An Intelligent Fibonacci Approach to Image Steganography

Shreyank N Gowda
Indian Institute Of Science
Bengaluru, India
kini5gowda@gmail.com

***Abstract*—Steganography is the art of hiding information in a medium. Steganography tends to leave little doubt about the fact that information is being hidden. In this paper an intelligent approach to image steganography is proposed. First, a key is passed using an algorithm such as Diffie-Hellman. This key is used to encrypt the text using an enhanced Caesar Cipher algorithm which provides security and is extremely quick in terms of speed of execution. Next, this key is divided by 10 to get a single digit remainder, say x. x is then used to determine which pixel value is going to be modulated. x is passed as the parameter to a Fibonacci p-code number system. The output obtained say m is taken as the first pixel to be modulated, afterwards all multiples of m such as 2m, 3m etc are modulated. Once all the multiples of m are completed, the last multiple is taken and divided by 10 and the process is repeated till complete data is hidden.**

***Index Terms*—Steganography, Fibonacci, Diffie-Hellman, Security**

I. Introduction

This Steganography is the field of study of hiding of information using a secure medium. When an image is used as the medium, it is referred to as image steganography. Steganography is a field that has been gaining a lot of attention in recent times. The main reason for that is the better capability to hide information than say cryptography. Cryptography encrypts information, which in turn will depict that information is actually being hidden. However, steganography conceals the fact that information is being hidden itself.

Transferring of information is becoming extremely difficult. The number of hacking related cybercrimes is always increasing. The older algorithms are becoming easier to crack using the increased computational power of the latest devices. Attacks like brute-force are done at an exponentially lower rate with this enhanced computing ability. Cryptographic algorithms like AES, DES, Blowfish etc are no more near as secure as they once were.

Steganographic algorithms are also not much safer. Recent steganalysis research has shown the ability to use deep learning to crack steganographic algorithms [1]. This would reduce the time needed to crack the algorithm by a remarkable amount. This constantly increasing computational power along with introduction of machine learning concepts to attack steganographic algorithms has resulted in researchers trying to make their proposed algorithm as complex as possible. This has led to an increased amount of time in terms of execution. This time-security trade-off is yet to be perfected.

978-1-5090-6255-3/17/$31.00

An algorithm which is quick in its execution tends to be more easily breakable. However, an algorithm that is very secure leads to a time constraint. This time constraint is also extremely difficult to handle.

Another recent trend has seen the combination of cryptography algorithms with steganography algorithms. This makes the information more secure. Even if the steganography algorithm is broken, there is the cryptography algorithm protecting the information.

Cryptographic algorithms help to enhance the security of the information by converting them into an encrypted form. Meanwhile, steganographic algorithms use this converted secure form to further enhance the security of the algorithm. Recently, there has been developments in terms of cracking open cryptographic algorithms by means of either deep learning or using high performance computing. To overcome this we need to add additional security to information.

To add the additional security we can either make the algorithm more complex to crack, which means even the encryption increases time of execution of the algorithm. Another option is to add another level of security by using a simple but secure algorithm. This has been the basis for most recent information security algorithms. A fine line between time and security has to be balanced and for this we use a combination of simple but effective algorithms.

II. Related Work

In [2] quantum steganography was proposed. It is a technique in which information was hidden into quantum covers called quantum images. Two LSB algorithms were proposed in the form of quantum circuits based on the novel enhanced quantum representation. The capacity and robustness of the algorithm could be adjusted based on the requirements on hand.

In [3] a block based approach was proposed to image steganography. The data to be hid was considered a block. Each block was broken down to n smaller blocks based on user input. These smaller blocks then contained a smaller amount of data. The blocks were then sent in a random order to the receiver. This increased the chaos factor of the algorithm and hence the security.

In [4] the message bits were embedded in select parts of the cover image. The pixels which held the data were called energetic pixels. These pixels were locations in the image in which where a perceptible change in intensity of pixel values occur. This selective process ensured a higher layer of security.

In [5] the author shows the use of texture as a cover medium. Since a texture is nothing but a set of repeating pixels, the author shows it is extremely efficient to use a texture to represent the same as it can accommodate any amount of data by just increasing the texture dimensions.

The authors in [6] proposed a combination of algorithms to enhance the security of the proposed algorithm. First a 3-DWT algorithm was used to determine pixel locations, next Lorenz chaotic encryption was used to enhance security of the data. This added 2 layers of security to the proposed approach.

