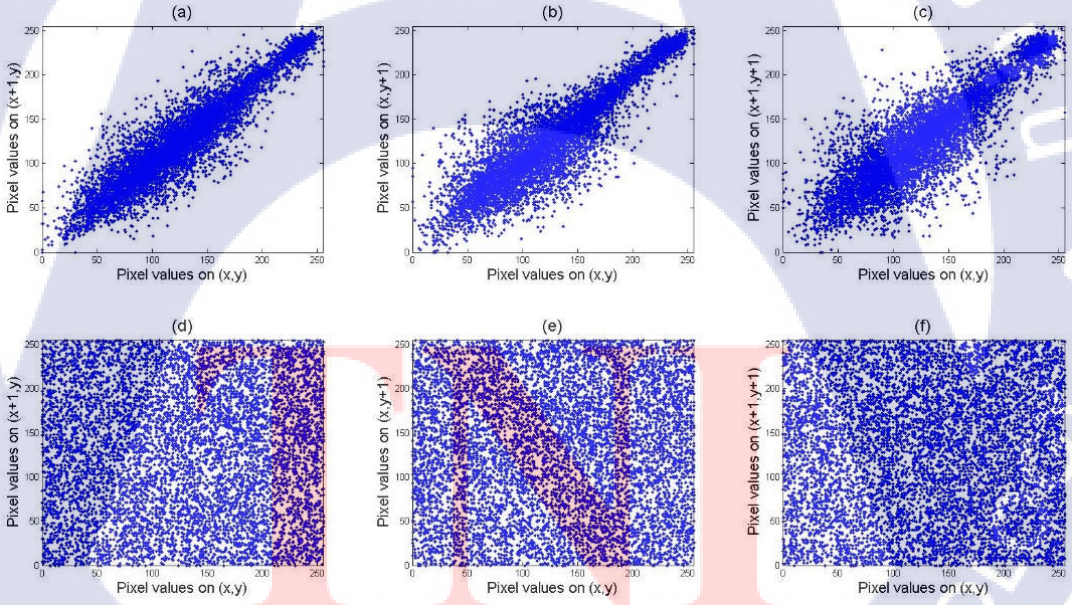


Figure 4.27 Correlation plots of pixels in plain image and encrypted image in red color plane from the first approach, using the Monkey image as the plain image.

figure 4.27 shows the correlation plots of pixels in plain images and encrypted images in red color plane. Figure 4.26 shows correlation plots of pixels of plain image and

encrypted image, using the LENA image in figure 4.22 as the plain image. Figure 4.26 (a), (b), and (c) show the correlation plots of pixels in the plain image with their adjacent in horizontal, vertical, and diagonal directions, respectively, while figure 4.26 (d), (e), and (f) show the correlation plots of pixels in the encrypted image with their adjacent in horizontal, vertical, and diagonal directions, respectively. Figure 4.27 shows correlation plots of pixels in the plain image and the encrypted image, using the Monkey image in figure 4.23 as the plain image. Figure 4.27 (a), (b), and (c) show the correlation plots of pixels in the plain image with their adjacent in horizontal, vertical, and diagonal directions, respectively, while figure 4.27 (d), (e), and (f) show the correlation plots of pixels in the encrypted image with their adjacent in horizontal, vertical, and diagonal directions, respectively. From correlation plots of pixels in figure 4.26 and figure 4.27, the correlation plots of the plain images shows the similarity of each pixel intensity to the intensity of the adjacent pixels, which can be inspected from the pixel intensities that are linearly close to others. On the other hand, correlation plots of the encrypted image shows the scattered intensity of each pixel compared to the intensity of the adjacent pixels. The scattered correlation plots of pixels in the encrypted images show that each pixel in the encrypted image is different to its adjacent, and has no relation to be implied as the trace of the plain image.

4.4.3.2 Quantitative Analysis of Proposed Cryptographic Approach I

The quantitative analysis of the results from the encryption process of the first proposed cryptographic approach consists of image correlation coefficients, number pixel changing rate (NPCR), unified averaged changed intensity (UACI), and the information entropy. Table 4.2 depicts the correlation coefficients which acquired by comparing the plain image in each color plane to all color plane of the encrypted image. Correlation coefficients of both LENA image and the Monkey image compared to their encrypted images, are relatively close to zero, which is the ideal value for two images that are completely different. Table 4.3 shows the NPCR and UACI value acquired by comparing two encrypted images which have the same plain image that have the difference only one pixel, encrypted with the same key. NPCR and UACI of the encrypted images are relatively close to zero percent for both LENA image and the Monkey image. Low NPCR and UACI means slight differences in the plain image does

Table 4.2 Correlation coefficients the plain images and the encrypted images from the first proposed cryptographic approach.

Correlation coefficients (C)	LENA image	Monkey image
$C_{red\text{-}red}$	0.0015	0.0001
$C_{red\text{-}green}$	0.0059	0.0022
$C_{red\text{-}blue}$	0.0009	0.0010
$C_{green\text{-}red}$	0.0039	0.0024
$C_{green\text{-}green}$	0.0067	-0.0055
$C_{green\text{-}blue}$	0.0035	0.0071
$C_{blue\text{-}red}$	0.0045	0.0019
$C_{blue\text{-}green}$	0.0080	-0.0050
$C_{blue\text{-}blue}$	0.0055	0.0044

Table 4.3 NPCR and UACI in percentage of the encrypted images from the first proposed cryptographic approach.

NPCR/UACI	LENA image	Monkey image
$NPCR_{red}$	0.0015	0.0015
$NPCR_{green}$	0.0015	0.0015
$NPCR_{blue}$	0.0015	0.0015
$UACI_{red}$	0.0006	0.0002
$UACI_{green}$	0.0007	0.0003
$UACI_{blue}$	0.0001	0.0008

Table 4.4 Information entropy of the encrypted images from the first proposed cryptographic approach.

Entropy (H)	LENA image	Monkey image
H_{red}	7.9935	7.9956
H_{green}	7.9955	7.9950
H_{blue}	7.9921	7.9959

not affect the outcomes of the encryption which utilized the same key. Consequently, the first proposed cryptographic approach is weak against the differential attacks, that may use similar plain images to get the key of the encryption, or in the worst case, get the algorithm of the first proposed cryptographic approach. Table 4.4 shows the information entropy value of red, green, and blue color planes of the encrypted images which each has its plain image as the LENA image and the Monkey image. The entropy values are calculated from equation (2.x) described in chapter 2. Entropy values of the encrypted images in all color planes of the encrypted images in table 4.4 are close to 8, which is an ideal value of images which intensities have same probability in the image.

4.5 Proposed Cryptographic Approach II

The second proposed cryptographic approach is also implemented on Android, similar to the first approach. On the contrary to the first approach, the second case of Chaotic Maps with Absolute Value Nonlinearity is implemented in the second proposed cryptographic approach instead. Due to the weakness of the first proposed cryptographic approach against the differential attacks measured by NPCR and UACI, the second approach implemented were modified to overcome this weakness. Figure 4.28 shows the block diagram of the encryption process of the second approach, and figure 4.29 shows the block diagram of the decryption process of the second approach.

4.5.1 Encryption and Decryption Process of the Second Approach

The encryption process as described in figure 4.28 starts by the acquisition of the plain image and the password which is the combination of 16 alphanumeric characters, same as the encryption process of the first approach. The acquired password is separated into 3 sets of 6 characters, and average intensities of the plain image in red, green, and blue color plane are calculated. The average intensities of the plain image and sets of separated password characters are used to generate the initial condition and the control parameter to be used in the chaotic map, as the block diagram shown in figure 4.30. The generated initial condition and control parameter are used to generate the chaotic sequence, using the iteration number as the number of pixels in the plain image plus 2. The chaotic sequence generated from the chaotic map is used to scramble the plain image, using the same method as the first approach. The scrambled plain

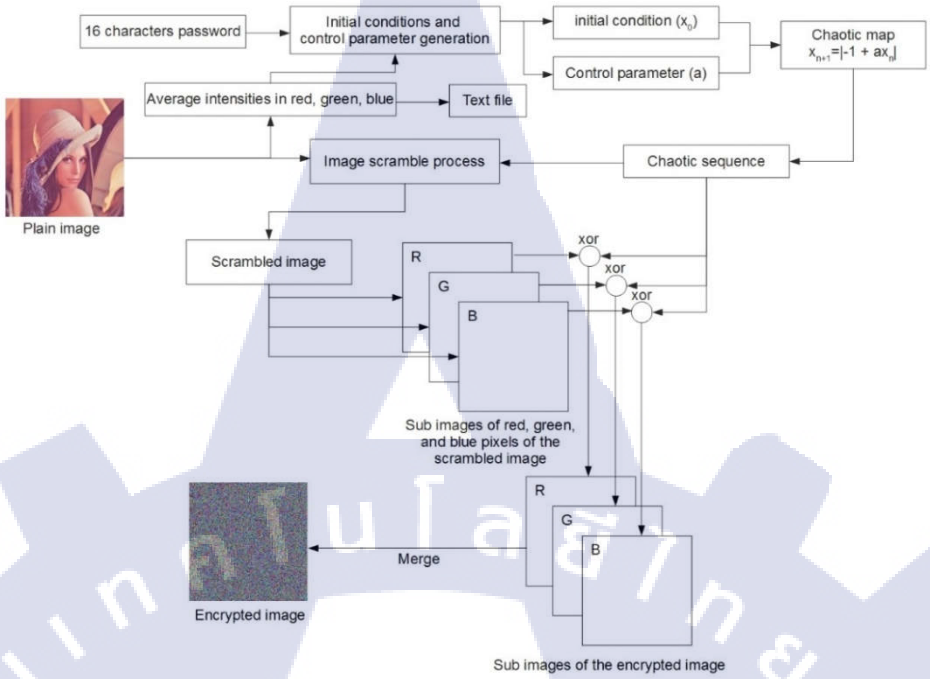


Figure 4.28 Block diagram of the encryption process of the second proposed cryptographic approach.

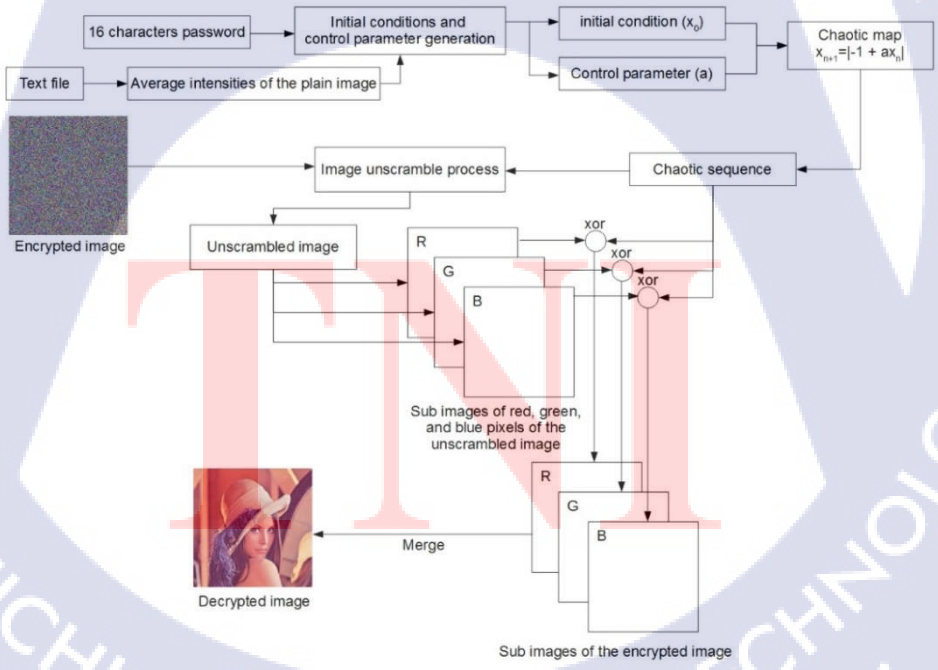


Figure 4.29 Block diagram of the decryption process of the second proposed cryptographic approach.

image is then operated with the generated chaotic sequence multiplied with 255 through XOR operation, in which the color planes of the scrambled image are operated with the chaotic sequence by element shifting method. In details, the red color plain XOR each pixel with each element in the chaotic sequence starting from the first element of the chaotic sequence which XOR with the first pixel on the top-leftmost of the image, and ends by XOR the element of the chaotic sequence which has element number equals to the number of pixels of the plain image with the last pixel on the bottom-rightmost of the image. For the green color plane, the process starts by XOR the second element of the chaotic sequence with the first pixel, and ends by XOR the last pixel with the element of the chaotic sequence at number equals to the number of pixels of the plain image plus 1. For the blue color plane, the process starts by operating the XOR operation the first pixel with the third element of the chaotic sequence, and ends by XOR the last pixel with the element of the chaotic sequence at number equals to the number of pixels of the plain image plus 2. After the XOR operation is completed, the sub images which acquired from the XOR operation of color planes with the chaotic sequence are merged into the encrypted image.

