Triple-A: Secure RGB Image Steganography Based on Randomization

Adnan Gutub, Ayed Al-Qahtani, Abdulaziz Tabakh

Computer Engineering Department, KFUPM, Dhahran 31261, SAUDI ARABIA
{gutub@kfupm.edu.sa, ayedsaad@gmail.com, atabakh@kfupm.edu.sa }

***Abstract*—A new image-based steganography technique – called triple-A algorithm - is proposed in this paper. It uses the same principle of LSB, where the secret is hidden in the least significant bits of the pixels, with more randomization in selection of the number of bits used and the color channels that are used. This randomization is expected to increase the security of the system and also increase the capacity. This technique can be applied to RGB images where each pixel is represented by three bytes to indicate the intensity of red, green, and blue in that pixel.**

Key-Words: Steganography, randomization, RGB Bitmap images, Triple-A Algorithm, Computer Security.

I. INTRODUCTION

Steganography is the art of hiding information into another covering media in a way that nobody except the receiver can detect the secret message and retrieve it. Steganography (which means “covered writing” in Greek) is an old art that has been used since the golden age of Greece where some practices were recorded like: writing a message on a wooden table then covering it with wax, and tattooing a messenger hair after shaving and then let his hair grow up before sending him to the receiver where his hair was shaved again. Other techniques use invisible ink, microdots, converting channels and character arrangement [1,5,6,7].

Digital steganography has many applications in today’s life. It could be used as a digital watermarking to protect the copy-rights, or to tag notes to digital images (like post-it notes attached to paper files), or to maintain the confidentiality of valuable data from possible sabotage, theft, and unauthorized viewing.

Image-based steganography techniques need an image to hide the data in. This image is called a cover media. Digital images are stored in computer systems as an array of points (pixels) where each pixel has three color components: Red, Green, and Blue (RGB). Each pixel is represented with three bytes to indicate the intensity of these three colors (RGB). Some techniques have been used for image steganography such as LSB, SCC, Pixel Indicator [1] and image intensity [7]. In LSB, the least significant bit of each pixel for a specific color channel or for all color channels is replaced with a bit from the secret data. Although it is a simple techniques, but the probability of detecting the hidden data is high. SCC technique is an enhancement. The color channel, where the secret data will be hidden in, is cycling frequently for every bit according to a specific pattern. For example, the first bit of the secret data is stored in the LSB of red channel, the second bit in the green channel, the third bit in the blue channel and so on. This technique is more secure than the LSB but still it is suffers detecting the cycling pattern that will reveal the secret data. Also it has less capacity than the LSB. Pixel indicator technique is another image steganography technique where the least two significant bits of specific color channel is used to indicate the existence of secret data in the least significant two bits of other two channels according to rules detailed in [1]. A new idea of RGB steganography is proposed in [7]. It is based on color intensity, which is outside the focus of this work.

Even though pixel indicator technique adds some randomization to harden the detection of the secret data, its capacity varies depending on the actual values of the indicator channel so the actual capacity is unpredictable. Our suggested technique tries to add more randomization to the selection of the pixels in which the secret data is stored, affecting the number of bits used to keep the secret data, and the channels that are used to store the secret data.

In section 2, we will explain the proposed triple-A technique, and then we will give an overview of the implementation and the experimental results in section 3. We will compare our algorithm with some existing algorithms in section 4. Finally the conclusion is given in section 5.

II. TRIPLE-A: PROPOSED ALGORITHM

Figure 1 shows the Triple-A algorithm taking the message (M), the carrier image (C), and the password based generated

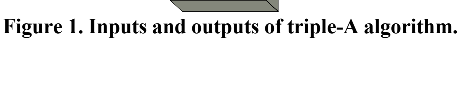


Figure 1. Inputs and outputs of triple-A algorithm.

key (K) depending on password (P), as inputs and produces the message (M) hidden inside the carrier image (C).

This algorithm can be divided into two major parts, Encryption and hiding as it is shown in Fig. 2.