In [7] an algorithm called magic-LSB was proposed. The method was based on the V plane of the HSV color metric. The input RGB image is taken and converted to a HSV image after transposition. The V-plane is divided into sub-images which then contain blocks of information. Encryption is done using a Multi-Level Encryption algorithm.

In [8] the image itself was divided into blocks. A mix column transform was developed with the help of irreducible polynomial mathematics. This transform was used on each block of the image.

In [9] a combination of cryptographic algorithms was used along with a block based approach. The entire data to be hidden was encrypted first using the Blowfish algorithm. The key for the blowfish algorithm was encrypted further using RSA. The entire encrypted block was then broken down and sent in a random order to increase the security of the algorithm.

In [10] RC4 technique was used to determine a pseudo random position to embed the secret data. The embedding was done using the Variable Least Significant Bit Algorithm. The entire image was broken down to blocks and for each block the RC4 technique was used to determine a random position.

In [11] edge detection was performed and then LSB was used for embedding the data. The technique proposed would also embed the login details and time stamp on to the secret image without causing visual distortion. In [12] DWT was first used to get the 4 frequency layers in the image and the lowest frequency layer was chosen. After hiding the information, the image is compressed. For recovery of the original image inverse DWT was used.

A novel algorithm was proposed in [13] where a combination of two algorithms were used. First, Bit Plane Complexity Segmentation (BPCS) analysis was utilized. Next, QR Decomposition (linear algebra) was done to determine the region in which the entire secret information is embedded. This region is determined to be having high security.

The method in [14] proposed a novel symmetric-key based image concealing algorithm. Pseudo-random keys were obtained with the help of a 1-D logistic map. These pseudo-random keys are then utilized to determine the pixel positions of the cover image in which the data will be embedded.

A novel distortion function was proposed in [15]. This function depended upon the magnitude of discrete cosine transformation coefficients (DCT), the 1st and 2nd order residuals and the total sum of zeros within a single DCT block. The magnitude part of a DCT block helps to indicate the ability of the current coefficient to conceal modification trace.

III. Proposed Algorithm

Before The steps of the proposed algorithm is given below. Provably secure Diffie-Hellman is used for key exchange during the algorithm [16]. The key once obtained is further used for 2 aspects of the algorithm. The first is the encryption part of the text and the second is the pixel location choosing using the p-Fibonacci series. The encryption done needed to be quick and add a layer of security. For this we used an enhanced Caesar Cipher algorithm [17]. The p-Fibonacci series is defined as shown below,

$$F_p(i) = \begin{Bmatrix} 0 \quad \forall \quad i < 0 \\ 1 \quad \forall \quad i = 0 \\ F_p(i-1) + F_p(i-p-1) \quad \forall i > 0 \end{Bmatrix} \quad (1)$$

Here, p is a non negative integer which decides the sequence of values given to a paricular series. Table I shows the pFibonacci series for p-codes from 0 to 3. p can have any value, however we have restricted it between 0 and 9.

TABLE I
Fibonacci p-codes

S.No	p-code	p-Fibonacci series
1	0	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024
2	1	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233
3	2	1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129, 189
4	3	1, 1, 1, 1, 2, 3, 4, 5, 7, 10, 14, 19, 26, 36, 50, 69, 95, 131, 181, 250

Now, when a key is obtained we first use the key to perform the enhanced Caesar Cipher algorithm. For this, the key is made to mod with 26 and then further used in the algorithm. The main reason for using the algorithm is to ensure the time for encryption is less. Using algorithms like Blowfish or AES will take away the main essence of reducing time complexity of the proposed algorithm.

Next, the original key is divided by 10 to obtain a remainder in the range of 0 to 9. Using this remainder, we find the value of which p-Fibonacci we needed to consider. Say, we obtained a remainder r then r-Fibonacci series is taken and all pixels which are multiples of the numbers in the series are taken and modulated. Once all the multiples are completed we check if data to be hidden still remains. If that is the case, we take the last pixel number which was modulated and consider it as the key and repeat the procedure. Also, we check if the new remainder is equal to r. If that is the case, we add 1 to the new remainder and if this value is 10 we convert it to zero. Following are the steps of the algorithm,

Step 1: Start

Step 2: Obtain key (k) using Provably secure Diffie-Hellman algorithm

Step 3: Divide k by 10 and store remainder in r

Step 4: Obtain r-Fibonacci series and take pixel positions whose values are multiples of the numbers in the series and embed information in them

Step 5: If data still remains to be hidden, take last pixel modulated as key and repeat steps

Step 6: If remainder obtained using new key is same as r we take r+1 in the series and if r+1 is equal to 10 we convert it to zero

Step 7: Stop

IV. Experimental Analysis

The proposed system was tested against some of the algorithms of recent times on 3 parameters:

1. Time for encryption
2. PSNR Value
3. Time for decryption by a Brute Force attack

Figure 1 shows an image used for testing of the algorithm before and after the execution of the algorithm. The database consisting of the images is the USC-SIPI database [18].