The decryption in figure 4.29 is similar to the encryption process, in which the process also consists of the XOR operation, which is the same as the encryption process. However, the decryption process requires three elements to complete the decryption, which are the encrypted image, password consists of 16 alphanumeric characters, and the text file which contains the average intensities of color planes of the plain image. The average intensities of the plain images are extracted from the text file and then used to generate the initial condition and control parameter using the same method as in the encryption process, displayed in figure 4.30. The generated initial condition and control parameter are then used to generate the chaotic sequence required for the decryption process in the same method as described the encryption process. The generated chaotic sequence is used to unscramble the acquired encrypted image, same as the unscramble process of the first approach in section 4.4.1. Sub images of the unscrambled image of red, green, and blue color planes are then operated with the generated chaotic sequence through the XOR operation in the same manner as the encryption process. Sub images which processed through the XOR operation are then merged into the decrypted image.

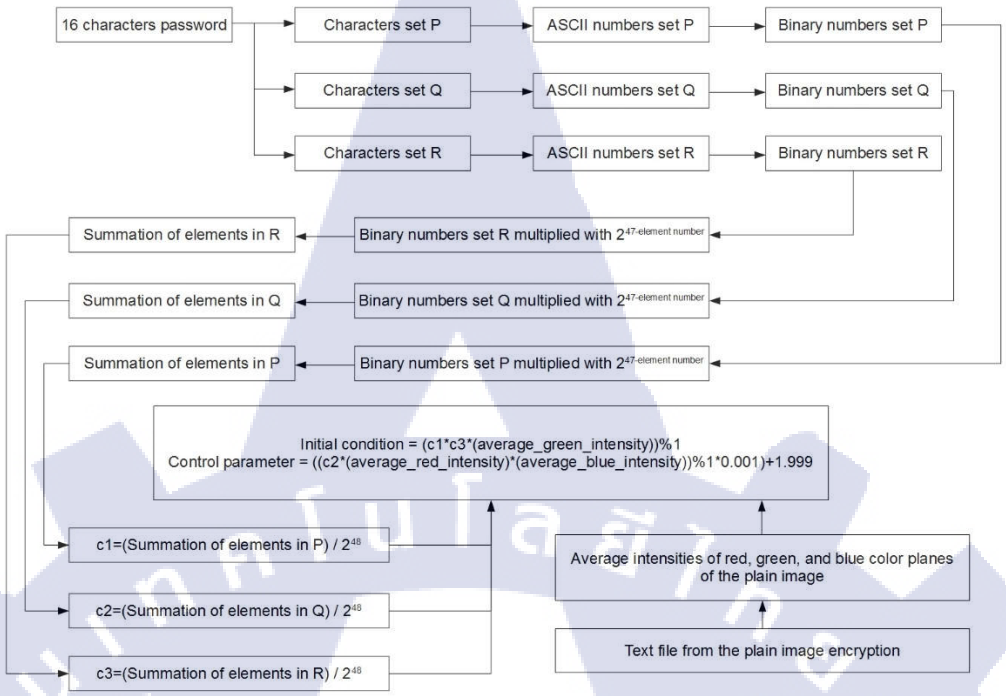


Figure 4.30 Block diagram of the initial condition and the control parameter generation process of the second proposed cryptographic approach.

4.5.2 Results of the Second Approach from Android Emulator

The results from the Android emulator which operating the application that implemented the second proposed cryptographic approach are displayed in figure 4.31 through figure 4.33. Figure 4.31 illustrates the results from the encryption process of the second approach, where figure 4.31 (a) shows the plain image, and figure 4.31 (b) shows the encrypted image. Figure 4.32 illustrates the results from the decryption process of the second approach, using the same password as the plain image, and the correct average intensities of the plain image, where figure 4.32 (a) shows the encrypted image from the encryption process, and figure 4.32 (b) shows the decrypted image. Figure 4.33 illustrates the results from the decryption process of the second approach, using either wrong password or incorrect average intensities of the plain image, where figure 4.33 (a) shows the encrypted image, and figure 4.33 (b) shows the decrypted image. From figure 4.31, the plain image has been encrypted and the encrypted image contain no evidence of the plain image. For the decryption process, successful decryption result which acquired by using the correct password and the average

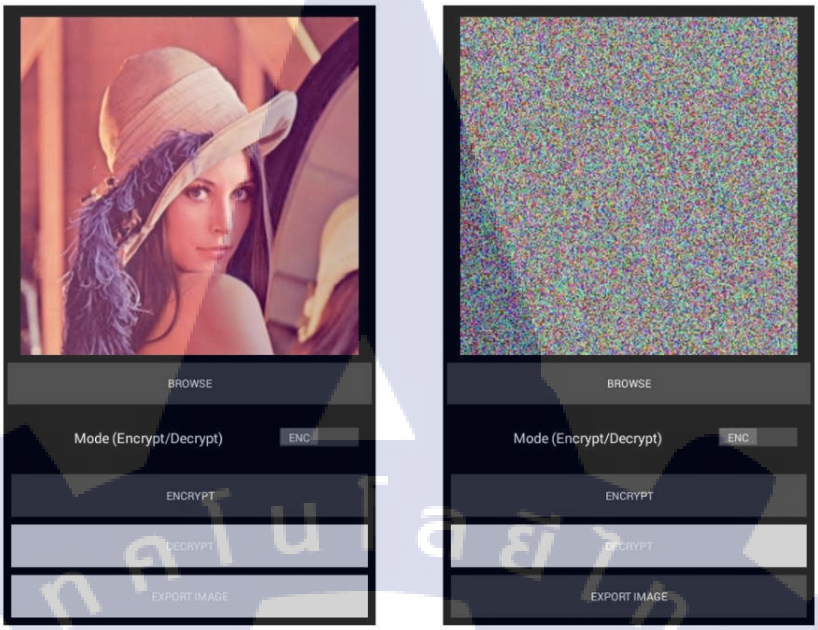


Figure 4.31 Results from the encryption process of the second approach, (a) the plain image before encryption, and (b) the encrypted image.

Figure 4.32 Results from the decryption process using the correct password of the second approach, (a) the encrypted image, and (b) the decrypted image.

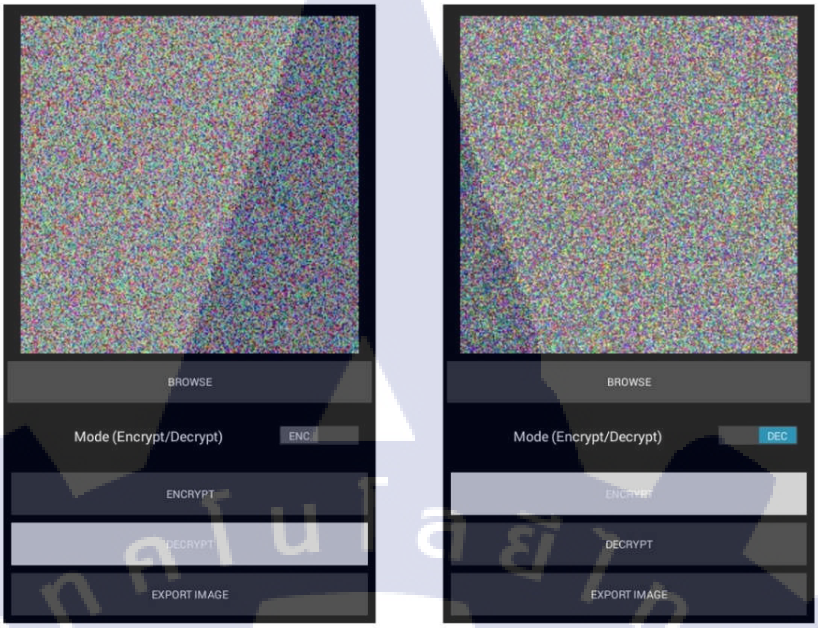


Figure 4.33 Results from the decryption process using the wrong password of the second approach, (a) the encrypted image, and (b) the decrypted image.

intensities of the plain image, can be visualized in figure 4.32 (b), where the result from the decryption process is the same as the plain image in figure 4.30 (a). However, an unsuccessful decryption which is the result from using either wrong password or incorrect average intensities of the plain image is shown in figure 4.33 (b), where the decrypted image has become more scrambled, and contain no evidence of the plain image in figure 4.30 (a). Table 4.5 depicts the computation time of encryption process and decryption process of the second approach. From table 4.5, the computation time is lower than half of a second for the image at the size of 256×256 for both encryption and decryption process, and lower than a second for 512×512. Comparing table 4.5 to table 4.1, the computation time of the second proposed cryptographic approach are lower than computation time of the first approach for all image sizes. The lower computation time in the second approach is acquired from lower steps in processing for both encryption and decryption, since the chaotic map is operated only once in the second approach, while the first approach requires the chaotic map to operate for three times to generate three chaotic sequences.

Table 4.5 Average computation time of encryption process and decryption process of the second proposed cryptographic approach.

Image size	Encryption time (ms.)	Decryption time (ms.)
256×256	225.8	179.6
512×512	733.8	601.2

4.5.3 Analysis of Results from Encryption Process of the Second Approach

The analysis of results from the second approach was completed on MATLAB. The methods of analysis in the second approach are the same as the analysis in the first approach, in which there are qualitative analysis and quantitative analysis in the second approach. The test images for the analysis in the second approach are also LENA image in figure 4.22, and Monkey image in figure 4.23, at the same size as used in the analysis of the first approach.