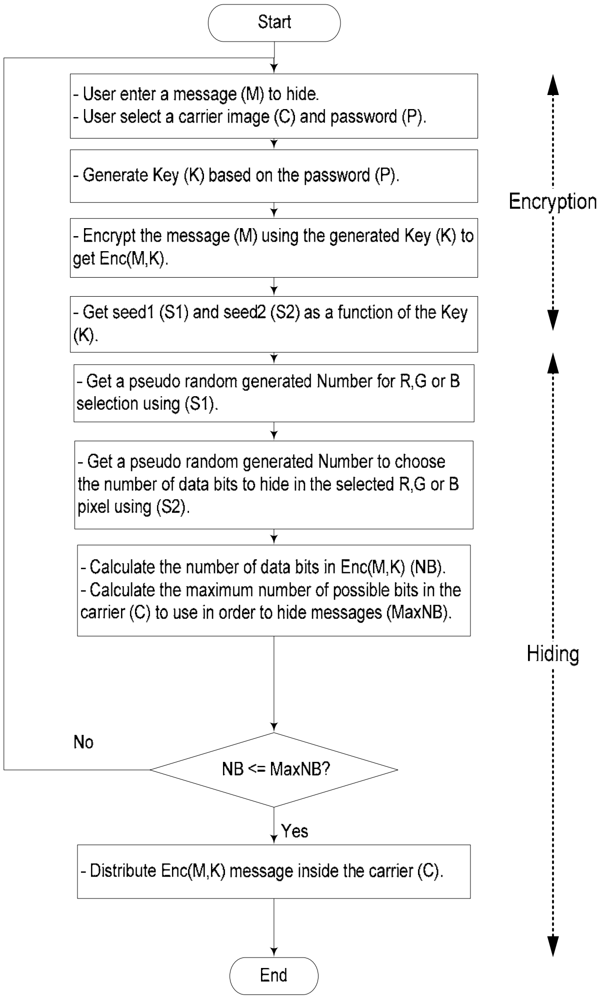

Start
- User enter a message (M) to hide.
- User select a carrier image (C) and password (P).
- Generate Key (K) based on the password (P).
- Encrypt the message (M) using the generated Key (K) to get Enc(M,K).
- Get seed1 (S1) and seed2 (S2) as a function of the Key (K).
- Get a pseudo random generated Number for R,G or B selection using (S1).
- Get a pseudo random generated Number to choose the number of data bits to hide in the selected R,G or B pixel using (S2).
- Calculate the number of data bits in Enc(M,K) (NB).
- Calculate the maximum number of possible bits in the carrier (C) to use in order to hide messages (MaxNB).
NB <= MaxNB?
No
Yes
- Distribute Enc(M,K) message inside the carrier (C).
End
Encryption
Hiding

Figure 2. Flow chart of triple-A algorithm.

In part one we are interested in encrypting the message (M) using AES algorithm which will produce Enc (M, K). In our implementation the key K can be generated from a set of user passwords each with a specific key using simple XOR. This will add more security especially when it is necessarily to make the secret message available only if all the users present their passwords.

In part two, the RGB Image is used as a cover media. It utilizes the advantage of the Bmp images, where every pixel is independent from the reset of the image file. Enc (M, K) is hidden according to our triple-A algorithm of Fig. 2 which needs to have a pseudorandom number generator (PRNG). The assumption for PRNG is to give two new random numbers in every iteration. The seeds of these PRNGs namely Seed1 (S1) and Seed2 (S2) are formed as a function of the Key (K). S1 is restricted to generate numbers in [0, 6] while S2 is restricted to the interval [1, 3]. S1 random number is used to determine the component of the RGB image which is going to be used in hiding the encrypted data Enc (M, K). Table 1 shows how (S1) random number selects the RGB components. On the other hand, (S2) random number determines the number of the component(s) least significant bits that is used to hide the secret data. On the same way Table 2 shows how (S2) random number determines the number of component bits. Bmp images are represented in computer systems as a two-dimensional array of X-position and Y-position. X-position and Y-position for the pixels with hidden data is distributed inside the image according to the size of the secret data Enc (M, K) and the carrier image (C) computed.

Table 1. Seed 1 random number usage.

	Random number	Meaning to the algorithm
1st PRNG	0	use R.
	1	use G.
	2	use B.
	3	use RG.
	4	use RB.
	5	use GB.
	6	use RGB.

Table 1 shows that the maximum number of bits used from a component are 3 bits. While table 2 shows that the maximum number of bits, which is used to determine the component bits, are 2 bits. This indicates that the algorithm may add up to a maximum of ± 7 to the value of the color component(s) in that pixel.

