Implementation Gifford Method For Digital Image Security

Fauduziduhu Laia¹, Erwin Panggabean²

¹²Informatics Engineering Study Program, STMIK Pelita Nusantara, Jl. Iskandar Muda No. 1 Medan, North Sumatra, Indonesia 20154

E-mail: fauduziduhulaia@gmail.com

Abstract - Rapid development of digital image technology secret causes images require security aspect. Reviews These digital secret image can be encrypted using cryptographic methods. After being encrypted, the image is randomized, so that if it is Obtained by an unauthorized party, the image has no meaning. The cryptographic algorithm used in this study is Gifford method. The Gifford method is a stream cipher, a symmetry encryption algorithm that transforms the data character by character. Gifford has 8 registers filled with key bits. The processes performed by the Gifford method are the Output Function process, the 1-bit Sticky Shift Right process, the 1-bit Left Shift process, the XOR operation and the shift register operation to the right. The decryption process must use the same key as the encryption process in order to Obtain the original image.

Keywords: Encryption, pixel color, Gifford

1. Introduction

Imagery is usually used to store information in the form of images. Digital imagery has been widely used in various fields of human life, including the medical, research, business, military and other fields. In some cases, digital images are used to document things that are confidential, so that the image can only be accessed by certain people only. Examples of the image containing confidential information is the image of the project design, building, office complex and the work in the world of civil engineering (Tempo.co, 2014). The owners do not want the image of the design is stolen and can be accessed by competitors. The other areas that require security is an image in the field of business, such as product design secrets to a company, the image of a patient's medical diagnosis in the medical field, the image of secret documents in the intelligence field, to the private image created for personal documentation (Madhu, et.al, 2016). Confidential so that the image is not accessible to everyone, then the digital image can be encrypted using cryptographic methods. Once encrypted, images become scrambled, so that even if acquired by an unauthorized person, the image has no meaning. To restore the image to its original shape, so do the decryption process by using a specific keyword. Thus, only certain parties can access the image. Gifford method is one of the cryptographic algorithm that can be used to secure the digital image. This method was developed by David Gifford. In the science of cryptography, Gifford is a stream cipher method, namely symmetric encryption algorithm that transforms data character by character (byte by byte). Stream ciphers can be made very quickly, much faster than that of any block cipher algorithm. Gifford method has 8 registers, namely: b0, b1, b2, b3, b4, b5, b6 and b7. These registers have a value that is filled with the bits of the key before further processing. The processes performed in the Gifford method is the process Output Function, Shift Right Sticky process one bit, the Left Shift 1 bit, XOR operation and the shift register operation to the right. Therefore Gifford is a symmetric key cryptography, the decryption process must use the same key to the encryption process, in order to obtain the original image. Based on the description in the background, it will develop an application that can do security on the digital image, through the encryption and decryption process using the Gifford. Thus, this thesis is entitled "Implementation of Gifford Method for Digital Image Security".

2. Theory

2.1. Cryptography

The word cryptography (cryptographic) is derived from the Greek, meaning that KRIPTOS secret (secret), and graphein, which means writing (writing). So, the secret cryptographic means writing (hieroglyph). There is some definitions of cryptography that has been presented in the literature. The definition used in the books of the old (before the 1980s) states that cryptography is the science and art to maintain the confidentiality of messages by way of encoding it into a form that can no longer understand the meaning. This definition may be suitable in the past where cryptography is used for safety critical communications such as communication
among the military, diplomats and spies. But nowadays cryptography is more than just privacy, but also for the purposes of data integrity, authentication and non-repudiation. (Mollin, 2010)

In his book entitled "Handbook of Applied Cryptography", Alfred Menezes, Paul van Oorschot and Scott A. Vanstone (1996: 4) defines cryptography as the study of mathematical techniques related to aspects of information security, such as confidentiality, data integrity and authentication. While Bruce Schneier (2016: 1) in his book "Applied Cryptography", defines cryptography as a science and an art to maintain the security of the message.