4.5.3.1 Qualitative Analysis of Cryptographic Approach II

The qualitative analysis of the results from the second proposed cryptographic approach consists of the intensity histogram of the image and the correlation plot of pixels in the image, same as the first approach. Figure 4.34 displays the intensity histograms of the plain image and the encrypted image which have the LENA image as the plain image. Figure 4.34 (b), (c), and (d) shows intensity histograms of red, green, and blue color plane of the plain image, respectively, while figure 4.34 (f), (g), and (h) shows intensity histograms of red, green, and blue color plane of the plain image, respectively. Figure 4.35 illustrates the intensity histograms of the plain image and the encrypted image which have the Monkey image as the plain image. Figure 4.35 (b), (c), and (d) shows intensity histograms of red, green, and blue color plane of the Monkey image, while figure 4.35 (f), (g), and (h) shows intensity histograms of red, green, and blue color plane of the encrypted image which has the Monkey image as the plain image. From figure 4.34 and figure 4.35, the intensity histograms of the encrypted images are uniformly distributed in all color planes, which means all intensities have almost the same probability to be appeared in the image, and intensities also appeared randomly in the encrypted image. The uniformly distributed of histograms of the encrypted image also show that the encrypted image contains no

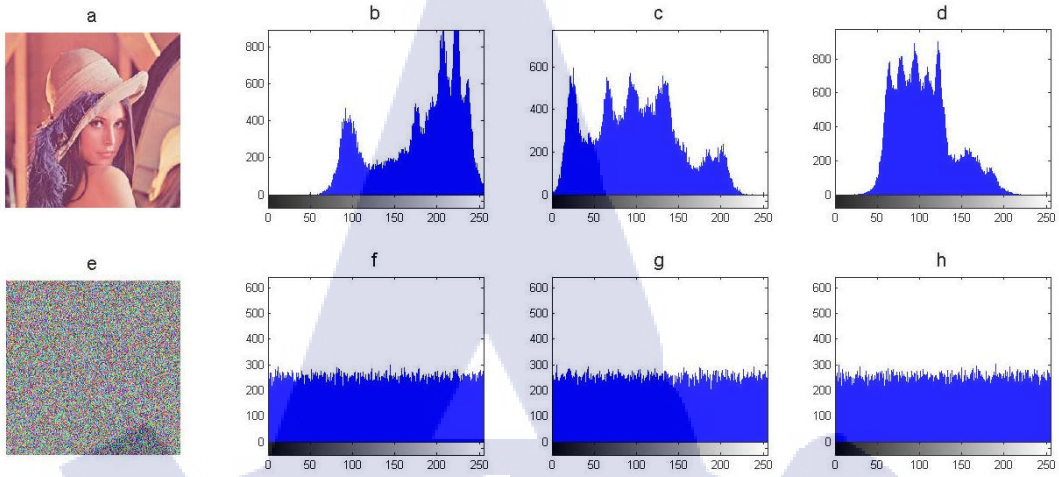


Figure 4.34 Intensity histogram of the LENA image and its encrypted image from the second proposed cryptographic approach.

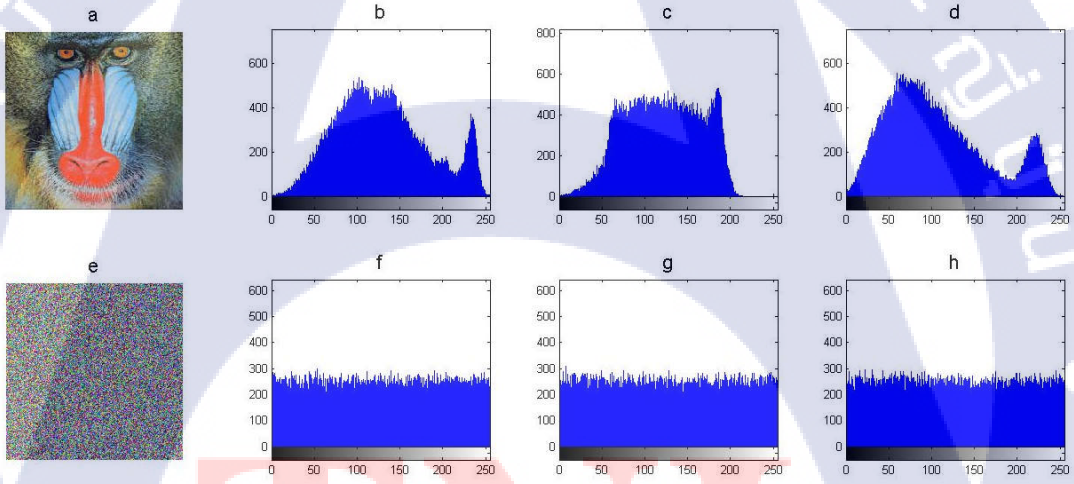


Figure 4.35 Intensity histogram of the Monkey image and its encrypted image from the second proposed cryptographic approach.

evidence of the plain image within. The image correlation plots are also included in the analysis of results from encryption process of the second approach. Figure 4.36 and figure 4.37 shows the correlation plots of pixels in plain images and encrypted images in red color plane. Figure 4.36 shows correlation plots of pixels of the plain image and the encrypted image, using the LENA image in figure 4.22 as the plain image. Figure 4.36 (a), (b), and (c) show the correlation plots of pixels in the plain image with their

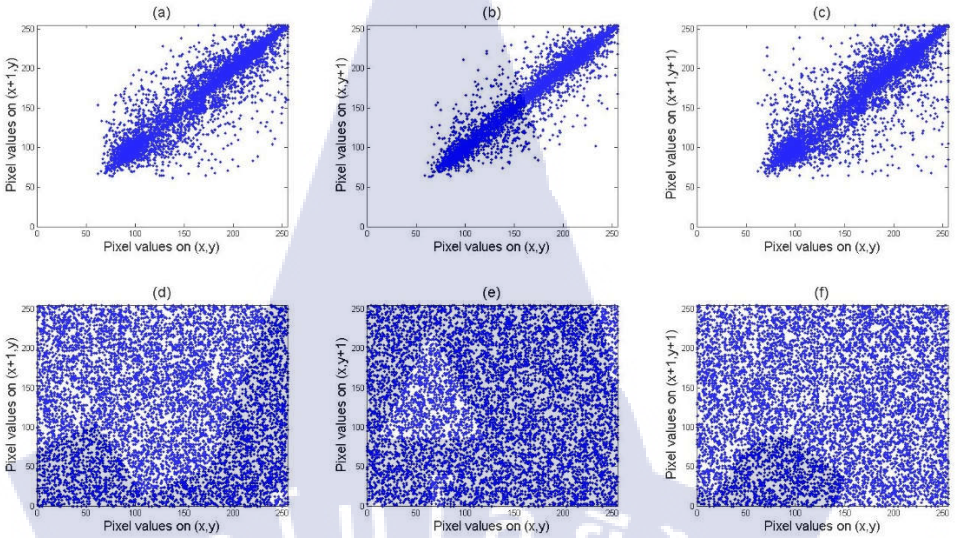


Figure 4.36 Correlation plots of pixels in plain image and encrypted image in red color plane from the second approach, using the LENA image as plain image.

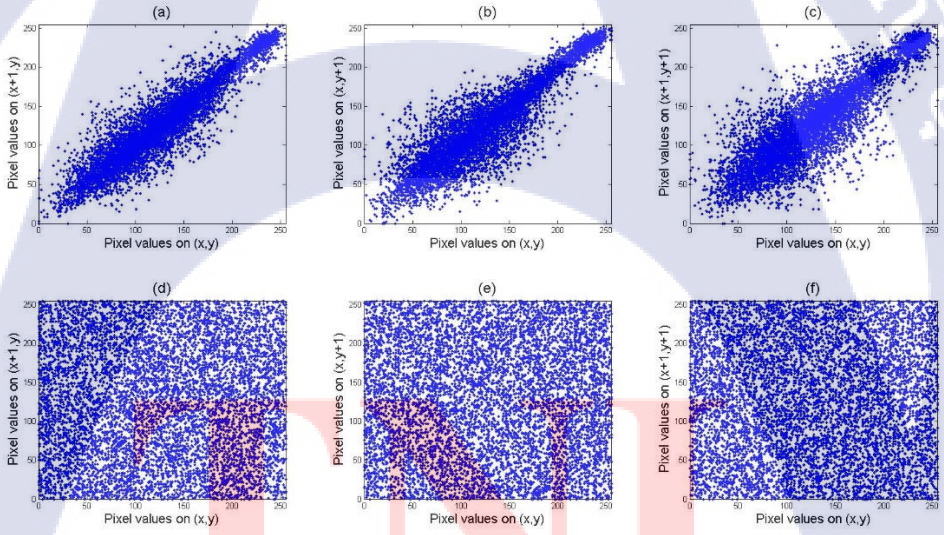


Figure 4.37 Correlation plots of pixels in plain image and encrypted image in red color plane from the second approach, using the Monkey image as plain image.

adjacent in horizontal, vertical, and diagonal directions, respectively, while figure 4.36 (d), (e), and (f) show the correlation plots of pixels in the encrypted image with their adjacent in horizontal, vertical, and diagonal directions, respectively. Figure 4.37 shows correlation plots of pixels in the plain image and the encrypted image, using the Monkey

Table 4.6 Correlation coefficients the plain images and the encrypted images from the second proposed cryptographic approach.

Correlation coefficients (C)	LENA image	Monkey image
$C_{red\text{-}red}$	0.0036	-0.0024
$C_{red\text{-}green}$	-0.0027	0.0036
$C_{red\text{-}blue}$	-0.0034	0.0075
$C_{green\text{-}red}$	0.0026	-0.0077
$C_{green\text{-}green}$	-0.0002	0.0013
$C_{green\text{-}blue}$	-0.0022	0.0042
$C_{blue\text{-}red}$	0.0046	-0.0057
$C_{blue\text{-}green}$	-0.0008	0.0002
$C_{blue\text{-}blue}$	0.0001	0.0012

image in figure 4.23 as the plain image. Figure 4.37 (a), (b), and (c) show the correlation plots of pixels in the plain image with their adjacent in horizontal, vertical, and diagonal directions, respectively, while figure 4.37 (d), (e), and (f) show the correlation plots of pixels in the encrypted image with their adjacent in horizontal, vertical, and diagonal directions, respectively. From correlation plots of pixels in figure 4.36 and figure 4.37, the correlation plots of the plain images shows the similarity of the intensity in each pixel to adjacent pixels, which can be inspected from the pixel intensities that are linearly close to others. On the other hand, correlation plots of the encrypted image shows the scattered intensity of each pixel compared to the intensity of the adjacent pixels, in all directions. The scattered correlation plots of pixels in the encrypted images show that each pixel in the encrypted image is different to its adjacent, and has no relation to be implied as the evidence of the plain image.

4.5.3.2 Quantitative Analysis of Cryptographic Approach II

The quantitative analysis of the results from the encryption process of the second proposed cryptographic approach consists of image correlation coefficients, number pixel changing rate (NPCR), unified averaged changed intensity (UACI), and the information entropy. Table 4.6 depicts the correlation coefficients

Table 4.7 NPCR and UACI in percentage of the encrypted images from the second proposed cryptographic approach.

NPCR/UACI	LENA image	Monkey image
$NPCR_{red}$	99.5697	99.5728
$NPCR_{green}$	99.6414	99.6613
$NPCR_{blue}$	99.6124	99.6353
$UACI_{red}$	33.5511	33.4416
$UACI_{green}$	33.4863	33.3747
$UACI_{blue}$	33.5433	33.4298

Table 4.8 Information entropy of the encrypted images from the second proposed cryptographic approach.

Entropy (H)	LENA image	Monkey image
H_{red}	7.9976	7.9973
H_{green}	7.9973	7.9969
H_{blue}	7.9973	7.9970

which acquired by comparing the plain image in each color plane to all color plane of the encrypted image. Correlation coefficients of both LENA image and the Monkey image compared to their encrypted images, are relatively close to zero, which is the ideal value for two images that are completely different. Table 4.7 shows the NPCR and UACI value acquired by comparing two encrypted images which have the same plain image that have the difference only one pixel, encrypted with the same key. NPCR values of the encrypted images are relatively close to 100 percent for both LENA image and Monkey image. Therefore, the encrypted image will change almost entirely when the plain image is changed, even only one pixel. UACI values of the encrypted images are relatively close to 33 percent for both LENA image and the Monkey image for all color planes. The value of 33 percent in UACI means the encrypted image change its intensities as the plain image changed, even only one pixel. From higher NPCR and UACI, the second proposed cryptographic approach is stronger against the differential attacks than the first approach. Table 4.8 shows the information entropy value of red,

green, and blue color planes of the encrypted images which each has its plain image as the LENA image and the Monkey image. Similar to the first approach, entropy values of the encrypted images in all color planes of the encrypted images in table 4.8 are relatively close to 8, which is an ideal value of images which their intensities of pixels have the same probability to be appeared in the image.