Table 2. Seed 2 random number usage

	Random number	Meaning to the algorithm
2nd PRNG	1	use 1 bit of the component(s).
	2	use 2 bit of the component(s).
	3	use 3 bit of the component(s).

Also, by combining data from the previous tables, we can see that the minimum number of bits used in each pixel is 1 if we use only one bit of one chosen components of the RGB image. The maximum is 9 bits if we used all the three components with three bits.

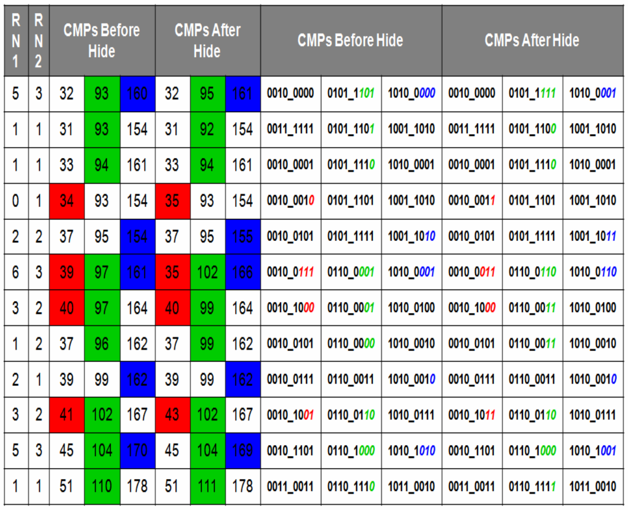
Table 3: shows an example of hiding five consecutive bytes of Enc (M, K) inside a carrier image shown in Fig. 3. These bytes are: 0x15, 0xF9, 0xCD, 0x5B, 0x09 = 0000_1111, 1001_1111, 1100_1101, 0101_1011, 0000_1001.

III. Implementation Overview

The Triple-A algorithm is increased the capacity ratio and the security level of the concealment operation. Theoretically, the average number of bits used per pixel is equal to 3.428 where the maximum number of bits used in SCC could be 3 and for LSB is 1. This shows us that the capacity of the new proposed technique is higher than the previous techniques.

By using this algorithm, the ratio between the number bits used inside a pixel to hide part of the secret message; and the number of bits in the pixels itself, which defined as the capacity factor can be in the range from 1/24 to 9/24 if we use a maximum of 3 bits as it is suggested by table 3. Moreover, if we extends the algorithm to hide 4 and even 5 bits the factor can be increased up to 15/24 which is above half of the pixel bits, but the down side is the additional noise introduced as the number of bits used to hide the secret data get higher.

Table 3. Triple-A Random-based image steganography example.



RN1	RN2	CMPs Before Hide			CMPs After Hide			CMPs Before Hide			CMPs After Hide		
5	3	32	93	160	32	95	161	0010_0000	0101_1*101*	1010_0*000*	0010_0000	0101_1*111*	1010_0*001*
1	1	31	93	154	31	92	154	0011_1111	0101_110*1*	1001_1010	0011_1111	0101_110*0*	1001_1010
1	1	33	94	161	33	94	161	0010_0001	0101_111*0*	1010_0001	0010_0001	0101_111*0*	1010_0001
0	1	34	93	154	35	93	154	0010_001*0*	0101_1101	1001_1010	0010_001*1*	0101_1101	1001_1010
2	2	37	95	154	37	95	155	0010_0101	0101_1111	1001_10*10*	0010_0101	0101_1111	1001_10*11*
6	3	39	97	161	35	102	166	0010_0*111*	0110_0*001*	1010_0*001*	0010_0*011*	0110_0*110*	1010_0*110*
3	2	40	97	164	40	99	164	0010_10*00*	0110_00*01*	1010_0100	0010_10*00*	0110_00*11*	1010_0100
1	2	37	96	162	37	99	162	0010_0101	0110_00*00*	1010_0010	0010_0101	0110_00*11*	1010_0010
2	1	39	99	162	39	99	162	0010_0111	0110_0011	1010_001*0*	0010_0111	0110_0011	1010_001*0*
3	2	41	102	167	43	102	167	0010_10*01*	0110_01*10*	1010_0111	0010_10*11*	0110_01*10*	1010_0111
5	3	45	104	170	45	104	169	0010_1101	0110_1*000*	1010_1*010*	0010_1101	0110_1*000*	1010_1*001*
1	1	51	110	178	51	111	178	0011_0011	0110_111*0*	1011_0010	0011_0011	0110_111*1*	1011_0010

A: original carrier.

