Journal of Computer Science and Technology Studies (JCSTS)
www.jcsts.one

Research Article

Data Hiding to Image Smart Phone Using AES and LSB Algorithms

Ei Phyu Sin Win
Associate Professor, Department of Information Technology Engineering, Technological University (Kyaukse), Myanmar
Corresponding Author: Ei Phyu Sin Win, E-mail: panthakhin9001@gmail.com

ARTICLE INFO

Article History
Received: May 05, 2019
Accepted: August 15, 2019
Volume: 1
Issue: 1

KEYWORDS

Cryptography, Steganography, Security, technology, communication.

ABSTRACT

In today's fast-paced world of technology, the use of smartphones or iPhones has become so widespread that communication technology has become so important. Protecting that telecommunications network has become very important. The new approach to addressing security challenges on smartphones is encryption and steganography. The user will see the practice of steganography hiding information or information in a hidden medium; For example; Picture, sound, video for most of the existing procedures with Intruders are exposed to the keys and succeed in choosing them required -bits. First, a 128-bit encryption key is generated using the AES algorithm from the cryptography algorithms. A 128-bit encrypted key is also used. Select the 128bits Image to hide the secret message again. A stego file is created using the LSB algorithm. This research study found that the combination of cryptography and steganography on smartphones makes it more secure than the other. This system uses an android studio, a USB cord, and a smartphone.

1. Introduction

Information security prevents unauthorized access to any type of information. Information technology (IT) industries are designed to protect against data and information theft through the method of confidentiality and concealment of information. One of the most popular methods is steganography. Secure steganography between secret senders and recipients is to hide the information in the images, videos, and texts via broadcast channels. The purpose is to ensure the confidentiality of the information transmitted through. There are two security steps: avoiding detection and decryption. Avoiding search covers both software and visual search. Alternatively, the secret text needs to be embedded in the image. So, the custom is not obvious. The intruder must be confused. The cover image may contain any confidential information and cover the part where a line is discovered by any chance must decrypt. The image carries confidential information that intruders could not afford. (2010 Maz)

In steganography, the hiding techniques will fail if an intruder can suspect the cover medium. While using a steganography tool, one should ensure whether the hiding technique used by the tool is prevailing all sort of detection by the intruder. In some cases, after hiding an information or data in an image, people can understand that there is some distortion in the image. The more the success rate of avoiding detection of a steganography tool is, the more trustworthy the tool will be. (2012 Sam)

2. Literature Review

Umamaheswari Sivasubramanian and Pindarian (2010) described the analysis of different algorithms for data hiding. They highlighted some of the most used steganography algorithm briefly such as blindside, pixel swap, hide, seek, filter first, and battle stage. After using two types of filters (i.e. Laplacian and Sobel), the best image zone to hide has been found. These are followed by a comparative analysis of performance among algorithms in terms of machine learning accuracy as a function of embedding rate. It concludes with the proposal and mathematical representation of a plausible deniability scheme.

Karim et al. (2011 car) focused on the aspirations to improve existing LSB strength one-way option to add one-level security to secure code, secrets in a persistent way. The red and blue channels are used as indicators, and the green and blue channels are used as data. Channel based on the secret information and LSB channel uses the Red, the secret information uses green or blue channels. Intruders cannot easily extract secrets. Information without a valid password. Also, the experimental results are better and stronger image quality.

The methods discussed so far produce images of low-quality products that can be easily discovered using HVS. Also, the information is included in the cover image without encryption. It can be blocked by an attacker. Studies that address these issues. In addition, we present a plan to improve the quality of stego images.

Open Stego is an open-source steganography software distributed under the GNU General Public License v2.0. In this software, the author has identified two main aspects: confidentiality and use of beta machines. This is a Java tool encrypts data based on the password security layer. The DES algorithm is used for data encryption. DES key with MD5 hashed the password provided. It uses a plugin-based plugin. Different plugins can be created for different kind of steganography algorithms.

Overall, according to the existing literature, the most popular stealth technology is LSB, the best hiding place. The location can be found using filters and laplace equations. The work according to the review survey, there are different approaches to the algorithms. Areas of the most suitable places (e.g. some places are best) that is some are best for watermarking and some are best for steganography).

3. Methodology

This section discusses the related prior works of different steganography data hiding techniques and cryptography algorithms.

