CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

This project will be used Spread Spectrum method with Least Significant Bit (LSB) method to embed message into the image. The message and the key process by Spread Spectrum method. Then the LSB method used to embed result of Spread Spectrum method into the image. The parameter used in this project is Peak Signal to Noise Ratio (PSNR) to know the quality of images in decibels (dB).

1. Least Significant Bit

The Least Significant Bit will be used to embed the data into the image. Least Significant Bit uses the last bit in binary to input the data. Therefore the effect to RGB value is not so big. Here is the example:

message: A

the binary of Red, Green, Blue (RGB) from cover image:

Table 4.1

binary	00110101	11110001	01011100	11101110	11111111	00011111	11001100	10101010
channel	R	G	B	R	G	B	R	G

binary of character A is 01000001, then insert it into RGB above sequentially so the new binary of RGB from cover image will be:

Table 4.2

binary	0011010**0**	1111000**1**	0101110**0**	1110111**0**	1111111**0**	0001111**0**	1100110**0**	1010101**1**
channel	R	G	B	R	G	B	R	G

The table **4.2** shown the result of embedded binary message into LSB of binary image. So for embed 8 bit or 1 binary it needs 8 channel of RGB and embed it sequentially to the least significant bit.

2. Spread Spectrum

The Spread Spectrum is the main method of this project. Spread Spectrum uses LSB method to embed the message into the image. But before embedding it into the image, the program will process the message and key first. First *888# will be added to the message as the end marker of the text. It has function to tell the program where the message end. After that convert the message to binary. The result of converted message will be spread with 4 scales. Then the program will process the key. The characters of the key will be Exclusive Or (XOR) each other. Then the result of XOR will be seed value for Linear Congruential Generator (LCG) formula. LCG has function to generate random numbers. The formula of LCG is:

$$X_{n+1} = (aX_n + c) \mod m$$

Xn = nth random number

a and c = constants of LCG

m = maximum range of random number

each parameter have rule:

1. m = modulus, $0 < m$

2. a = multiplier, $0 < a < m$

3. c = increment, $0 \leq c < m$

4. X0 = first value, $0 \leq X0 < m$ (seed value)

5. c and m are relatively prime

6. a – 1 can divided by prime factor of m

7. a – 1 is multiple 4 if m multiple 4 too

8. a must be very big number

LCG in this project will generate 4 random numbers. The result of LCG will be convert to binary. After that XOR it with the spreaded binary message. Then embed the result into the image with LSB method. This process also known as embedding process.

The extraction process is a process to extract hidden message from stego image. First select the stego image that has message inside it. Then read the RGB inside it. Convert the RGB to binary. After that extract all of Least Significant Bit (LSB) from the image. Then the program will process the key. The characters of the key will be Exclusive Or (XOR) each other. The result of this process will be seed value for LCG formola like before to generate 4 random numbers. After that, the program will convert the random numbers to binary. Then the program will XOR it with the extracted LSB from the image. The next step is de-spreading the result with 4 scales. The program will eliminate the last 3 bits every 4 bits. Then divides the result in 8 so the program can read it as 1 binary (8 bit). Then the program will convert 1 binary to a character until the program found the end of text marker. When the program found *888# characters in the converted characters then the program will stop converting binary to character. The result will be appeared on pop up window and the process of extraction is done.

3. PSNR

Peak Signal to Noise Ratio (PSNR) is a parameter use to calculate the quality of the result image compare to the original. The formula of PSNR is:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

MAX = maximum value of the pixel (255)

MSE = Mean Squared Error

The formula of MSE:

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

m x n = resolution of image (width * height)