Fig. 1. Before and after execution of algorithm on a test image from USC-SIPI

As can be seen from the output, there is no visually perciptible change and hence the algorithm shows good strength. Table II shows the comparison of algorithms in terms of speed for a file of 50 kB size.

TABLE II
COMPARISON OF SPEED OF EXECUTION FOR 50 KB FILE

S.No	Name of Algorithm	Speed in Seconds
1	Proposed	4.014
2	[5]	4.987
3	[6]	4.879
4	[9]	4.681
5	[10]	4.762

Table III shows a comparison for a file of size 100 kB.

TABLE III
COMPARISON OF SPEED OF EXECUTION FOR 100 KB FILE

S.No	Name of Algorithm	Speed in Seconds
1	Proposed	8.104
2	[5]	9.847
3	[6]	9.179
4	[9]	8.881
5	[10]	9.462

Peak Signal to Noise ratio is a metric commonly associated with steganography algorithms to determine how efficient they are. To compute the PSNR, first the mean-squared error needs to be calculated, it is done using the following equation:

$$MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (2)$$

Next, the PSNR is calucated as follows,

$$PSNR = 10log_{10}(\frac{R^2}{MSE}) \quad (3)$$

Table IV shows a comparison of PSNR values for embedding text of size 50, 100 and 200 kB.

TABLE IV
COMPARISON OF SPEED OF EXECUTION FOR 100 KB FILE

Name of Algorithm	PSNR for 50 kB	PSNR for 100 kB	PSNR for 200 kB
Proposed	74.89	63.15	53.57
[3]	70.45	54.26	40.64
[5]	70.31	52.66	40.16
[6]	70.73	53.84	40.84
[10]	69.66	48.65	34.11

Figure 2 shows a graphical comparison of the algorithms in terms of speed of execution.

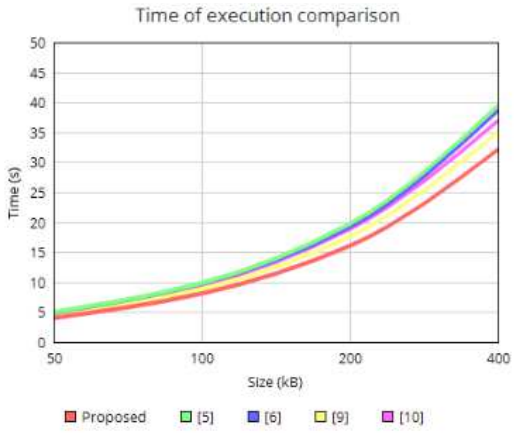

Fig. 2. Graphical comparison of speed of execution for the proposed algorithm with recent state of the art

We also compared the algorithms by using a brute force attack on files of size 1 kB and 5 kB to show the strength of the proposed algorithm.

Table V shows the comparison of decryption time of algorithms for a file of 1 kB size.

TABLE V
COMPARISON OF SPEED OF DECRYPTION FOR 1 KB FILE

S.No	Name of Algorithm	Speed in Seconds
1	Proposed	54.014
2	[5]	49.977
3	[6]	54.189
4	[9]	51.881
5	[10]	56.878

TABLE VI
COMPARISON OF SPEED OF DECRYPTION FOR 5 KB FILE

S.No	Name of Algorithm	Speed in Seconds
1	Proposed	648.581
2	[5]	541.681
3	[6]	651.676
4	[9]	611.615
5	[10]	644.797

Table VI shows the comparison of decryption time of algorithms for a file of 1 kB size.

As can be seen from the results, the algorithm shows a strength of security comparable to most recent state of the art algorithms.

V. CONCLUSION

A novel approach was proposed. Provably secure Diffie-Hellman key exchange was used. The key obtained was divided by 10 and the remainder obtained was then used as the parameter to a p-Fibonacci series. Pixels whose location were multiples of the numbers in the series were taken and the data was embedded in them.