In the study, two major parts are divided to complete techniques. Firstly, a secret message is embedded in a cover image to avoid the detection of data using the LSB algorithm and a new approach of choosing the pixel index in the cover image. On the other hand, to avoid decryption using AES -128 bits encryption technique to encrypt the secret text. (2017 3rd ICE ICT)

Algorithm 1: Algorithm for embedding Data in the cover image

Step-1: Take input of secret text, cover image, and a key.
Step-2: Check image quality according to the text. If not good enough, then give the warning to increase.
Step-3: Get the byte array from the image and from a pixel array using the only blue color value of each pixel of the image.
Step-4: Encrypt the given secret text using the AES algorithm with the given key known as ciphertext.
Step-5: Apply Algorithm 3 on the pixel byte array.
Step-6: Save the Steg Object.

Secondly, the next major part is to extract secret data from the image. The counterpart of the previous algorithm extracts the hidden data. In this case, we give input of the key with the stego image known as stego-object and use AES-128-bit encryption mode as shown in algorithm 2.

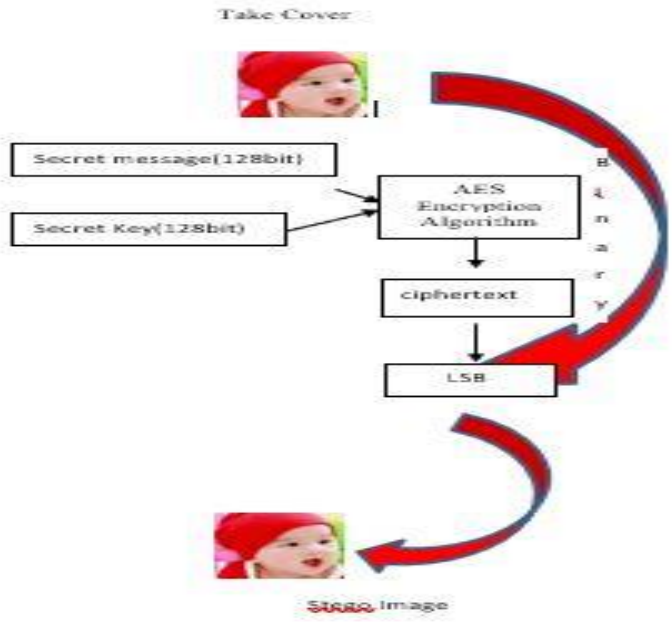

Figure 1. Proposed System Design for Embedding Process

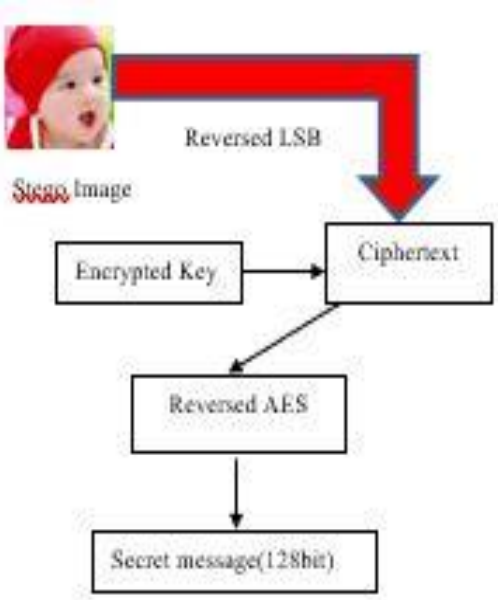

Figure 2. Proposed System Design for Extraction Process

Algorithm 2: Algorithm for extracting hidden data from the stego-object

Step 1: Take inputs: stego-object and the key. Step 2: Apply the counterpart of the Algorithm 3 to get hidden bitstream.
Step 3: Get the char value of the bitstream.
Step 4: Decrypt using AES decryption using the key.
Step 5: Outputs the secret Text.
For algorithm 1 and algorithm 2 we need to find the index of cover image pixel (Algorithm3).

Let the ciphertext which will be converted in the cipher bitstream is, 101110101 Let the byte array of an 8-by-8 image is the following figure 3. (2017 3rd ICE ICT)

At first, (row, column) = (i, j) = (1, 1) = first binary
value of the byte array (10110111) of Figure 3. Next, the ciphertext in-depth bit of the cipher bitstream is 1, so insert 1 in the LSB of the value of (1, 1). As the LSB of 10110111 is 1, so no change takes place. If the LSB would be 0, then the user would change it into 1.

	j=	1	2	3	4	5	6	7	8
i =	1	10110111	00111001	10101010	11110000	10101010	01010101	11100111	10110100
i =	2	10111101	01111000	10011001	00010101	10110101	01010101	10111001	10001010
i =	3	11010111	10110101	10110001	10101010	10110001	11010011	10110011	11101001
i =	4	10010010	10010010	10110110	10110010	11010010	10010101	01011001	01001010
i =	5	10101010	11010101	10101011	11010101	10110110	10001001	10101010	01001100
i =	6	01010011	10100101	01010100	00000010	00000000	10010101	00011110	00101001
i =	7	10101111	10101110	10101011	00111101	10101001	10111010	10001010	10100001
i =	8	00110101	01010111	11010111	01010010	11111111	01011111	00010101	10100100