I = original image

K = result image

i,j = coordinate of pixel

4.2 Design

1. Embedding Process

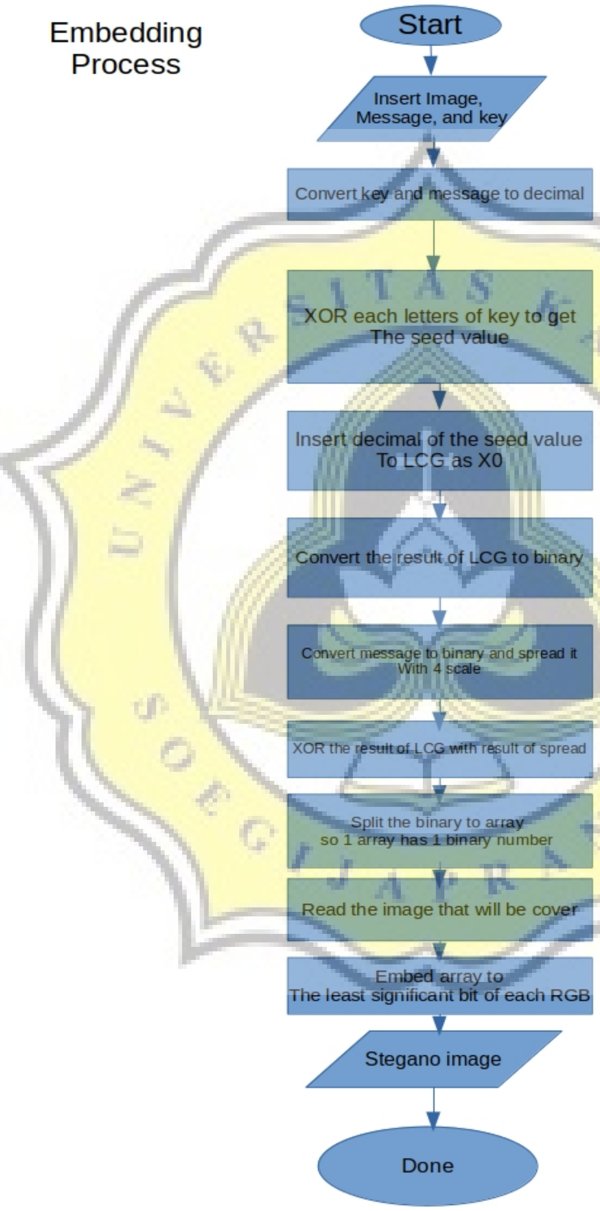
Embedding Process
Start
Insert Image, Message, and key
Convert key and message to decimal
XOR each letters of key to get The seed value
Insert decimal of the seed value To LCG as X0
Convert the result of LCG to binary
Convert message to binary and spread it With 4 scale
XOR the result of LCG with result of spread
Split the binary to array so 1 array has 1 binary number
Read the image that will be cover
Embed array to The least significant bit of each RGB
Stegano image
Done

Illustration 4.1: Flowchart of Embedding Process

The illustration **4.1** explains about the process of embedding. First select image, then input the message and the key. Then the program will convert the message and the key to decimal. XOR each characters of the key to get the seed value. Then insert the seed value to the LCG formula. The result will be converted to binary. Then process the message, add *888# to the message as end marker of the text, then convert it to binary, then spread it with 4 scales. After that, XOR the result of LCG with message. Split the result to array, so 1 array has 1 bit of binary. Read the RGB of the image, then convert it to binary. Then embed the array into the LSB of binary RGB.

2. Extraction Process

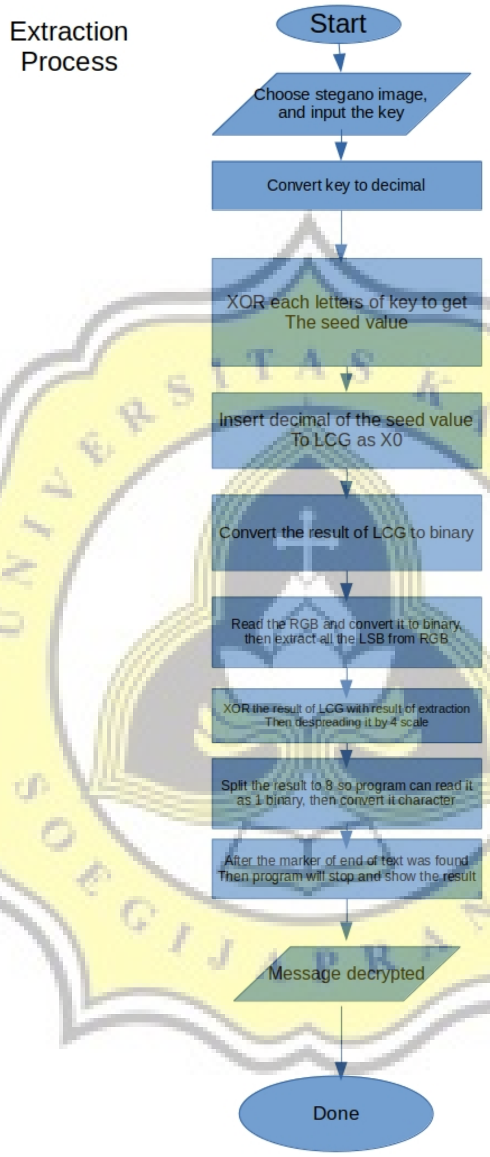
Extraction Process
Start
Choose stegano image, and input the key
Convert key to decimal
XOR each letters of key to get The seed value
Insert decimal of the seed value To LCG as X0
Convert the result of LCG to binary
Read the RGB and convert it to binary, then extract all the LSB from RGB
XOR the result of LCG with result of extraction Then despreading it by 4 scale
Split the result to 8 so program can read it as 1 binary, then convert it character
After the marker of end of text was found Then program will stop and show the result
Message decrypted
Done

Illustration 4.2: Flowchart of Extraction Process

The illustration **4.2** explains about the process of extraction hidden message from the image. First input the stego image and the key. Then the program will process the key to find seed value. After that insert the seed value to

LCG formula to generate 4 random numbers. The result will be converted to binary. After that, process the image by extracting the LSB of binary RGB. Then XOR the result of LSB with the result of binary LCG. The result will be de-spread with 4 scales. Then convert the de-spread result to character. When the end of text was found, the program will stop and show the decrypted message.