Cryptography is the science and art to keep your messages. Security is obtained by encrypting messages into messages that do not mean anything. Today, the confidentiality of information into something important. Confidential information that needs to be hidden so as not known by people who are not eligible. Someone certainly do not want a credit card or a PIN number of his ATM card known. Or, if a message written in secret and does not want to know or read by others. Cryptography can be used to disguise the confidential information of the person or party who is not entitled to read it. (Munir, 2012: 203)

2.2. Gifford

David Gifford find a stream cipher used to encrypt mail and cable (wire news report) in Boston. In cryptography, Gifford is a stream cipher, the symmetric encryption algorithm that transforms data character by character. This method has an 8-byte register, namely: b0, b1, b2, b3, b4, b5, b6 and b7. These registers have values or initial conditions before the subsequent processing (Schneier, 2016).

2.3. Key Generation Process Gifford

This process must be done to generate key bits to be used in the encryption and decryption process. The processes performed in the Gifford methods generate key bits is as follows,

1. Process 1: Process Output Function.

   output of this process is random along the 8 bits. To generate a random number, perform the merging operation (concatenate) between registers b0 and b2 registers and between registers and registers b4 b7. Then, perform a multiplication operation between the result of the merger to produce a 32-bit number. Byte third from left is the output of random numbers. Output process function can be seen in Figure 1.

   Fig 1. Work Procedures Asymmetric Algorithm

2. Process 2: Sticky Shift Right 1 bit to register b1.

   Take the contents of registers b1 and do the process right shift 1 bit sticky to b1. This means shifting bits (shift) to the right. Rated left most bit is maintained and vacant shifted bits (bits that are behind the left most bit) is replaced with the value left most bits. The result is stored to the variable A.

3. Process 3: Left Shift 1 bit face to register b7.

   Take the contents of the register b7 and do surgery left shift one bit. This means that all the bits shifted (shift) to the left and right most bit value is filled with value 0. The result is stored to the variable B.

4. Process 4: XOR operation to register b0, the value of the variable A and the value of the variable B

   Perform XOR operation of the contents of the register b0, the value of the variable A (the result of the second process) and the value of the variable B (the result of the third process). This means that the operation will generate a bit value of '0' (zero) for the bits of the same value and will generate a bit value of '1' (one) for the different bits. Hasilnya stored to the variable C.

5. Process 5: Slide blocks register b to the right one block

   Scroll register b (b0 ... b7) 1 block to the right. This means that the register b7 removed (discard), register b6 filled by registers b5, register b5 filled by registers b4, register b4 filled by the register b3, register b3 filled by the register b2 and register b2 filled by the register b1, while the register b0 empty, filled by the value of the variable C (the results of the fourth process).

Gifford bit key generation process can be seen in full in Figure 2.
2.4. Encryption and decryption processes Gifford

The encryption process on Gifford method will use the key bits to transform plaintext into ciphertext, and vice versa in the decryption process. Encryption and decryption process is in the form of an XOR operation of plaintext and ciphertext or key to produce the ciphertext and the key XOR operation to generate plaintext (Schneier, 2016).

\[
P = C \oplus K
\]
\[
C = P \oplus K
\]

with:
- \(P\) = plaintext
- \(K\) = Key
- \(C\) = ciphertext

2.5. Imagery

Image is a representation (picture), likeness or imitation of an object. Image as the output of a system is an optical data recording can be images, is an analog form of video signals such as images on a TV monitor, or digital nature that can be directly stored on a storage medium. (Sutoyo, et al, 2010: 9)

3. Analysis

The analysis process to discuss the workings of the process of formation of key bits, the encryption process digital images and digital image decryption process by using Gifford.