Algorithm 3: Pseudo algorithm for finding index of pixel
from image and inserting cipher bits Input: image, secret text
Initialization: set: i=1, j=1, jumping series index = 1,
zero series index = 1, cipher bits index = 1,
jumping series = [2 3 2 3 3 3 2 2], zero series = [7 3 5 4 2 6 1];
Find: cipher text = aes (secret text); cipher len = length
(cipher text); [row, col] = size (image);
while *i ≤ row* do
while *j ≤ col* do
insert cipher text [cipher bits index] in the LSB
of image [i, j];
cipher bits index ++;
if *cipher bits index > cipher len* then
break;
end
n1 = jumping series [jumping series index] ;
if *n1 == 2* then
Take decimal value of 2 MSB bits by setting:
n = image [i, j] / 64;
else
Take decimal value of 3 MSB bits by setting:
n = image [i, j] / 32;
end
if *n == 0* then
n = zero series [zero series index];
if *zero series index < length (zero series)*
then
zero series index ++;
else
zero series index = 1;
end
end
if *jumping series index < length*
(jumping series) then
jumping series index ++;
else
jumping series index = 1;
end
update: j = j + n;
end
update: i= i + 1;
update: j = n;
if *cipher bits index > cipher len* then
break;
end
end

AES Algorithm Calculation
Three types of AES algorithms, namely AES-128, AES-192, and AES-256 and have a key extension, and then these blocks run on a byte array and are arranged as 4 × 4 matrices called state. In the encryption process, data are transmitted through 10, 12, and 14 circles. The following steps are calculated for AES-128-bit encryption;

- Substitute byte
- Shifting lines

- Mix columns
- Add round button

And decryption process, the following steps are calculated;

- Inverse shift row
- Inverse substitute byte
- Add round key
- Inverse mix columns

Substitute byte Procedure

Substitute byte transformations are linear byte substitutions that operate independently of a replacement table called the S-box. The independent replacement byte count is displayed as follows;

	0	1	2	3	4	5	6	7	8	9	0a	0b	0c	0d	0e	0f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
40	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

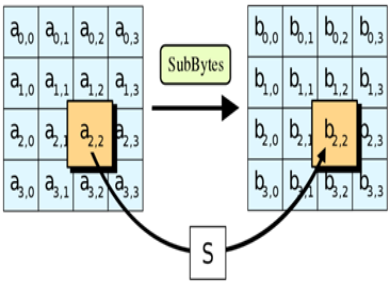

Figure 4. Operation of substitute byte (2020 AES Wik)

Shift rows Procedure

In the Shift Rows operation, the bytes in the last three rows of the State are cyclically shifted left over different numbers of bytes.

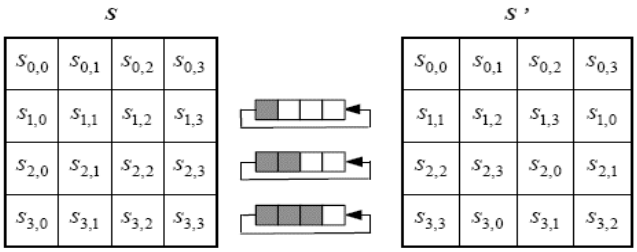

Figure 5. Operation of Shift Row (2020 AES Wik)

Mix columns Procedure

Mixed columns treat each column as a multi-country column by four terms. Columns are considered polynomials above GF (2 ^ 8) and multiplied by certain polynomials a (x) given by modulus x 4 + 1,

$$a(x) = [03]\, x^3 + [01]\, x^2 + [01]x + [02] \qquad eq^n (1)$$

The resultant columns are shown in the figure below. This is the operation of mix columns.

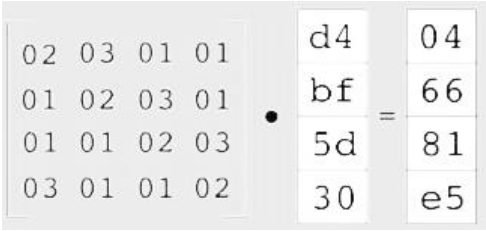

Figure 5. Mix Column (2020 AES Wik)