4.6 Conclusions

This chapter has proposed two cryptographic approaches, based on Chaotic Maps with Absolute Value Nonlinearity. The mathematical analysis and simulation of the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity, in terms of chaotic waveforms in time domain, frequency domain, bifurcation diagram, and Lyapunov exponent spectrums on MATLAB and Android were also included. The results from the encryption process of the first and the second proposed cryptographic approaches were analyzed qualitatively through intensity histograms and image correlation plots, and quantitatively through image correlation coefficients, NPCR, UACI, and the information entropy.

The analysis of results states that the second approach was an improvement of the first approach, in which the results of the second approach were strong against differential attacks, measured using NPCR and UACI. Moreover, the computation time of the second approach were lower than the first approach, due to less process in the second approach. For other results from the analysis, the first approach and the second approach have similar results which were close to ideal values for each analysis.

Chapter 5
Conclusion

5.1 Introduction

This chapter summarize the thesis research and suggestion for further researches and implementation. The first part of this chapter summarizes the objectives and proposed approaches in this thesis. The second part of this chapter discusses the results from the proposed cryptographic approaches, suggestions for further researches, and implementations into fields of applications.

5.2 Summary

The objectives of this thesis as described in the first chapter were, to study Chaotic Maps with Absolute Value Nonlinearity on Android, and create cryptographic systems on Android. This thesis has satisfied all objectives described in the first chapter. The first case and the second case of Chaotic Maps with Absolute Value Nonlinearity were analyzed on MATLAB and Android, and two cryptographic approaches which implemented the first case and the second case of Chaotic Maps with Absolute Value Nonlinearity were proposed in this thesis.

Chaotic Maps with Absolute Value Nonlinearity were analyzed mathematically and simulated in this thesis. Despite the four cases exist in the Chaotic Maps with Absolute Value Nonlinearity, only the first case and the second case were analyzed and simulated on MATLAB through bifurcation diagrams, Lyapunov exponents spectrums, chaotic waveforms in time domain, and chaotic waveforms in frequency domain. The bifurcation diagrams and Lyapunov exponents spectrums of the first case and the second case show the robust dynamic of the chaotic maps, in which there is no periodic window in both bifurcation diagram and Lyapunov exponents. Moreover, the chaotic waveforms in time domain also shows the sensitivity of the first case and the second case to the initial condition. The first case of Chaotic Maps was simulated on Android using the same method as the simulation on MATLAB, and the simulation results which acquired from the Android emulator were almost the same as the results from MATLAB.

Two cryptographic approaches were proposed in this thesis. The first proposed cryptographic approach utilized the first case of Chaotic Maps with Absolute Value Nonlinearity as chaotic sequence generator. The first approach was relying on password of 16 characters to generate three initial conditions and a control parameter to be used in the chaotic map. The generated chaotic sequence was used to scramble the plain image, or unscramble the encrypted image, and create the image which was used as key to encrypt or decrypt the image. The method of encryption in the first approach was the XOR operation, by XOR the scrambled plain image with the key, or the unscrambled encrypted image with the key. For the second proposed cryptographic approach, the method of encryption was the XOR operation, same as the first approach, and the scramble and unscramble method were the same as the first approach. The differences of the second approach compared to the first approach were the generation of initial condition, which was reduced to one initial condition, and the use of the average intensities of the plain image to generate the initial condition and the control parameter. The chaotic sequence in the second approach was reduced to one sequence, instead of three sequences of the first approach. The results from the encryption process of the first and the second approaches were analyzed qualitatively through intensity histograms and correlation plots, and quantitatively through correlation coefficients, NPCR, UACI, and the information entropy.

5.3 Conclusions

The implementations of chaotic maps to the cryptographic approaches on Android were successful. Results from the encryption process of the first proposed cryptographic approach and the second proposed cryptographic approach were analyzed through methods mentioned earlier.

The proposed cryptographic approaches were analyzed in terms of computation performances by measure computation times of the encryption and decryption process. As results from Android emulator in Android Studio 2.0, the computation time of the first approach and the second approach were about 0.2 to 0.3 seconds, where the computation time of the second approach for encryption and decryption were lower than the first approach for images at the size of 256×256. The computation time increased as the image size increased, where computation time for

encryption process and decryption process were increased to 0.7 to 0.8 seconds for images at the size of 512×512 for both proposed cryptographic approaches.

For the analysis of results from the encryption process, there were two means of analysis which were qualitative analysis and quantitative analysis. The qualitative analysis consists of intensity histograms and image correlation plots of pixels. Results from the qualitative analysis for the first approach and the second approach were almost the same, where the intensity histograms of the encrypted images in the first approach and the second approach were uniformly distributed, which show no evidence of intensities from the plain image in the encrypted images. Moreover, correlation plots of pixels in the encrypted image of the first and the second approaches were scrambled, or intensity of pixels were almost completely different to the adjacent pixels. For the quantitative analysis, the results from correlation coefficients of results from the first approach were similar to the second approach, in which correlation coefficients in all color planes are relatively close to zero. Moreover, the information entropy value which show the probabilities of information were close to 8, which is an ideal value for images which have intensities stored in 8-bit system. However, the NPCR and UACI test which test results from the encryption against differential attacks were close to zero for the first approach, which shows the weakness of the first approach against differential attacks. For the second approach, the NPCR values were about 99% for all color planes of test images, and the UACI value were about 33%, which means the results from the second approach changes as the plain image changes, even one pixel.

Despite successful implementations and satisfied results from the encryption process, the proposed cryptographic approaches are still weak in terms of usage. The first approach is weak against differential attacks, and the second approach requires additional text file to complete the decryption process. Moreover, proposed cryptographic approaches use the symmetric key algorithm, in which keys needed to be sent from sender to receiver through private channels which have potentials to be intercepted if the channels are not secured enough. Besides, the proposed approaches capable of encryption and decryption only image files, while there are many more types of files require security and in many cases, privacy of information.

The proposed cryptographic approaches can be improved to be able to process with more types of files. Especially, the second approach which is strong against

various types of attacks, and does not need to create any data of the same type as the key to encrypt or decrypt the data. Moreover, the proposed cryptographic approaches can be implemented into various devices and platforms of applications. For example, the implementation into versatile hardware such as FPGA, Raspberry Pi, and other microcontrollers, or other platforms such as Python programming, web-based systems, and IOS mobile platform of Apple. For ease of usage and more security, the proposed cryptographic approaches should be improved to use asymmetric algorithms, which will reduce the difficulties in sending and receiving password or additional text files needed in decryption of the encrypted data.

Finally, the applications of the proposed cryptographic approaches can be in various fields of applications. For security, this system can be used to encrypt secret images which need more security such as government document images, picture of suspects in crimes, or celebrity secret images. Moreover, the proposed cryptographic approaches can be implemented for security of other file types, which will give security of any data in present, and more variations of applications will appear. Thus, more alternatives of security for data and information will be available for users to choose.

References

References

[1] D. Houcque, *Introduction to MATLAB for Engineering Students*, Chicago: Northwestern University, 2005.

[2] B. Guttman and E. A. Roback, *An Introduction to Computer Security: The NIST Handbook*, Gaithersburg: National Institute of Standards and Technology, 1995.

[3] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, Boston: Cengage Learning, 2009.

[4] P. Blanchard et al., *Differential Equations*, Boston: Cengage Learning, 2012.

[5] M. Cecini et al., *Chaos from Simple Models to Complex Systems*, Singapore: World Scientific, 2016.

[6] R. Hilborn, *Chaos and N. Dynamics: An Introduction for Scientists and Engineers*, New York: Oxford University Press, 2000.

[7] W. San-Um and P. Ketthong, "The generalization of mathematically simple and robust chaotic maps with absolute value nonlinearity," *2014 IEEE Region 10 Conference, TENCON 2014*, Bangkok, Thailand, October 22–24, 2014, pp. 1–4.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Upper Saddle River: Pearson Education, 2010.

[9] M. Evening, *Adobe Photoshop CS3 for Photographers*, Oxford: Elsevier, 2007.

[10] Google Inc., *"Android, the world's most popular mobile platform,"* [Online]. Available: https://developer.android.com/about/android.html. [Accessed: January 23, 2016].

[11] Google Inc., *"Android Studio, The Official IDE for Android,"* [Online]. Available: http://developer.android.com/sdk/index.html. [Accessed: January 4, 2016].

[12] Google Inc., *"Developer Workflow: App Workflow,"* [Online]. Available: http://developer.android.com/tools/workflow/index.html. [Accessed: January 31, 2016].

[13] Y. Wu et al., "NPCR and UACI randomness tests for image encryption," *Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 4, pp. 31–38, April 2011.

[14] Y. Wu et al., "Local shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, no. 1, pp. 323–342, February 2013.

[15] M. Naor and A. Shamir, "Visual cryptography," *Workshop on the Theory and Application of Cryptographic Techniques, EUROCRYPT'94,* Perugia, Italy, May 9-12, 1994, pp. 1–12.

[16] M. Ahmad and M. S. Alam, "A new algorithm of encryption and decryption of images using chaotic mapping," *International Journal on Computer Science and Engineering*, vol. 2, no. 1, pp. 46–50, January 2010.

[17] H. Hsiao and J. Lee, "Fingerprint image cryptography based on multiple chaotic systems," *Signal Processing*, vol. 113, no. 1, pp. 169–181, August 2015.

[18] R. F. Martinez-Gonzalez and J. A. Diaz-Mendez, "Implementation of a stream cipher based on bernoulli's map," *International Journal of Computer Science & Information Technology*, vol. 6, no. 6, pp. 113–121, December 2014.

[19] G. Savithri and K. L. Sudha, "Android application for secret image transmission and reception using chaotic steganography," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 7, pp. 5107–5113, July 2014.

[20] H. Liu et al., "Chaos-based color image block encryption scheme using S-box," *AEU - International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 676–686, July 2014

[21] Ch. K. Volos et al., "Image encryption process based on chaotic synchronization Phenomena," *Signal Processing*, vol. 93, no. 5, pp. 1328–1340, May 2013.

[22] L. Merah et al., "Design and FPGA implementation of lorenz chaotic system for information security issues," *Applied Mathematical Sciences*, vol. 7, no. 5, pp. 237–246, January 2013.

Appendices

THE DEVELOPMENT OF AN ANDROID APPLICATION FOR THE CHAOTIC MAP WITH ABSOLUTE VALUE NONLINEARITY

Wimol San-Um, and Sivapong Nilwong

Intelligent Electronic Systems (IES) Research Laboratory
Faculty of Engineering, Thai-Nichi Institute of Technology (TNI)
1771/1 Pattanakarn 37 Rd., Suanluang, Bangkok, Thailand, 10250. E-mail: wimol@tni.ac.th

ABSTRACT

This paper presents an Android application which illustrates dynamic properties of the chaotic map with absolute value nonlinearity in terms of bifurcation diagram, LE spectrum, chaotic waveforms in time-domain, and chaotic waveforms in frequency-domain. This application has been developed using the Java programming language on an ADT plug-in integrated Eclipse IDE. The proposed application has been tested on an Android emulator included in the ADT. As the result from an emulator, the proposed application can be run on Android virtual devices.

Keywords: Android application, chaotic map, bifurcation.

1. INTRODUCTION

Chaotic system possesses significant properties, including the sensitivity to initial conditions and system parameters which make the system dynamic. Chaotic map is an iterated function in a discrete time-domain which can perform chaotic behaviors [1]. Recently, many forms of chaotic map have been employed in many fields of researches. For instances, in computer security, such as the proposing of four chaotic maps with absolute value nonlinearity which has a random bit generation and randomness tests included [1], the random bit generation using the skew tent map [2], and the pseudo-random binary sequence generation using two-dimensional Hénon map [3], in image encryption such as an image encryption technique using two-dimensional Baker map [4], and image encryption which used multiple chaotic maps including logistic map, tent map, quadratic map, and Bernoulli map as layers of chaotic maps [5].