B: carrier with secrete using SCC or Triple-A algorithms.

Figure 3. Image steganography testing example.

The above paragraph is related to pixels point of view. If we look at the image as a whole, since we used PRNG, we should average the number of bits used to hide the secrete data over the image carrier. Thus, the capacity ratio is = (Number of bits used each possible case)/ (Total number of cases * 24). Since we have a total of 21 cases decomposed as:

- Using One component case: here we have 3 ways to determine the bits * 3 ways to decide the component R, G or B. this results in 9 cases.
- Using Two component case: here we have 3 ways to determine the bits * 3 ways to decide the component RG, RG or GB. This results in 9 cases.
- Using Three component case: here we have 3 ways to determine the bits * one way to decide the component which is RGB. This results in 3 cases.

The average capacity ratio is around 1/7 or 14% of the original cover media size. This is better than SCC algorithm in which the capacity ratio is 1/24 or 4%.

The security can be considered two layers the hiding layer and the encryption layer. Layer one is the hiding part of triple-A algorithm shown in Fig. 2. Notice that the secret data is scattered throughout the whole image. Also, extracting the secret data without the Knowledge of seeds is almost impossible.

On top of layer one, layer two uses AES algorithm to encrypt the data. Therefore, even if the attacker knows how to extract the data from the image it is still encrypted.

Our algorithm has the same unpredictable message size as the pixel indictor scheme but the former has maximum capacity ratio better than the latter. Also, the unpredictability in pixel indicator is function of the image carrier (C) which is usually has mega sizes. Triple-A in the other case depends on the key (K) which is of smaller size.

IV. EXPERIMENTATIONS & COMPARISONS

Triple-A algorithm is implemented using software package developed using C#. The tool encrypts the data before hiding it using ASE encryption scheme. The resulting stego-images is tested and compared with the original images by using histograms generated by MATLAB to check the level of noise or distortion caused by the Triple-A algorithm.

The results are compared with other stego-images generated using SCC algorithm. The level of distortion and the capacity issues are highlighted.

Figure 3 shows an original carrier compared to the same carrier with secret using both SCC and Triple-A algorithm. From the first moment, you can not see difference within the images; but the histogram of the images shown in Fig.4 shows a minor different in the value of the components: R, G and B.

The example in Table 3 is taken from the first few pixels of Fig. 3. The table shows that the capacity ratio is around 13.2% with a difference of 5% compared to the calculated average.

Table 4. Comparing Triple-A and SCC.

Size of M (bytes)	SCC to hide M inside C		Triple-A to hide M inside C	
	Pixels used	capacity	Pixels used	capacity
28 KB	27984	4.16%	7169	16.27%

Table 4 shows a comparison between Triple-A algorithm and SCC. The result is obtained using different carrier images and averaging the number of pixels used in the hiding operation. It is shown that triple-A algorithm enhance the capacity ratio with a factor of around 4. The table shows that the capacity ratio of Triple-A is around 16.27% with increasing of 20% compared to the calculated average.

SCC algorithm has a fixed small capacity ratio equal to 1/24 while our implementation of triple-A algorithm results in a moderate increase of capacity ratio without affecting the image with noise or distortion.

The secret message has retrieved correctly using both methods. As a stego-analysis procedure, the hidden data of SCC can be easily extracted once you know that there is a possible hidden data in the cover media. This is not the case of Triple-A which is more difficult to guess.

To clarify the minor differences in Fig. 3 and 4 another Red image carrier is used to hide small size data. This Red image did not show any difference between figures of the image having secret data and not, which is considered an advantage. But the histogram of SCC algorithm introduced a clearer different other than Fig. 3 and 4. Note that the red image study had all the pixels in R component histogram with range up to the maximum value of 255, while G and B components are fixed to 0's. The SCC uses only the least significant bit which justifies the new column or spike in 254 at the histogram.

On the other hand, Triple-A algorithm uses up to 3 bits which introduces difference up to ±7; this justifies the spikes near the 255 in the histogram of triple-A algorithm.