3.1. Formation Bit Key

For example, if the key used is 'Fauduzid' and about to be raised 10 pieces of sub key, then the key establishment process of the key sub-1 to sub-10 is key to the following:

**The initial value of the register b (0) to b (7):**

- **Input Key** = 'Fauduzid'
  - \(b\) (0) = binary (ascii of the letter 'F') = binary (70) = 01.00011 million
  - \(b\) (1) = binary (ascii of the letter 'a') = binary (97) = 01,100,001
  - \(b\) (2) = binary (ascii of the letter 'u') = binary (117) = 01110101
  - \(b\) (3) = binary (ascii of the letter 'd') = binary (100) = 01.1001 million
  - \(b\) (4) = binary (ascii of the letter 'u') = binary (117) = 01110101
  - \(b\) (5) = binary (ascii of the letter 'z') = binary (122) = 01.11101 million
  - \(b\) (6) = binary (ascii of the letter 'i') = binary (105) = 01101001
  - \(b\) (7) = binary (ascii of the letter 'd') = binary (100) = 01.1001 million

**Key sub-1**

1) **Output Function**
   a. Combine the contents of \(b\) (0) and \(b\) (2), and \(b\) (4) and \(b\) (7).
      - \(b\) (0) and \(b\) (2) = 0100010111110101
      - \(b\) (4) and \(b\) (7) = 0111010110100100
   b. Multiply the result of merging the register:
      - \(0100010111110101 \times 0111010110100100 = 00100001001111111111110110100100\)
   c. The results of the key sub-1 = 11.111111 million (third from left byte)
      - = FE (hex)
2) **Sticky shift right 1 bits in b (1)**
   - \(A = SSR (b\) (1))
= SSR (01100001)
    = 00110000

3) Left shift 1 bits in b (7)
   B = LS (b (7))
   = LS (01100100)
   = 11001000

4) XOR b (0), A and B
   C = b (0) XOR A XOR B
   XOR XOR 00110000 = 01000110 11001000
   C = 10111110

5) Update register b by sliding block B to the right of 1 byte.
   a. Discard the contents of the register b (7).
   b. Scroll to the right block b
      b (7) = b (6) = 01101001
      b (6) = b (5) = 01.11101 million
      b (5) = b (4) = 01110101
      b (4) = b (3) = 0.1001 million
      b (3) = b (2) = 01110101
      b (2) = b (1) = 01,100,001
      b (1) = b (0) = 00.00011 million
   c. The contents of b (0) to the value C.
      b (0) = 10.11111 million

Key sub-2
1) output Function
   a. Combine the contents of b (0) and b (2), and B (4) and b (7).
      b (0) and b (2) = 1011111001100001
      b (4) and b (7) = 0110010001101001
   b. Multiply the result of merging the register:
      X 1011111001100001 0110010001101001
      = 010010101010111111100111001001
   c. The results of the key sub-2 = 11,111,001 (byte third from left)
      = F9 (hex)

2) Sticky shift right 1 bits in b (1)
   A = SSR (b (1))
   = SSR (01000110)
   = 00100011

3) Left shift 1 bits in b (7)
   B = LS (b (7))
   = LS (01101001)
   = 11010010

4) XOR b (0), A and B
   C = b (0) XOR A XOR B
   XOR XOR 01000110 = 0110010001101001
   C = 01001111

5) Update register b by sliding block B to the right of 1 byte.
   a. Discard the contents of the register b (7).
   b. Scroll to the right block b
      b (7) = b (6) = 00.110,001
      b (6) = b (5) = 01101001
      b (5) = b (4) = 01100011
      b (4) = b (3) = 01101110
      b (3) = b (2) = 01110101
      b (2) = b (1) = 01,001,011
      b (1) = b (0) = 00.010,111
   c. The contents of b (0) to the value C.
      b (0) = 01.01011 million