Android is one of the most popular mobile platforms which have been used in millions of mobile devices. Android also gives developers a platform for applications development and an open marketplace for distributing the developed applications [6].In spite of being widely used, only few Android applications have implemented the chaotic maps. For example, there is a recent research which proposed an Android application which used the Hénon map in the chaotic steganography for the secure image transmission and reception [7].

This paper aims to generalize one case of the chaotic maps with absolute value nonlinearity on an Android application. Such chaotic maps haven't been widely used on mobile applications development. Therefore, this might be a good demonstration to the usage of the chaotic maps with absolute value nonlinearity on an Android platform. The developments of the chaotic time-domain and frequency-domain waveforms, the bifurcation diagram, and the LE spectrum of one case of the chaotic maps with absolute value nonlinearity on an Android application have been demonstrated. The results from experiments of the proposed application on an Android emulator also included.

2. CHAOTIC MAPS WITH ABSOLUTE VALUE NONLINEARITY

Chaotic maps with absolute value nonlinearity were proposed in the recent research in four cases. These chaotic maps have been tested for the robustness and simplicity by passing all NIST test in all cases [1]. Despite the four proposed cases in the recent research, the proposed Android application in this paper uses the first case of these chaotic maps which is described as

$$x_{n+1} = |1 - a x_n| \qquad (1)$$

where a is a control parameter and in the range of [0, 2]. Normally, basic analysis of chaotic behaviors in chaotic maps can be achieved qualitatively and quantitatively. The qualitative analysis can be achieved through the bifurcation diagram. In contrast, the quantitative analysis can be achieved through the Lyapunov Exponent (LE) [1]. The term bifurcation is used to describe any sudden change in the dynamic of the system, i.e. when a fixed point changes character as parameter values are changed [8]. Hence, the bifurcation diagram indicates possible long-term values, involving fixed points of a system as a function of a bifurcation parameter. Fig. 1 shows the bifurcation diagram of (1) from MATLAB. It can be seen from the bifurcation diagram that chaotic behaviors occur in the range of a in [1,

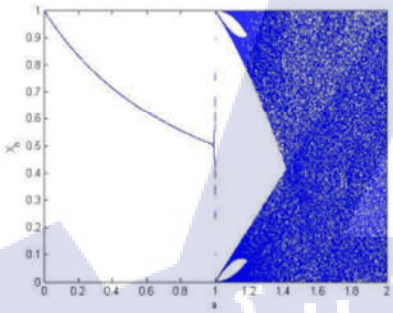


Fig.1 Bifurcation diagram of (1) where the control parameter α is in the range [0, 2].

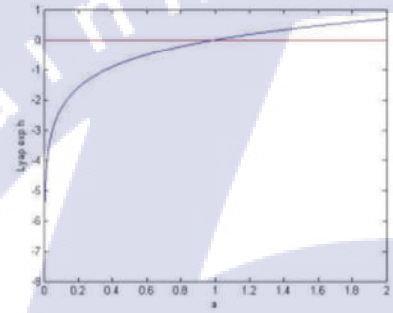


Fig. 2. LE spectrum of (1) where the control parameter α is in the range [0, 2].

2]. On the contrary to the bifurcation diagram, the LE provides a measurement of the instability of the system. Moreover the LE quantify the mean rate of divergence of trajectories which start infinitesimally close to the reference and expressed as [9]

$$\lambda = \lim_{N\to\infty} \left(\frac{1}{N}\right) \sum_{n=1}^{N} \ln\left(\left|f'(x_n)\right|\right) \qquad (2)$$

where N is the number of iterations, λ is the LE, and f'(x) is the first derivative of the chaotic map. Fig. 2 shows the LE spectrum of (1) from MATLAB. From the LE spectrum, the value of LE is more than 1 in the range of α in [1, 2] which is the same region that chaotic behaviors occur in the bifurcation diagram.

Table 1. The configurations of an Android emulator.

Configurations	Selected options/values
Device	4.7" WXGA(1280 × 720 xhdpi)
Target	Android 4.0.4 – API level 15
CPU/ABI	ARM
Skin	Skin with dynamic hardware controls
Back camera	None
RAM	512 MiB
VM heap	64 MiB
Internal storage	512 MiB
SD card	512 MiB

3. DEVELOPMENT AND TEST ENVIRONMENTS

Android applications development can be done on many development tools. However, the recommended tool for new developers by an official Android developer site and is used in this paper is the ADT. Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to give developers an integrated environment which can build Android applications. ADT extends the capabilities of Eclipse to let developers set up new Android projects, create an application UI, add packages based on the Android Framework API, and debug applications using the Android SDK tools included in the ADT [10].

ADT also includes an Android Virtual Device Manager that is used to create an Android Virtual Device (AVD) which is an emulator configuration that let developers model a device by defining hardware and software options to be emulated by the Android emulator. The Android emulator mimics all of the hardware and software features of a typical mobile device except actual phone calls on a computer and let developers' prototype, develop and test Android applications without using a physical device [11]. The configuration of an emulator which is used in this paper is illustrated in Table 1.

Despite being rich features, the ADT does not consist of the capabilities of chart plotting or FFT calculating. In order to solve this problem, the achartengine library and the JTransforms library have been included to an Android application development project in this paper. The achartengine library gives the ADT a potential to create charts which are needed in this paper. Moreover, the JTransforms library gives the ability of FFT calculating which is also needed in this paper to the ADT.

Fig. 3. The flow chart of the proposed Android application.

Fig. 4. The main menu page of the proposed application

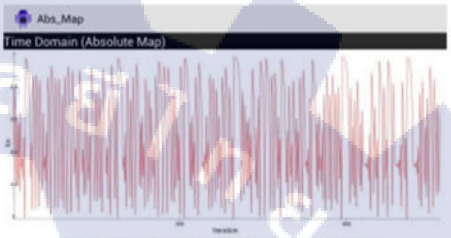


Fig. 5. The chaotic time-domain waveforms.

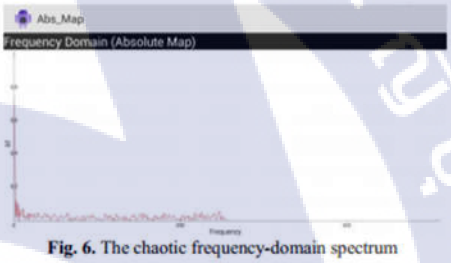


Fig. 6. The chaotic frequency-domain spectrum

4. PROPOSED ANDROID APPLICATIONS

The proposed Android application demonstrates the characteristics of the chaotic map with absolute value nonlinearity or the absolute map in terms of bifurcations, Lyapunov exponents, and chaotic waveforms in time-domain and frequency-domain. The flow chart of the proposed application is illustrated in Fig. 3. As illustrated in the flow chart, the main menu page as illustrated in Fig. 4 is the first page to be shown when the application is started. After users of the application selected the desired page by pressing the button on the main menu screen, the application will be computed as programmed for each page.

For the time-domain page, the value of x from (1) in each iteration is calculated for 512 iterations using the initial condition of x as 0.1 and the control parameter α as 1.999. Afterward, all calculated values of x are displayed as the chaotic time-domain waveforms on the application as shown in Fig. 5.

Similar to the time-domain page, the values of x from (1) in each iteration are also calculated in the frequency-domain page using the same initial condition, control parameter, and the number of iterations as in the time-domain page. However, the calculated values of x are then converted into the frequency-domain data by the complex Foward method which is included in the JTransforms library and the sampling frequency of 512 Hz. The converted data are then displayed as the frequency-domain waveforms on the application as shown in Fig. 6. For the bifurcation page, the data which are displayed as the bifurcation diagram on the application in Fig. 7, are peak values of x in time-domain which calculated from (1) for each value of α which has an incremental of 0.001 and in the range [0, 2].

In LE spectrum page, for each value of α which has an incremental of 0.001, values of x have been calculated from (1) same as the bifurcation diagram page. However, after all values of x have been calculated, values of x are then employed to the LE calculation as described as (2). After the LE values have been calculated, the LE values are then employed in the LE spectrum plotting and displayed as in Fig. 8. In addition to Fig. 5 to Fig. 8, the colors have been inverted to illustrate plotted charts more clearly.

It can be seen from the experiments on an Android emulator, the proposed Android application can be operated on an emulator which has the configurations similar to the typical low-end Android devices. Moreover, results of the bifurcation diagram and the LE spectrum from the proposed application are closely similar to ones from MATALB.

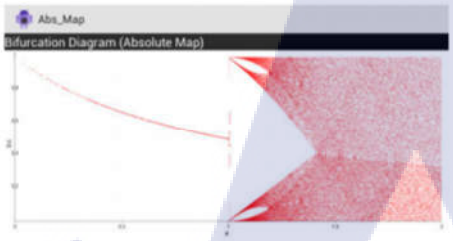


Fig. 7. The bifurcation diagram from the bifurcation page.

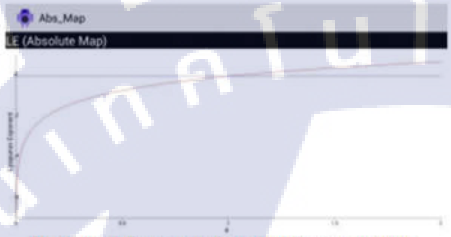


Fig. 8. The LE spectrum from the LE spectrum page.

5. CONCLUSIONS

This paper has proposed an Android application which illustrates bifurcation diagram, LE spectrum, and chaotic waveforms in time-domain and frequency-domain of one the case from one-dimensional chaotic maps with absolute value nonlinearity. The application has been tested on an emulator in the ADT plug-in integrated Eclipse IDE. In addition, this application also demonstrated that one case of the chaotic maps with absolute value nonlinearity can be run on Android. Hence, it can be an alternative map to further android application developments which have chaotic maps involved.

ACKNOWLEDGEMENTS

The authors are grateful to Research and Academic Service Division of Thai-Nichi Institute of Technology (TNI) for research fund.

REFERENCES

[1] W. San-Um and P. Ketthong, "The Generalization of Mathematically Simple and Robust Chaotic Maps with Absolute Value Nonlinearity".

[2] N. K Pareek, V. Patidar, and K. K Sud, "A random bit generator using chaotic maps," *International Journal of Network Security*, vol. 10, no. 1, pp. 32–38, Jan. 2010.

[3] M. Suneel, "Cryptographic pseudo-random sequences from the chaotic Hénon map," *Sādhanā*, vol. 34, Part 5, pp. 689–701, Oct. 2009.

[4] L. Gupta, R. Gupta, and M. Sharma, "Low Complexity Efficient Image Encryption Technique Based on Chaotic Map," *International Journal of Information & Computation Technology*, vol. 4, no. 11, pp. 1029–1034, 2014.

[5] G. A. Sathishkumar, K. B. Bagan, and N. Sriraam, "Image Encryption Based on Diffusion and Multiple Chaotic Maps," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 3, no. 2, pp. 181–193, Mar. 2011.

[6] http://developer.android.com/about/index.html

[7] G. Savithri and K. L. Sudha, "Android Application for Secret Image Transmission and Reception Using Chaotic Steganography," *International Journal of Innovative Research in Computer and Communication Engineering.*, vol. 2, issue 7, pp. 5107-5113, Jul. 2014.

[8] R. Hilborn, "Bifurcation Theory," *Chaos and Nonlinear Dynamics: An Introduction for Scientist and Engineers*, pp. 106, 2000.

[9] M. Cencini, F. Cecconi, A. Vulpiani, "Characteristic Lyapunov exponents," *CHAOS From Simple Models to Complex Systems*, pp. 111-126, 2010.