Add round button Procedure

In ad round key conversion, the round key is added to the status, with a simple bitwise XOR activity. The Round key is derived from the cipher key through the original scheduling process. State and round keys are the same sizes and subsequent state XOR activities are performed as components;

$$b(i, j) = a(i, j) \oplus k(i, j) \qquad eq^{n} (2)$$

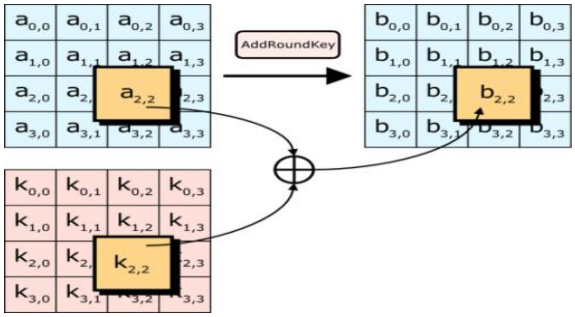

Figure 6. Operation of Shift Row (2020 AES Wik)

4. Experimental Results

The system needs to run the following requirements;
Hardware Requirements
Processor PC with a Core i3 processor
RAM 4GB(Minimum)
Hard circle 320GB
Other SmartPhone

Software Requirements
Android Studio 3.6 and above (JDK Toolkits)
In this system, the user must write command the following;
setting>>additional setting>>developer option >>USB debugging (on) >> OK >> install via USB (on)
Thus, the proposed apk " Helloworld" appears on the smartphone.

If the system users run this system, the GUI will appear. In the "Home View" includes two Buttons such as "ENCODE" and "DECODE". If the "ENCODE" Button is clicked, the user will see " CHOOSE IMAGE" Button to choose an image (*. JPG) image, MESSAGE to encrypt the plaintext message 128bit and also encrypted key 128 bits exactly. (i.e. 128 bits are equal to 16 words). This process is an embedding process to intend the third party cannot spoil the message. The authorized person can know the secret key. The message is encrypted with the key based on AES (Advanced Encryption Standard) and hide with LSB (Least Significant Bit).

Figure 7. Proposed APK

Figure 8. Home Android View

If the user chooses Input Image to browse from the phone, and the secret key (128bits) will type, and the message (128bits) wanted to encrypt. Then the user must click the encode button and save the stego image to get.

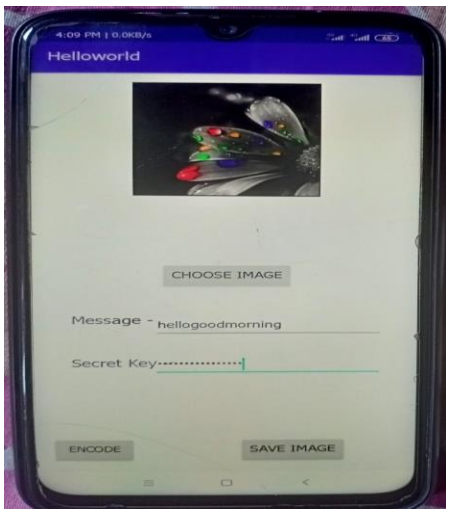

Figure 8. Encryption Site

Figure 9. Decryption Site

The user must choose the stego image and the secret key will type again right. Therefore, Figure 10 will appear in the secret message.

Figure 10. Recovered Message

5. Conclusion

In this research study, image steganography is to double data security detection. The combination of AES and LSB algorithms is to get a new technique. An intruder will face extreme difficulty to reveal the message due to the procedure of dynamically choosing the bits to hide as a function of the values of pixel bits. This app is a built-in android platform and designed by an android studio. In the operating system Windows 10, this app has been developed. In the future, this android app will be more developed and thus it will be more reliable and useful.

References

[1] Chaeikar, S. S., Manaf, A. B. A., & Zamani, M. (2012). Comparative analysis of Master-key and Interpretative Key Management (IKM) frameworks. *Cryptography and security in computing*, 203.

[2] Karim, S. M., Rahman, M. S., & Hossain, M. I. (2011, December). A new approach for LSB based image steganography using secret key. In *14th international conference on computer and information technology (ICCIT 2011)* (pp. 286-291). IEEE.

[3] Umamaheswari, M., Sivasubramanian, S., & Pandiarajan, S. (2010). Analysis of different steganographic algorithms for secured data hiding. *IJCSNS International Journal of Computer Science and Network Security*, *10*(8), 154-160.

[4] Zamani, M., Manaf, A. A., Ahmad, R., Jaryani, F., Taherdoost, H., Chaeikar, S. S., & Zeidanloo, H. R. (2010). A novel approach for genetic audio watermarking. *Journal of Information Assurance and Security*, *5*(1), 102-111.

[5] Umamaheswari, M., Sivasubramanian, S., & Pandiarajan, S. (2010). Analysis of different steganographic algorithms for secured data hiding. *IJCSNS International Journal of Computer Science and Network Security*, *10*(8), 154-160.