Perform the same calculation to obtain the keys to the sub-3 to sub key 15th. Sub keys 1 through sub key 15th-generated from the input key = 'Fauduzid' can be seen in Table 1.
### Table 1

| Sub kunci ke- | Sub Kunci (dalam Biner) | Sub Kunci (dalam Heks) |
|---------------|-------------------------|------------------------|
| 1             | 1111 1110              | FE                     |
| 2             | 1111 1001              | F9                     |
| 3             | 1011 1110              | BE                     |
| 4             | 0110 0011              | 63                     |
| 5             | 1110 0101              | E5                     |
| 6             | 0001 0110              | 16                     |
| 7             | 0111 1110              | 7E                     |
| 8             | 0110 0001              | 61                     |
| 9             | 1100 0111              | C7                     |
| 10            | 0010 0111              | 27                     |
| 11            | 1110 1110              | EE                     |
| 12            | 0001 1011              | 1B                     |
| 13            | 0001 0000              | 10                     |
| 14            | 0010 1100              | 2C                     |
| 15            | 1101 1011              | DB                     |

#### 3.2. Encryption Process Imagery

Image made of pixels, and each pixel is composed of three elements of color, the Red (red), Green (green) and Blue (blue). Each color component has an intensity value from 0 to 255, or 8-bit binary. In the image of the encryption process, each 8-bit color components of the pixels will be XOR with 8 bit sub-key, sub-key taken from the top.

For example, suppose the pixel-1 to be encrypted with color value as follows:

**Pixel-1:**
- R = 228 = E4 (hex)
- G = 54 = 36 (hex)
- B = 106 = 6A (hex)

By using the key sub-1, 2 and 3 are worth F9 BE FE (hex), then the process of encryption of the first pixel are as follows:

**Results encryption pixel-1:**
- R (new) = E4 (hex) xor FE (hex) = 1A (hex) = 26 (decimal)
- G (new) = 36 (hex) xor F9 (hex) = CF (hex) or 207 (decimal)
- B (new) = 6A (hex) xor BE (hex) = D4 (hex) = 212 (decimal)

Suppose the 2nd pixel color value as follows:

**Pixel-2:**
- R = 203 = CB (hex)
- G = 35 = 23 (hex)
- B = 178 = B2 (hex)

By using the rest of the key sub-4, 5 and 6, which is worth 63 E5 16 (hex), then the encryption process from pixel-2 is:

**Results encryption pixel-2:**
- R (new) = CB (hex) xor 63 (hex) = A8 (hex) = 168 (decimal)
- G (new) = 23 (hex) xor E5 (hex)
Examples of the encryption process to the three pixels of an image can be compiled and shown in Table 2.

### Table 2

| Piksel   | Warna Enkripsi (Desimal) | Sub Kunci (Hex) | Operasi Enkripsi | Warna Hasil Enkripsi (Desimal) |
|----------|--------------------------|-----------------|------------------|--------------------------------|
| Piksel-1 | 228 E4 FE               | 36 F9           | 36 xor F9        | 1A 26                          |
| Piksel-2 | 203 CB 63               | 5 E5            | 22 xor E5        | A8 168                         |
| Piksel-3 | 25 19 7E                | 8A 61           | 8A xor 61        | 67 103                         |
| Piksel-4 | 105 69 27               | 9 A7 18         | 9 xor 18         | 4E 78                          |
| Piksel-5 | 201 C9 10               | 12 B8 2C        | 12 xor 2C        | D9 217                         |

In the same way, do the encryption process on the pixels 6th, 7th pixel to the last pixel image. Examples of the original image and encrypted by using a key = "Fauduzid", can be seen in figure 3.

**Fig 3. Example of Encryption Process**

Figure 3 shows an example of the original image that is encrypted with a particular key and generates an encrypted image that has been scrambled and no meaning.

### 3.3. Decryption Process Imagery

Decryption process is the same image as the image encryption process, using XOR function of cipher-image with key sub-images to produce plain. The function of the decryption process can be expressed as follows:

\[ P = C \oplus K \]

With:

\[ P = \text{Plain-image} \]
\[ K = \text{Key} \]
\[ C = \text{Cipher-image} \]

Image results by using the decryption key = "Fauduzid", can be seen in Figure 4.

