Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 6 (2): 124-134, 2014
ISSN 1815-7432 / DOI: 10.3923/rjit.2014.124.134

A Secure Steganographic Algorithm Based on Fibonacci Representation Using Cellular Automata

Tuan Duc Nguyen, Somjit Arch-int and Ngamnij Arch-int
Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Corresponding Author: Tuan Duc Nguyen, Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand Tel: +66881325034

ABSTRACT

This study proposes a steganographic technique based on Fibonacci representation using cellular automata. The pixel's color component is decomposed into Fibonacci domain to introduce more available bit-planes which can be used in data-hiding. In order to increase the robustness of hidden data against modification attacks, the secret bits are embedded into a higher LSB layer. Nevertheless, the embedding secret bit in the higher LSB layer causes more distortion to cover-image. An adaptive adjustment is applied to minimize the degradation of the cover-image. The security of hidden message is enhanced by employing the Game of Life, a cellular automaton, to generate the used key stream for encryption and image pixel selection. By employing the adaptive adjustment, every pixel in the image can be used in the embedding process. Therefore, the capacity of the embedding secret message is the same as the classical LSB algorithm. Experimental results show that the existence of hidden message cannot be figured out by the visual or statistically attack and the robustness of hidden data in the stego-image is enhanced.

Key words: Secure steganographic, cellular automata, fibonacci representation, adaptive adjustment

INTRODUCTION

Since the rapid development of network techniques, valuable data transferred via the internet has significantly increased. These data can suffer from security breaches (network sniffing and data tampering) if they are without any protection. There have been many steganography techniques which have been developed to protect this information when it is transferred via the internet. These steganographic algorithms can be categorized into spatial and frequency domain techniques. Least Significant Bit (LSB) insertion, which is the simplest hiding technique in the spatial domain, embeds secret message in a subset of the LSB planes of a cover object. Although there are several disadvantages such as vulnerability to attack and ease of detection by statistical attacks (Regular and Singular groups (RS) (Fridrich *et al*., 2001) and Sample Pairs Analysis (SPA) (Dumitrescu *et al*., 2003) or visual attacks (enhanced LSB) (Fridrich and Goljan, 2002), LSB steganography is a popular method due to its low computational complexity and high embedding capacity. LSB algorithm can be used to embed secret messages in many kinds of file format, such as image (Roque, 2009; Ahmed and Ali, 2011; BrahmaTeja *et al*., 2012) and audio (Cvejic and Seppanen, 2005; Bandyopadhyay and Datta, 2011; Liu *et al*., 2012).

The algorithms in (Bandyopadhyay and Datta, 2011) (named as 2SideAdj) embed the secret bit into the higher bit-plane in order to increase the robustness of hidden message in the cover

object. Subsequently, the adjustment is applied on both sides of the k^{th} LSB layer, which has a secret bit embedded in order to reduce the distortion of cover-data. However, the security of the proposed algorithms does not cover statistical attacks due to the number of flipping bits may be larger than the LSB algorithm. The presented algorithm (SLSB) (Roque, 2009) has reduced the statistical change of cover-image by choosing the blue color component to embed secret message. Nevertheless, the change is still high since there are 3 bits of secret message embedded in a blue color component.

There are some steganographic algorithms using Fibonacci representation that have been proposed (Agaian *et al*., 2006; Mammi *et al*., 2009; Patsakis and Fountas, 2010; Aroukatos *et al*., 2012a, b). Through the decomposition, the cover data can be represented using 12 individual bit-layers instead of 8 in binary domain. As a result, there are more secret bits that can be embedded into the cover-data using more bit-planes to embed data. Furthermore, the greater resistance against the RS analysis has been introduced. A novel approach that combines Fibonacci and Catalan numbers was presented (Aroukatos *et al*., 2012a) in order to increase the number of bit-planes that can be used in data-hiding. However, these proposed methods alter the LSBs of the cover-data, which leads to increased possibility of attack by visual method, such as enhanced LSB. Moreover, the robustness of hidden data, which is embedded by algorithms in (Aroukatos *et al*., 2012b), is low because the message bits are still embedded into LSBs.

In this study, a secure adaptive adjustment steganographic algorithm (called AAJFIBO) is presented. At first, the value of the pixel color component is represented by a Fibonacci representative to obtain more available bit-plane which can be used in data hiding. After that, the secret bits are embedded into the high LSB layers to obtain the higher robustness in comparison with the classical LSB algorithm. Therefore, the adaptive adjustment is performed to minimize the distortion of the cover-image. This adaptive adjustment does not modify the LSB of the pixel color component, thus, the statistical properties of the cover-image is not changed as a result. Therefore, the security of hidden data against visual and statistical attack is improved.

Fibonacci decomposition: In a color image, a color component of a pixel can be represented in decimal value in range (0, 255). The value of a color component in the binary system is equivalent to 8 bit representation as follows:

$$D_{10} = b_0 + b_1 2^1 + b_2 2^2 + \ldots = \sum_{i=0}^{7} b_i 2^i$$

where, $b \in \{0, 1\}$.