[10] http://developer.android.com/tools/sdk/eclipse-adt.html

[11] http://developer.android.com/tools/devices/emulator.html

An Image Encryption Scheme and Its Android Application using Robust Chaotic Map with Absolute Value Nonlinearity

Sivapong Nilwong and Wimol San-Um
Intelligent Electronic Systems (IES) Research Laboratory
Faculty of Engineering, Thai-Nichi Institute of Technology (TNI)
1771/1 Pattanakarn 37 Rd., Suanluang, Bangkok, Thailand, 10250. E-mail: sivapong@tni.ac.th

***Abstract*—This paper presents a chaos-based image encryption approach for an Android application based on a chaotic map with absolute value nonlinearity. The chaotic map with absolute value nonlinearity is analyzed and dynamic properties are described in terms of bifurcation diagram and Lyapunov Exponent (LE) spectrum before employing in the Android application. The process of encryption and decryption of the implemented Android application are designed based on XOR operation of the separated color plane of the plain image and the key image which is generated from the chaotic map. The encryption performances of the application are evaluated qualitatively through the pixel density histogram and pixel correlation plots, and quantitatively through encryption time and correlation coefficients.**

***Keywords*—Image Cryptography, Chaotic Map, Android, Absolute value nonlinearity.**

I. INTRODUCTION

Data transmissions and storage over the internet has recently become more significant as the internet technology grows especially for image data which is transmitted over the internet in various applications, for instances, medical and military imaging systems. Recent research proposed various approaches for image encryption, such as non-chaotic watermarking by dividing the plain image into secret shares and embedded to the envelope image using invisible digital watermarking [1], the chaotic-based image encryption process using the XOR operation with the key generated the true random bits generator based on the interaction between two nonlinear circuits [2], the chaotic-based image encryption scheme using the two dimension cat map for shuffling the pixels of the plain image and logistic map for the encryption [3], the chaos-based image encryption technique using two-dimensional Baker map [4], the image encryption approach which used multiple chaotic maps including logistic map, tent map, quadratic map, and Bernoulli map as layers of chaotic maps [5]. The varieties of the chaotic-based encryption approaches were suggested because of the properties of the chaotic systems, including the sensitivity to initial conditions and system parameters which make the systems dynamic. The chaotic map which is an iterated function in a discrete-time domain which can perform chaotic behaviors [6] is one of the most implemented chaotic systems. Despite the fact that various usages of the chaotic maps in the field of image encryption have been suggested, only few implementation of the chaotic-based image encryption on real devices were reported, for instance, the chaotic-based secret image transmission and reception using Hénon map on Android [7]. Moreover, the implemented chaos-based encryption approaches on real devices make the system complex and relatively difficult to be used.

This paper therefore proposes the image encryption scheme based on chaotic map with absolute value nonlinearity which can be implemented on Android. The absolute value nonlinearity-based chaotic maps was generalized on Android in terms of chaotic waveforms in time-domain and frequency-domain, bifurcation diagram, and LE spectrum which illustrated the availability of these chaotic maps on Android [8].

II. THE FIRST CASE OF CHAOTIC MAPS WITH ABSOLUTE VALUE NONLINEARITY

Robust chaotic maps which based on absolute value nonlinearity were proposed in recent research in four cases [6]. All cases of absolute value nonlinearity-based chaotic maps were tested for their robustness by passed the National Institute of Standards and Technology (NIST) statistical test suite from 800-22rev1a special publication for all 15 standard tests from the test suite with the random bits sequence of 1,000,000 bits generated from the chaotic maps. Despite the fact that the absolute value nonlinearity-based chaotic maps were proposed in four cases, this paper implements only the first case of the absolute value nonlinearity-based chaotic maps as follows;

$$x_{n+1} = |1 - ax_n| \tag{1}$$

where a is a control parameter of the chaotic map which has

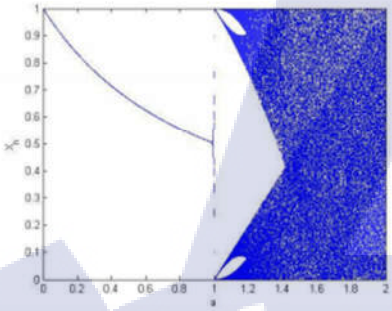


Fig. 1. Bifurcation diagram of (1) where the value of control parameter a is in the range of [0, 2].

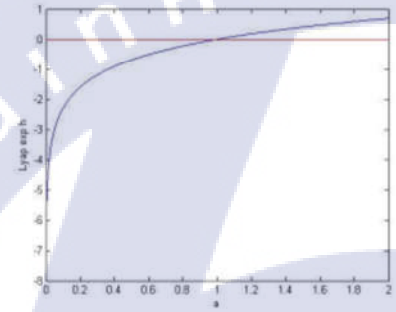


Fig. 2. LE spectrum of (1) where the value of control parameter a is in the range of [0, 2].

the value in the range of [0, 2] and x_n is the state variable of the chaotic map which has the value in the range of [0, 1].

Typically, chaotic behaviors of chaotic maps can be analyzed qualitatively through the bifurcation diagram and quantitatively through the LE spectrum. The bifurcation diagram indicates possible long-term values, including fixed points of a system as a function of a bifurcation parameter, since the term bifurcation is used to describe any sudden change in the dynamic behaviors of the system [9]. Fig. 1 shows the bifurcation diagram of (1) generated from MATLAB. This bifurcation diagram illustrates chaotic behaviors that are possible to occur when the value of control parameter a is in the range of [1, 2] as numerous peaks of the condition value x_n from (1) is indicated in this region. The Lyapunov exponent, however, quantifies the mean rate of divergence of trajectories which start infinitesimally close to the reference and also provides a measurement of the instability of the system. The Lyapunov exponent is expressed as [10]

TABLE I. SUMMARY OF 16 ALPHANUMERIC CHARACTERS PASSWORD FOR INITIAL CONDITIONS AND CONTROL PARAMETER GENERATIONS.

Set	Characters
P	$A_0A_4A_2A_6A_8A_{10}$
Q	$A_{13}A_{11}A_{15}A_5A_9A_3$
R	$A_7A_1A_{14}A_{12}A_0A_4$

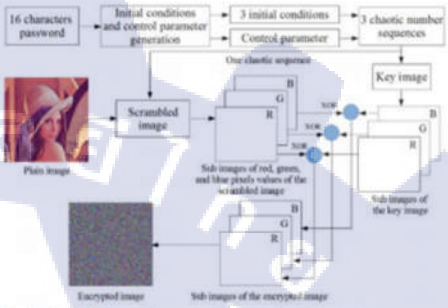


Fig. 3. Block diagram of the proposed encryption scheme.

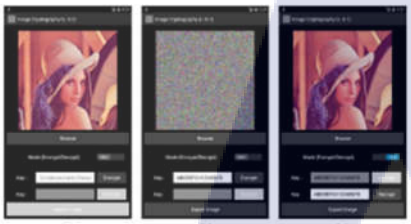
$$\lambda = \lim_{n\to\infty}\left(\frac{1}{N}\right)\sum_{n=1}^{N}\ln f'(x_n) \qquad (2)$$

where N is the number of iterations, λ is the LE value, and $f'(x_n)$ is the first derivative of the chaotic map, Fig. 2 shows the LE spectrum of (1) from MATLAB. As seen from the LE spectrum, the LE value is greater than 0 or the system has chaotic behaviors occurred when the value of control parameter a is in the range of [1, 2] which is the same region that the bifurcation diagram of (1) depicts the chaotic behaviors.

III. PROPOSED IMAGE ENCRYPTION SCHEME

The proposed chaotic-based image encryption scheme to be implemented for Android application in this paper is based on XOR operation of the separated color planes of the plain image and the key image which is created using the number sequences generated from (1).

The encryption process of the proposed scheme which has the diagram shown in Fig. 3 starts with the acquisition of the plain image and the password of 16 alphanumeric characters defined as $A=A_0A_1A_2A_3....A_{15}$. The alphanumeric password A is then separated into 3 sets of alphanumeric characters P, Q, R as described in Table 1 and translated to ASCII codes. The translated ASCII codes are then converted into binary represents defined as B_{P0} to B_{P47}, B_{Q0} to B_{Q47}, and B_{R0} to B_{R47} which takes 8 bits for one



(a) (b) (c)

Fig. 4. (a) Plain image before encrypting, (b) encrypted image, and (c) plain image after decrypting the encrypted image with the correct key.

(a) (b)

Fig. 5. (a) Encrypted image and (b) encrypted image after decrypting with the wrong key.

ASCII code and converted to real number as the followings;

$$c_1 = (B_{P0} \times 2^{47} + B_{P1} \times 2^{46} + \ldots + B_{P47} \times 2^0)/2^{48} \quad (3)$$

$$c_2 = (B_{Q0} \times 2^{47} + B_{Q1} \times 2^{46} + \ldots + B_{Q47} \times 2^0)/2^{48} \quad (4)$$

$$c_3 = (B_{R0} \times 2^{47} + B_{R1} \times 2^{46} + \ldots + B_{R47} \times 2^0)/2^{48} \quad (5)$$

The calculated real numbers are implemented in 3 initial conditions, i.e. I_r, I_g, and I_b and the control parameter a generation which are generated by

$$I_r = (c_1 \times c_2)\%1 \quad (6)$$

$$I_g = (c_1 \times c_3)\%1 \quad (7)$$

$$I_b = (c_2 \times c_3)\%1 \quad (8)$$

$$a = (((c_1 \times c_2 \times c_3)\%1) \times 0.1) + 1.9 \quad (9)$$

The initial conditions I_r, I_g, and I_b are then employed in (1) to create the chaotic sequences of numbers S_r, S_g, and S_b, respectively. The chaotic sequences S_r, S_g, and S_b will be used in the image key generation which achieved by

TABLE II. CONFIGURATIONS OF THE ANDROID EMULATOR

Configurations	Selected options/values
Device	5.1" WXGA(480×800 mdpi)
Target	Android 4.4.2 – API level 19
CPU/AB1	armeabi-v7a
Back camera	None
RAM	512 MB
VM heap	32 MB
Internal storage	512 MB
SD card	512 MB

TABLE III. COMPUTATION TIME FOR 256×256 AND 512×512 LENA IMAGE BETWEEN (1) AND THE LOGISTIC MAP

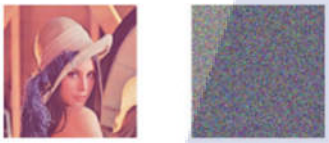
Chaotic Map	256x256	512x512
Chaotic map with absolute value nonlinearity	5.565 s.	20.360 s.
Logistic map	5.790 s.	22.463 s.

multiplying each number in S_r, S_g, and S_b by 255 and set to the red, green, and blue value of the pixels, respectively, starting from the first pixel on the top-left of the image key. Additionally, the chaotic sequence S_r was also employed in the scrambling of the plain image by multiplying each number in S_r by (image size – 1) then swap each pixel T_n with the pixel positioned at the number on the n position of the sequence of the multiplied S_r, where n of T_n and multiplied S_r starts from 0 or the top-left pixel and finish at (image size – 1) or the bottom-right pixel of the image.

After the plain image was scrambled, the scrambled image is separated into 3 color planes: red, green, and blue and then XOR each pixel with the pixel of the separated color plane from the image key at the same pixel position and the same color plane. After the XOR operation of pixels was completed, the result color planes from the XOR operation are then combined into the output image of the encryption scheme. For the decryption of the proposed scheme, the process is reversed by implementing the XOR operation first, then unscramble the result from the XOR operation into the decrypted image.

IV. THE IMPLEMENTATION OF ANDROID APPLICATION

The implementation of Android application was developed on the Android Development Tools (ADT) plug-in integrated Eclipse Integrated Development Environment (IDE) and the performance in term of computation time was tested on the emulator on Android Studio which is the official IDE for Android application development [11]. The



(a) (b)

Fig. 6. (a) LENA image and (b) encrypted image.