**Fig 4. Example of Process Decryption**

Figure 4 shows an example of the encrypted image is decrypted by specific key and reproduces the original image.

### 4. Result

#### 4.1. System planning
The modeling system can be done by using the Unified Modeling Language (UML). UML diagrams to explain the behavior of the system and not how the system works. Activity diagram is a UML diagram. Image encryption process that occurs in the application can be described by an activity diagram as shown in Figure 5.

![Activity Diagram of the Process Encryption](image)

**Fig 5. Activity Diagram of the Process Encryption**

Figure 5 shows a diagram of activity when the user wants to perform the encryption process on the image. Decryption is the reverse of the encryption process. Image decryption process that occurs in the application can be described by an activity diagram as shown in Figure 6. Figure 6 shows a diagram of activity when the user wants to perform the decryption process of the encrypted image.

![Activity Diagram of the Process Decryption](image)

**Fig 6. Activity Diagram of the Process Decryption**

*Use case* is a diagram of the Unified Modeling Language (UML) which can be used to analyze and model the system. Figure 7 shows the interaction between users and systems in use case diagrams.
Implementasi Metode Gifford untuk Pengamanan Citra Digital

USER
Input Kunci
Enkripsi
Enkripsi Citra
Input Kunci
Dekripsi
Dekripsi Citra
Buka Citra
Simpan Citra
About
Keluar

Figure 7 shows the use case diagram of the application. In the notation Use Case, the relationship "include" inter-use case, means the use case use case X uses Y entirely. In use case Figure 7, it appears that the actors of the system is the user (user). Users can open the image that you want to secure, perform image encryption process (before the user must enter the encryption key), perform the decryption process images and save the image of the encryption or decryption process.

5. Conclusion

After completing the application design implementation Gifford method for securing digital image, some things can be concluded is:

a. Applications can perform the encryption process by implementing the method of Gifford, is to perform the XOR operation between the intensity of the color of the pixel image with key byte Gifford result of output function, so that the encrypted digital image can not be accessed by unauthorized parties.

b. Applications can restore an encrypted image to the initial image through the decryption process uses a stream cipher method Gifford.

c. Applications can only decrypt the scrambled image back to the initial image when the keys are used exactly the same encryption key.

d. Applications can display the time required to process the image of the encryption and decryption process.

6. Reference

[1] Ario, Implementation of Presidential Decree No. 26 of 2009 on the Implementation of E-KTP NIK Based in Samarinda, E-journal Science of Government, 2014.
[2] Ariyus, D. Introduction to Cryptography. Andi, Yogyakarta, 2013.
[3] Basuki, et al. Digital Image Processing Using Visual Basic, Graha Science, Yogyakarta, 2015.
[4] Kadir, A. and Susan, A. Theory and Application of Image Processing. Andi, Yogyakarta, 2013.
[5] Kartika, et al. Implementation seal on Data Security, Ambassador Christian University Discourse, Yogyakarta, 2010.
[6] Mollin, RA An Introduction to Cryptography. 2nd ed. Florida: Chapman & Hall / CRC. 2010.
[7] Mesterjon, B. and Zulita, LN Implementation Method for Determining Value Selection Sort Student Achievement Class 3 and Class 4 SD Negeri 107 Seluma, University Dehasen, Bengkulu, 2015.
[8] Munir, R. Discrete Mathematics, Revised Fifth, Informatics, Bandung, 2012.
[9] Munir, R. Cryptography. Informatics, Bandung, 2016.
[10] Son, D. Digital Image Processing. Andi, Yogyakarta, 2010.
[11] Schneier, B. Applied Cryptography. John Wiley and Sons Inc., Canada, 2016.
[12] Sutoyo, T., et al. Theory of Digital Image Processing. Andi, Yogyakarta, 2010.