For this reason, the decomposition of a pixel color component of an image is 8 bit-planes and this presentation does not introduce the redundant. Since the message bit is embedded into the LSB of color component, it is vulnerable even with a small change of the pixel. The message bits need to be embedded into a higher bit-planes with minimal distortion in order to increase the robustness of the LSB embedding scheme. The Fibonacci decomposition is very useful when it introduces 12 bit-planes to represent the values in the range [0, 255] (Agaian *et al*., 2006).

Cellular automata and cryptography: Cellular Automata (CA) are discrete, abstract computational systems that have proved useful both, as general models of complexity and as more

specific representations of nonlinear dynamics in a variety of scientific fields. CA are spatially and temporally discrete: They are composed of a finite or a denumerable set of homogeneous, simple units, the atoms or cells. The cells instantiate one of a finite set of states at each time unit. They evolve in parallel at discrete time steps, following state update functions or dynamical transition rules: The update of a cell state obtains by taking into account the states of the cells in its local neighborhood. A group of cells can be arranged in 1-D, 2-D or 3-D.

CA, first introduced by von Neumann in the early 1950s was used as a prototyping model for a large variety of natural systems, this model have attracted the attention of several research groups.

In the late 1970s, a well know two-dimensional cellular automata was proposed by John Conway, called "Game of Life", which is based on a biological model. The game is played on a grid of squares called cells; each cell is "alive" or "dead". In each iteration, the status of any given cell is determined by a set of four rules:

- Any live cell that is touching fewer than two alive neighbors dies
- Any live cell that is touching four or more alive neighbors dies
- Any live cell that is touching two or three alive neighbors does nothing
- Any dead cell that is touching exactly three alive neighbors becomes alive

After n-generations, the status of the cells in the grid is un-predictable when the CA rules are applied. Thus, CA have previously been used as a pseudo-random sequence generation (Tomassini and Perrenoud, 2001; Abdul Hameed and Eldin, 2007) for cryptographic applications.

The bitwise XOR operator is employed to encrypt the plaintext with a random key due to its simplicity and secure when a non-repeating key stream is used.

Fibonacci based steganography: In general, a pixel's value is represented by 8 bit-planes in binary domain. Therefore, steganographic based on LSB has a limitation in capacity and the low imperceptual quality of stego-images when more bit-planes are used to hide a secret message.

By presenting a pixel's value in Fibonacci domain, more available bit-planes are introduced to embed more secret bits into image pixel with appreciate degradation.

Figure 1 illustrates the principle of Fibonacci based steganographic. At first, the image pixel's value is represented in Fibonacci presentation. A secret bit is then embedded into the Fibonacci sequences by LSB algorithm. Finally, the Fibonacci sequence with secret bit embedded is converted to binary domain before being written to stego-image.

This steganography algorithm is very simple and has a high performance. Nevertheless, not Fibonacci represtation of all pixels in an image can be used for hiding data if embedding secret bit makes the Fibonacci sequences not satisfying the Zeckendorf's theorem. Moreover, the robustness of the secret bit is weak when it is hidden in the LSB.

METHODOLOGY

This section proposed the algorithm that hides secret bits into higher LSB layers of image pixel color component in a Fibonacci domain in order to improve the robustness and security of hidden data embedded into the stego-image. The proposed approach includes 6 steps as follows (Fig. 2):