V. Conclusion

Triple-A concealment technique is introduced as a new method to hide digital data inside image-based medium. The algorithm adds more randomization by using two different seeds generated from a user-chosen key in order to select the component(s) used to hide the secret bits as well as the number of the bits used inside the RGB image component. This randomization adds more security especially if an active encryption technique is used such as AES. The capacity ratio is increased above SCC and pixel indicator scheme. Triple-A has a capacity ratio of 14% and can be increased if more number of bit is used inside the component(s).

As a final note, we can say that SCC algorithm is a special case of Triple-A algorithm if the number of bit used is fixed and equal 1 and Seed2 is restricted to [0,2] with circular effect.

VI. Acknowledgment

The authors would like to thank the Computer Engineering Department and King Fahd University of Petroleum and Minerals (KFUPM) for supporting this work.

VII. References

[1] Adnan Gutub, Mahmoud Ankeer, Muhammad Abu-Ghalioun, Abdulrahman Shaheen, and Aleem Alvi, "Pixel Indicator high capacity Technique for RGB image Based Steganography", *WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications*, University of Sharjah, Sharjah, U.A.E. 18 – 20 March 2008.

[2] Donovan Artz, Los Alamos National Laboratory, "Digital Steganography: Hiding Data within Data", *IEEE Internet Computing*: May, 2001

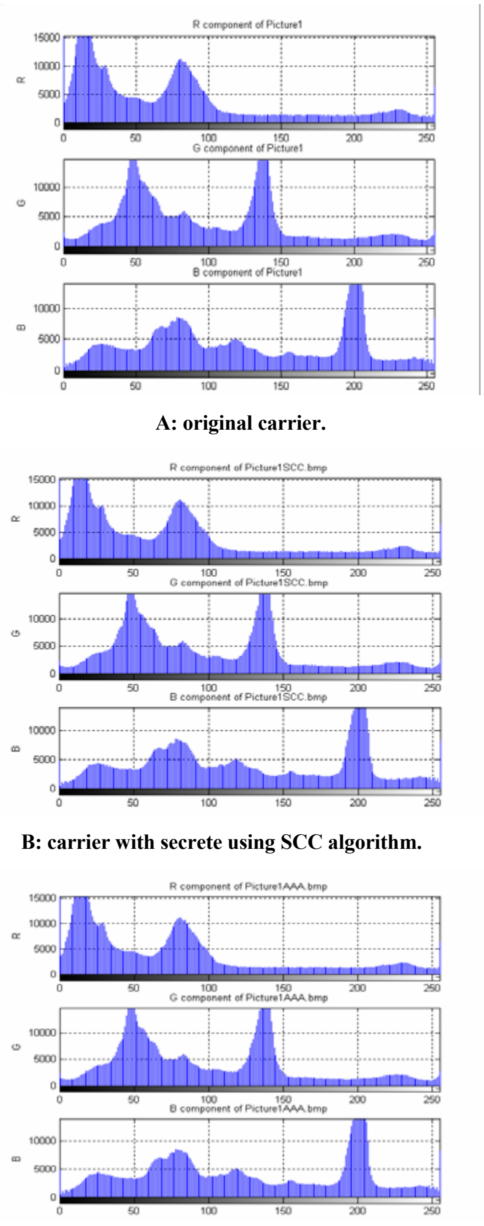
[3] Neil F. Johnson. Sushil Jajodia, "Exploring Steganography: Seeing the Unseen", *IEEE computer*, 1998.

[4] Kevin Curran, Karen Bailey, "An Evaluation of Image Based Steganography Methods", *Multimedia Tools and Applications*, Vol. 30, No. 1, Pages: 55 – 88, July 2006.

[5] Adnan Gutub, Lahouari Ghouti, Alaaeldin Amin, Talal Alkharobi, and Mohammad K. Ibrahim, "Utilizing Extension Character 'Kashida' With Pointed Letters For Arabic Text Digital Watermarking", *International Conference on Security and Cryptography - SECRYPT*, Barcelona, Spain, July 28 - 31, 2007.

[6] Adnan Gutub and Manal Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", *WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE)*, Vienna, Austria, May 25-27, 2007.

[7] Mohammad Tanvir Parvez and Adnan Gutub, "RGB Intensity Based Variable-Bits Image Steganography", *APSCC 2008 – Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference*, Yilan, Taiwan, 9-12 December 2008.



A: original carrier.

B: carrier with secrete using SCC algorithm.

C: carrier with secrete using Triple-A algorithm.

Figure 4. Image steganography histograms.