Also, an additional layer of security was provided using the enhanced Caesar Cipher algorithm. From analysis, it was seen that the algorithm was not only much quicker than recent proposed algorithms over even small sizes, the algorithm also gave output having much higher PSNR Value.

Also, the addition of a quick encryption algorithm ensured that security was added to the algorithm without costing the algorithm too much speed constraint. The level of security can be seen in the results from the experimental analysis.

The proposed algorithm gave much better output than recently proposed algorithms and also ensured a high level of security for the algorithm.

REFERENCES

[1] Zeng, J., Tan, S., Li, B. and Huang, J., 2016. Large-scale JPEG steganalysis using hybrid deep-learning framework. arXiv preprint arXiv:1611.03233.

[2] Jiang, N., Zhao, N. and Wang, L., 2016. LSB based quantum image steganography algorithm. International Journal of Theoretical Physics, 55(1), pp.107-123.

[3] Gowda, S.N. and Sulakhe, S., 2016, April. Block Based Least Significant Bit Algorithm For Image Steganography. Annual Int'l Conference on Intelligent Computing, Computer Science & Information Systems (ICCSIS-16).

[4] Paul, G., Davidson, I., Mukherjee, I. and Ravi, S.S., 2016. Keyless dynamic optimal multi-bit image steganography using energetic pixels. Multimedia Tools and Applications, pp.1-27.

[5] Wu, K.C. and Wang, C.M., 2015. Steganography using reversible texture synthesis. IEEE Transactions on Image Processing, 24(1), pp.130-139.

[6] Banik, B.G. and Bandyopadhyay, S.K., 2015, December. Secret sharing using 3 level DWT method of image steganography based on Lorenz chaotic encryption and visual cryptography. In Computational Intelligence and Communication Networks (CICN), 2015 International Conference on (pp. 1147-1152). IEEE.

[7] Muhammad, K., Sajjad, M., Mehmood, I., Rho, S. and Baik, S.W., 2016. A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image. Multimedia Tools and Applications, 75(22), pp.14867-14893.

[8] Abduallah, W.M., Rahma, A.M.S. and Pathan, A.S.K., 2014. Mix column transform based on irreducible polynomial mathematics for color image steganography: A novel approach. Computers & Electrical Engineering, 40(4), pp.1390-1404.

[9] Gowda, S.N., 2016, October. Using Blowfish encryption to enhance security feature of an image. In Information Communication and Management (ICICM), International Conference on (pp. 126-129). IEEE.

[10] Bardhan, O., Bhattacharya, A. and Sinha, B.P., 2014, September. A steganographic technique based on vlsb method using rc4 stream cipher. In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on (pp. 1402-1407). IEEE.

[11] Grover, N. and Mohapatra, A.K., 2013, December. Digital image authentication model based on edge adaptive steganography. In Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on (pp. 238-242). IEEE.

[12] Gupta, S. and Jain, R., 2015, December. An innovative method of Text Steganography. In Image Information Processing (ICIIP), 2015 Third International Conference on (pp. 60-64). IEEE.

[13] Banik, B.G. and Bandyopadhyay, S.K., 2017. Image Steganography Using BitPlane Complexity Segmentation and Hessenberg QR Method. In Proceedings of the First International Conference on Intelligent Computing and Communication (pp. 623-633). Springer Singapore.

[14] Rajendran, S. and Doraipandian, M., 2017. Chaotic Map Based Random Image Steganography Using LSB Technique. International Journal of Network Security, 19(4), pp.593-598.

[15] Wang, Z., Yin, Z. and Zhang, X., 2017. Distortion Function for JPEG Steganography Based on Image Texture and Correlation in DCT Domain. IETE Technical Review, pp.1-8.

[16] Boyko, V., MacKenzie, P. and Patel, S., 2000, May. Provably secure password-authenticated key exchange using Diffie-Hellman. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 156-171). Springer Berlin Heidelberg.

[17] Gowda, S.N., 2016, September. Innovative enhancement of the Caesar cipher algorithm for cryptography. In Advances in Computing, Communication, & Automation (ICACCA)(Fall), International Conference on (pp. 1-4). IEEE.

[18] Weber, A.G., 1997. The USC-SIPI image database version 5. USC-SIPI Report, 315, pp.1-24.