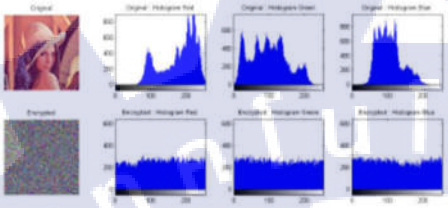


Fig. 7. Pixel density histogram of the plain image and the encrypted image in red, green, and blue color plane, respectively.

emulator has the configurations as shown in Table 2. Fig. 4 shows the results from the implemented application when the plain image was encrypted and decrypted using the correct key as "ABCDEFGH12345678". On the other hand, Fig. 5 shows the result when the password was wrong at the sixth character which was "ABCDEEGH12345678" since the sixth character was one of the least significant characters of the password when calculated from Table 1. The computation time of the encryption process when encrypting the LENA image as shown in Fig. 6 (a) at the size of 256×256 and 512×512 using equation (1) and the logistic map was described and compared in Table 3 which reveals that the equation (1) is faster than the logistic map.

V. ENCRYPTION RESULT ANALYSIS

The result image shown in Fig. 6 (b) was taken from the implemented Android application which used the LENA image at the size of 256×256 as the plain image was analyzed using MATLAB R2013a quantitatively through the correlation coefficients and qualitatively through the pixels correlation plot and the pixel density histogram. Fig. 7 shows the pixel density histogram of the plain image and the encrypted image which was uniformly distributed among the pixel values. Fig. 8 illustrates the pixel correlation diagram in the red color plane of the plain image on the top row and the encrypted image on the bottom row of Fig. 8. The pixel correlation diagram of the plain image depicts the values of pixels which are similar to the adjacent pixels values horizontally ((X, Y) and $(X+1, Y)$), vertically ((X, Y) and $(X, Y+1)$), and diagonally ((X, Y) and $(X+1, Y+1)$) whilst the pixel correlation diagram of

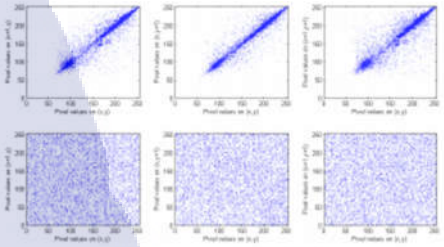


Fig. 8. Pixel correlation diagram of the plain image and encrypted image of the red color plane with the adjacent pixel horizontally, vertically, and diagonally, respectively.

TABLE IV. CORRELATION COEFFICIENTS BETWEEN THE PLAIN IMAGE AND THE ENCRYPTED IMAGE

Correlation coefficients (C)	Value
C_{RR} (Red-Red)	0.0057
C_{RG} (Red-Green)	6.8983×10^{-4}
C_{RB} (Red-Blue)	-0.0053
C_{GR} (Green-Red)	0.0098
C_{GG} (Green-Green)	0.0011
C_{GB} (Green-Blue)	-0.0016
C_{BR} (Blue-Red)	0.0098
C_{BG} (Blue-Green)	8.7718×10^{-4}
C_{BB} (Blue-Blue)	4.3062×10^{-4}

the encrypted image shows the scattered values of pixels when compared to the next pixels horizontally, vertically, and diagonally. The correlation coefficients of the encrypted image compared to the plain image defined as C shown in Table 4 is significantly close to zero when compared each color plane of the plain image to all color plane of the encrypted image. The analysis result of the encrypted image reveals that the proposed image cryptography scheme is capable of encrypting and decrypting the plain image at basic level of data security.

CONCLUSIONS

This paper has proposed an image encryption scheme based on the absolute value nonlinearity-based chaotic map. The proposed encryption scheme can be employed on Android and the used absolute value nonlinearity-based chaotic map can compute faster than the logistic map when tested the implemented application on an emulator in the Android studio. The result from the encryption also illustrated the robustness of the encryption scheme when tested on MATLAB using the pixel density histogram and

pixel correlation diagram with the satisfied result on the histogram which is uniformly distributed and the pixel value in the pixel correlation diagram scattered. Even though the satisfied result is achieved, the algorithm of the proposed encryption scheme needed to be improved and tested with tests from more types of security attacks.

ACKNOWLEDGEMENTS

The authors are grateful to Research and Academic Service Division of Thai-Nichi Institute of Technology (TNI) for financial support.

REFERENCES

[1] S. Kandar, A. Maiti, and B.C. Dhara, "Visual Cryptography Scheme for Color Image Using Random Number with Enveloping by Digital Water Marking", *IJCSI International Journal of Computer Science Issues*, vol. 8, issue 3, no. 1, pp. 543-549, May 2011.

[2] Ch.K. Volos, I.M. Kyprianidis, and I.N. Stouboulos, "Image encryption process based on chaotic synchronization phenomena," *Signal Processing*, vol. 93, pp. 1328–1340, 2013.

[3] M. Ahmad et al, "A New Algorithm of Encryption and Decryption of Images Using Chaotic Mapping," *International Journal on Computer Science and Engineering*, vol. 2, pp. 46–50, 2009.

[4] L. Gupta, R. Gupta, and M. Sharma, "Low Complexity Efficient Image Encryption Technique Based on Chaotic Map," *International Journal of Information & Computation Technology*, vol. 4, no. 11, pp. 1029–1034, 2014.

[5] G. A. Sathishkumar, K. B. Bagan, and N. Sriraam, "Image Encryption Based on Diffusion and Multiple Chaotic Maps," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 3, no. 2, pp. 181–193, 2011.

[6] W. San-Um and P. Ketthong, "The Generalization of Mathematically Simple and Robust Chaotic Maps with Absolute Value Nonlinearity", *TENCON 2014-2014 IEEE Region 10 Conference*. Pages 1–4, 2014.

[7] G. Savithri and K.L. Sudha, "Android Application for Secret Image Transmission and Reception Using Chaotic Steganography," *International Journal of Innovative Research in Computer and Communication Engineering.*, vol. 2, issue 7, pp. 5107-5113, 2014.

[8] W. San-Um and S. Nilwong, "The Development of an Android Application for The Chaotic Map with Absolute Value Nonlinearity", *MITicon 2014*, pp. 112-115, 2014.

[9] R. Hilborn, "Bifurcation Theory," *Chaos and Nonlinear Dynamics: An Introduction for Scientist and Engineers*, pp. 106, 2000.

[10] M. Cencini, F. Cecconi, and A. Vulpiani, "Characteristic Lyapunov exponents," *CHAOS From Simple Models to Complex Systems*, pp. 111-126, 2010.

[11] Google Inc. (2015). Android Studio Overview [Online]. Available: http://developer.android.com/tools/studio.

A Highly-Secured Chaos-Based Cryptographic Algorithm on Android for Private Image Storage and Transmissions

Sivapong Nilwong and Wimol San-Um
Intelligent Electronic Systems (IES) Research Laboratory
Faculty of Engineering, Thai-Nichi Institute of Technology (TNI)
1771/1 Pattanakarn 37 Rd., Suanluang, Bangkok, Thailand, 10250. E-mail: sivapong@tni.ac.th

***Abstract*—This paper presents a chaos-based cryptographic algorithm on Android for private image storage and transmissions. The robust chaotic map based on absolute value nonlinearity is presented and analyzed before employing in the encryption process. Dynamic behaviors of the chaotic map are described in terms of Equilibria, Jacobian matrix, bifurcation diagram, and Lyapunov Exponent spectrum. The designed encryption process and decryption process were based on XOR operations of pixel values with the random bit signals generated from the chaotic map. The keys were designed based on a 16 alphanumeric characters password for the initial condition and control parameter. The implemented Android application was developed using Java programming language on the Android studio. The generally used LENA image with the size of 256×256 and 512×512 were employed for demonstrations of encryption and decryption procedures. Encryption qualitative performance were evaluated through pixel density histograms and correlation plots. Encryption quantitative performances were evaluated through encryption speed, correlation coefficients, Net Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI). Results from actual Android application on a smart phone with both correct and wrong keys are also included.**

Keywords*—*Chaos, Cryptographic Algorithm, Android, Image Storage, Image Transmission.

I. INTRODUCTION

Recent advances in communications, especially the internet, have led to great demands of secured data storage and transmissions. Security and reliability of data are needed in various applications such as medical, industrial, military, and internet of things [1]. From recent research, image data is one of the most used data types in data cryptography [2-5]. Various cryptographic approaches were suggested for both software and hardware, involving chaos-based approaches. Chaos-based cryptography exploits the properties of chaotic systems, including sensitivity to initial conditions and parameters of the system. Chaotic maps which are iterative functions in discrete-time domain that can exhibit behaviors, were frequently implemented in chaos-based cryptography [3-4]. For example, the image encryption and decryption algorithm using the logistic map and two-dimensional cat map [3], and the colored image encryption algorithm using the logistic map and characteristics of plain image [4]. However, chaotic maps implemented recently were not robust and led to complex design of the cryptographic system. Despite the fact that, an implementation of cryptographic systems on mobile platforms provide convenience in image encryption and decryption, only few implementations of chaos-based cryptography using chaotic maps on mobile device were reported, such as the chaos-based secret image transmission and reception on Android [5].

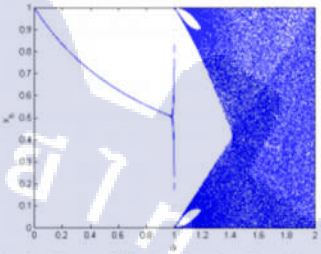


Fig. 1. Bifurcation diagram of (1) where the value of control parameter *a* is in the range of [0, 2].

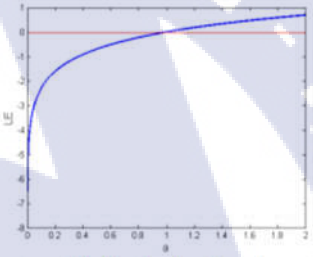


Fig. 2. LE spectrum of (1) where the value of control parameter *a* is in the range of [0, 2].

According to the report in 2015 by the Internet Society, Android has a significant portion of mobile devices share [6]. Therefore, this paper proposed the cryptographic algorithm on Android mobile platform. The proposed cryptographic algorithm is based on the second case of chaotic maps with absolute value nonlinearity and the XOR operation. The implementation of chaotic maps with absolute value nonlinearity on Android was experimented in previous research by generalizing the first case of chaotic maps with

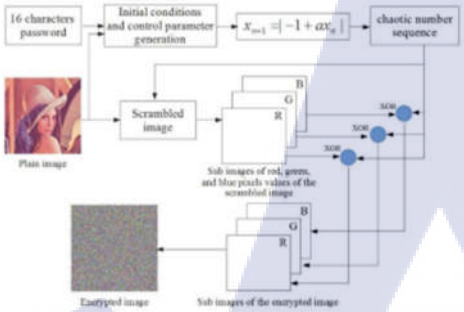


Fig. 3. Block diagram of encryption procedure.

absolute value nonlinearity on Android [7]. Results from the experiment on Android stated that chaotic maps with absolute value nonlinearity can be implemented on Android.

II. THE SECOND CASE OF CHAOTIC MAPS WITH ABSOLUTE VALUE NONLINEARITY

The second case of chaotic maps with absolute value nonlinearity which implemented in this paper is one of four robust chaotic maps proposed in the recent research [8]. All cases of chaotic maps with absolute value nonlinearity were tested for their robustness by testing random bit sequences which consist of 1,000,000 bits generated from the chaotic maps. Random bit sequences generated from chaotic maps with absolute value nonlinearity passed all 15 standard tests of the National Institute of Standards and Technology (NIST) statistical test suite from 800-22rev1a special publication. The second case of chaotic maps with absolute value nonlinearity can be mathematically described as

$$x_{n+1} = |-1 + a x_n| \tag{1}$$

where a is the control parameter of the chaotic map which has the value in the range of [0, 2] and x_n is the state variable of the chaotic map which has the value in the range of [0, 1]. Additionally, the Jacobian matrix of (1) is calculated as

$$J = [a(sign(-1 + ax))] \tag{2}$$

, and the fixed points of (1) are indicated as

$$x = \frac{1}{a \pm 1} \tag{3}$$

Chaotic behaviors of (1) are also analyzed in this paper, qualitatively through the bifurcation diagram, and quantitatively through the Lyapunov Exponent (LE) spectrum.

Fig. 4. a. The plain image, b. The encrypted image, c. The decrypted image using the correct key, and d. The decrypted image using the wrong key

The bifurcation diagram indicates possible long-term values as a function of a bifurcation parameter. Fig. 1 shows the bifurcation diagram of (1) from MATLAB. From the bifurcation diagram, chaotic behaviors of (1) occurred when the control parameter a is in the range of [1, 2] in which the number of blue dots that refer to possible values in time domain of (1) is enormous in this region. The LE quantifies the mean rate of divergence of trajectories which start infinitesimally close to the reference and also provides a measurement of the instability of the system. The LE can be expressed mathematically as [9]

$$\lambda = \lim_{n \to \infty} \left(\frac{1}{N}\right) \sum_{n=1}^{N} \ln f'(x_n) \tag{4}$$

where N is the number of iterations, λ is the LE value, and $f'(x_n)$ is the first derivative of the chaotic map, Fig. 2 shows the LE spectrum of (1) from MATLAB. As seen from the LE spectrum, the LE value is greater than 0 when the value of control parameter a is in the range of [1, 2], that is, equation (1) exhibits chaotic behaviors in this region, the same region as the bifurcation diagram of (1).

III. PROPOSED SECURED CHAOS-BASED CRYPTOGRAPHIC ALGORITHM

The proposed chaotic-based cryptographic algorithm for Android in this paper is based on XOR operation of the pixel values in each separated color planes of the plain image with numbers in a chaotic sequence generated from (1).

Fig. 3 shows the block diagram of the encryption process in the proposed chaos-based cryptographic algorithm. The encryption process starts by the acquisition of the plain image and the password which is a set of 16 alphanumeric characters. The acquired password is transformed into a set of ASCII number which each alphanumeric character is transformed into 8-bit ASCII number, resulted in a 128-bit numeric password. The transformed password and average pixel values of each color plane of the plain image are used to calculate the initial condition and the control parameter for the chaotic map in (1). The generated initial condition and control parameter are then employed in (1) to generate a chaotic sequence of numbers

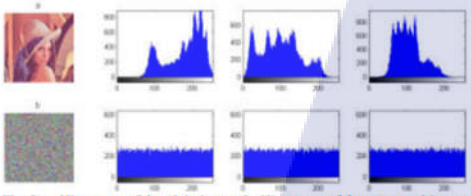


Fig. 5. a. Histogram of the plain image, b. Histogram of the encrypted image in red, green, and blue color planes, respectively

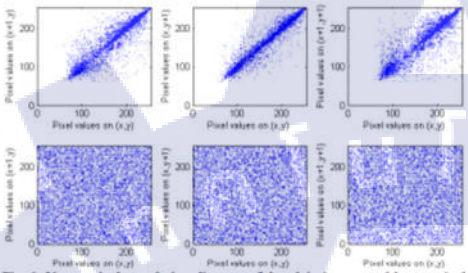


Fig. 6. Upper, pixel correlation diagram of the plain image, and lower, pixel correlation of the encrypted image in red color plane to adjacent pixels in horizontal, vertical, and diagonal directions, respectively

with the number of iterations used equals to number of pixels+2. After the chaotic sequence is generated, the acquired plain image is then scrambled using the generated chaotic sequence to swap pixel values of each pixel in all color planes with the pixel indicated by the number of each element of the chaotic sequence multiplied by the number of pixels-1, starts from the top-leftmost pixel of the plain image and the first element of the chaotic sequence. The scrambled image is then processed through the XOR operation by XOR pixel values in each color plane with numbers in the generated chaotic sequence from (1) multiplied by 255, starts from the top-leftmost pixel of the scrambled image. Additionally, XOR operation of pixel values in red color plane starts by XOR the multiplied number of the first element of the chaotic sequence with the first pixel value in red color plane, while the XOR operation of green and blue color planes starts by XOR multiplied numbers of the second and the third elements of the chaotic sequence with the first pixel values in green and blue color plane, respectively. After the XOR process was completed, the processed or encrypted image is displayed and prompt users to export the encrypted image. The encrypted image can be exported as .PNG image file along with average pixel values in each color plane as .txt text file. The exported .txt text file which contains average pixel values is required in the decryption process.

The decryption process is similar to the encryption process. However, there are some differences which are, first, the decryption process requires average pixel values in each color plane of the original plain image which contained in the text

TABLE I. CORRELATION COEFFICIENTS BETWEEN THE PLAIN IMAGE AND THE ENCRYPTED IMAGE AT THE SIZE OF 256×256 AND 512×512

Correlation coefficients (C)	256×256	512×512
C_{RR} (Red-Red)	0.0036	0.0008
C_{RG} (Red-Green)	-0.0027	-0.0016
C_{RB} (Red-Blue)	-0.0034	-0.0008
C_{GR} (Green-Red)	0.0026	0.0010
C_{GG} (Green-Green)	-0.0002	0.0003
C_{GB} (Green-Blue)	-0.0022	-0.0009
C_{BR} (Blue-Red)	0.0046	0.0017
C_{BG} (Blue-Green)	-0.0008	0.0018
C_{BB} (Blue-Blue)	0.0001	-0.0007

TABLE II. NPCR AND UACI OF THE ENCRYPTED LENA IMAGE AT THE SIZE OF 256×256 AND 512×512

	256×256	512×512
NPCR (Red)	99.5697	99.5975
NPCR (Green)	99.6414	99.6250
NPCR (Blue)	99.6124	99.6143
UACI (Red)	33.5511	33.4609
UACI (Green)	33.4863	33.5438
UACI (Blue)	33.5433	33.5533

file that exported from the encryption process to generate initial conditions and control parameters for (1). Second, the XOR operation is operated before the image unscrambling. The last difference is that, the image unscrambling process which has the same mechanism as the scrambling process, starts from the pixel on the bottom-rightmost of the image instead of the top-leftmost pixel.

IV. EVALUATION OF THE PROPOSED CHAOS-BASED CRYPTOGRAPHIC ALGORITHM

The implemented Android application was developed on the Android Studio Integrated Development Environment (IDE) which is an official IDE for Android application development. Despite the fact that the Android Studio has an emulator which can be used to test the developed application on virtual device, the implemented application in this paper was tested on an actual Android device, the SONY Xperia M2 smartphone. Fig. 4 displays the results on the smartphone which operating the implemented application, where (a) shows the plain image before the encryption, (b) shows the encrypted image, shows the decrypted image which used the same key as the encrypted key, and (d) shows the decrypted image using the wrong key. The average computation time used in the encryption process when encrypt the LENA image in sizes of 256×256 and 512×512 pixels, 5 times for each image, were measured. The experiment on the smart phone shows that

average computation time of the 256×256 pixels LENA image is 0.837 seconds, and average computation time of 512×512 pixels LENA image is 3.591 seconds.

The result from encryption process of the implemented Android application which used the 256×256 LENA image as the plain image, was analyzed using MATLAB 2013a quantitatively and qualitatively. Quantitative analysis was achieved through the correlation coefficients, the Number of Pixel Changing Rate (NPCR), and the Unified Averaged Changed Intensity (UACI). Besides, qualitative analysis was achieved through the pixel density histograms and the pixel correlation diagrams. Fig. 5 shows the pixel density histograms of the plain image and the encrypted image. The pixel density histograms show that the encrypted image has uniformly distributed histograms in all color planes, i.e. red, green, and blue color planes, while the pixel density histograms in all color planes of the plain image are not. Fig. 6 illustrates the pixel correlation diagrams of the plain image and the encrypted image in red color plane. Pixel correlation diagrams depict the pixel values of each pixel compared to adjacent pixels in horizontal, vertical, and diagonal directions. The correlation diagrams of the plain image reveal that pixel values of each pixel are similar to adjacent pixels. On the other hand, pixel values of each pixel in the encrypted image are significantly differ to their adjacent pixels, the correlation diagrams of the encrypted image are scattered in all directions.

Table 1 depicts the correlation coefficients between the plain image and the encrypted image in all color planes. The correlation coefficients were achieved through the correlation coefficient of pixels, comparing each color plane of the plain image to all color planes of the encrypted image. As described in table 1, all correlation coefficients between the plain image and the encrypted image were close to zero. Table 2 shows the NPCR and UACI value of the encrypted 256×256 pixels and 512×512 pixels LENA image. The NPCR is used to measure the changing rate of pixels in encrypted images, and the UACI is used to measure changes in pixel values of the encrypted images, where their corresponding original plain images have some slight changes, i.e. only one pixel changed [10]. The NPCR and UACI can be described as

$$NPCR = \frac{\sum_{i,j} D(i,j)}{T} \times 100 \qquad (6)$$

$$UCAI = \frac{\sum_{i,j} C_1(i,j) - C_2(i,j)}{T \times P} \times 100 \qquad (6)$$

where C_1 and C_2 denote the pixel in encrypted image 1 and 2 which has one pixel changed, D indicates if the pixel value is changed which is 0 if $C_1 = C_2$, and 1 if $C_1 \neq C_2$, i and j is the horizontal and vertical position of pixels in images, T is number of pixels, and P is maximum value of the image format, in this case is 255. The NPCR and UACI in table 2 is close to 99% and 33% for both 256×256 and 512×512 images, which were close to theoretically values.

CONCLUSIONS

This paper has proposed a chaos-based cryptographic algorithm based on the chaotic map with absolute value nonlinearity. The proposed cryptographic algorithm can be implemented on actual Android device, and was tested on a smart phone. The test results on smart phone reveal that the proposed cryptographic algorithm can encrypt the test image which has the size of 256×256 pixels in less than 1 second. The result from the encryption process of the proposed cryptographic algorithm were analyzed qualitatively through pixel density histograms and pixel correlation diagrams, and quantitatively through the correlation coefficients, NPCR, and UACI on MATLAB. Even though the satisfied results are acquired from the analysis, the proposed image cryptographic algorithm need improvements with more types of tests. However, the proposed cryptographic algorithm provides an alternative algorithm to be implemented on mobile platforms.

REFERENCES

[1] M. B. Barcena and C. Wueest, *Insecurity in Internet of Things*, version 1.0, March 2015.

[2] Ch.K. Volos, I.M. Kyprianidis, and I.N. Stouboulos, "Image encryption process based on chaotic synchronization phenomena," *Signal Processing*, vol. 93, pp. 1328–1340, 2013.

[3] M. Ahmad et al, "A New Algorithm of Encryption and Decryption of Images Using Chaotic Mapping," *International Journal on Computer Science and Engineering*, vol. 2, pp. 46–50, 2009.

[4] M. A. Murillo-Escobar et. al, "A RGB image encryption algorithm based on total plain image characteristics and chaos," *Signal Processing*, vol. 109, pp. 119–131, 2015.

[5] G. Savithri and K.L. Sudha, "Android Application for Secret Image Transmission and Reception Using Chaotic Steganography," *International Journal of Innovative Research in Computer and Communication Engineering.*, vol. 2, issue 7, pp. 5107-5113, 2014.

[6] Internet Society, *Internet Society Globas Internet Report 2015*, 2015.

[7] W. San-Um and S. Nilwong, "The Development of an Android Application for The Chaotic Map with Absolute Value Nonlinearity", *MITicon 2014*, pp. 112-115, 2014.

[8] W. San-Um and P. Ketthong, "The Generalization of Mathematically Simple and Robust Chaotic Maps with Absolute Value Nonlinearity", *TENCON 2014-2014 IEEE Region 10 Conference*, pp. 1–4, 2014.

[9] M. Cencini, F. Cecconi, and A. Vulpiani, "Characteristic Lyapunov exponents," *CHAOS From Simple Models to Complex Systems*, pp. 111-126, 2010.

[10] Yue Wu, "NPCR and UACI Randomness tests for Image Encryption", *Journal of Selected Areas in Telecommunication (JSAT)*, April Edition, pp. 31-38, 2011