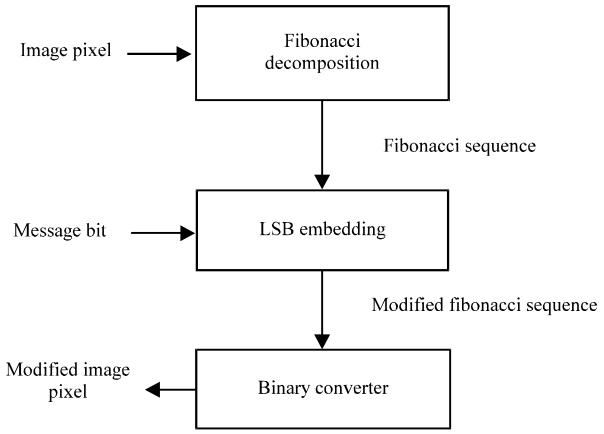

Fig. 1: Fibonacci based steganography

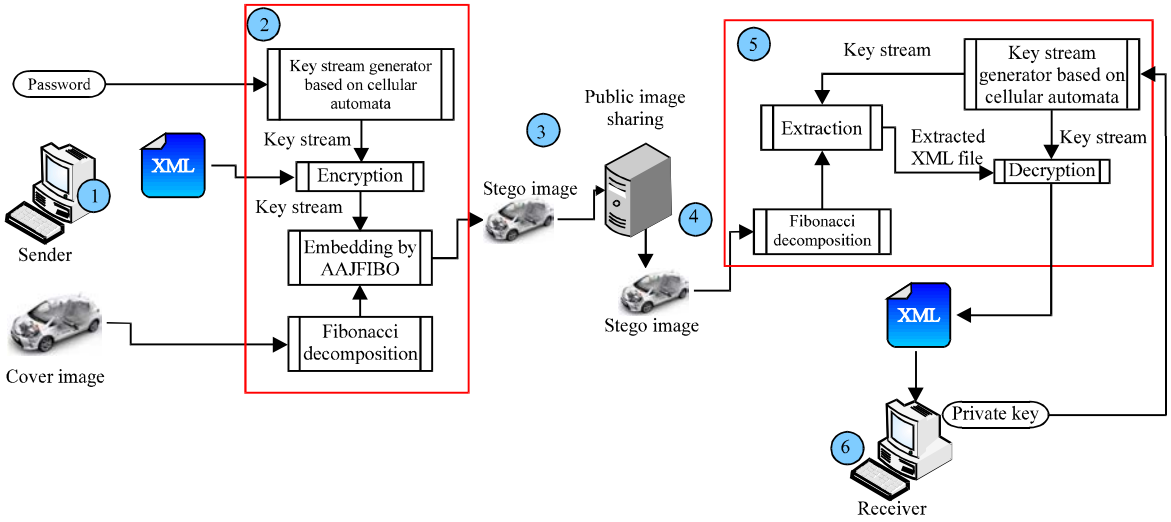

Fig. 2: Conceptual framework of the proposed scheme

- **Step 1:** The password from user is used to produce the key stream
- **Step 2:** The secret message is encrypted by applying the bitwise XOR operator to every bit using a generated key stream. The selected color component (the selection is done using the generated key stream in the first step) is represented into Fibonacci representation. The secret bits are embedded into a k^{th} LSB layer of decomposed image pixel color component. The adaptive adjustment is applied to minimize the distortion of color component of image pixels
- **Step 3:** The stego-image is transferred to public image sharing service where the user B can download it easily without a notice
- **Step 4:** The receiver (user B) downloads the stego-image from the public image sharing service
- **Step 5:** The private key is provided by receiver (user B) to create the initial data for producing the key stream. The generated key stream is then used to identify the color components that carry the secret bits

The blue color component of image pixel is used to embed the secret bit in order to further reduce the distortion of the cover-image. In general, the value of the image pixel is a combination of three color components (Red, Green and Blue), thus, the blue color component modification causes less distortion than green and red (Roque, 2009). Therefore, in this approach, the secret bits are embedded into green or blue color components to minimize the degration and improve the security. The used color components are identified by the value of current bit in key stream. If its value is 0, the green color component is selected to embed secret bit and vice versa.

Encryption based on cellular automata: The secret message is encrypted before the embedding to increase the security performance of the hidden message. The encryption is done by XOR'ing the plaintext with a random key. If the key is non-repeating, as large as or greater than the plaintext, never reuse in whole or part and kept secret, the encrypted data will be impossible to break without knowing the key (Seredynski *et al*., 2004; Protechnix, 2013; Reuvers and Simons, 2013). Hence, the CA rules are applied to the Game of Life (Adamatzky, 2010) to generate the random key stream. Based on our experiment, after limited generations, the number of dead cells in the grid increases very quickly when the four CA rules of Conway are applied. Thus, we employed the Fredkin's CA rule to generate the non-repeating key stream that can pass the Die-Hard battery of tests of randomness.

This generating ability is in accordance with the following steps:

- **Step 1:** A password provided by a user is used to create a random key set. A key's length is longer than 32 and contains uppercase letters, lowercase letters and numbers
- **Step 2:** The generator randomly selects a key from the key set (created in step 1) and uses it to calculate the value of parameter seed. This parameter then is used in Matlab's command rand to generate a two-dimensional array, of size 256×256 Seed Array (SA). The initial status of cells in the grid is identified by comparing the corresponding elements in SA with a threshold 0.5. If the value of a considering element in SA is smaller than 0.5, then the value of the corresponding cell in the grid is set to "0", otherwise, "1" is assigned to this element. The selected key, which is used to initialise the grid, is encrypted by public key cipher before being sent to the receiver in order to re-generate the key stream
- **Step 3:** From the initial state, the status of each cell in the grid is made more complex by applying the Fredkin's rule
- **Step 4:** In each iteration, the differences between the data of current and previous rounds in the grid are measured. If the percentage of the same values is lower than 25, then the data in the grid are written to a key stream. This stage yields the increasing of randomness of the data in the generated key stream. The generation is finished if the generated data are enough to encrypt the given message

The Die-Hard set of statistical tests is performed by the Diehard suite of statistical tests (Marsagliam, 1995), the software developed by George Marsaglia, which measures the repetitiveness of generating the key stream.

An adaptive adjustment LSB algorithm: The message bits are embedded into higher LSB layers in order to increase the robustness of hidden data against any common modification attack (such as Gaussian or Salt-Pepper noise). This process causes significant distortion to the

cover-image pixel. From this fact, the adaptive adjustment is applied after embedding to reduce the distortion. The proposed algorithm, which hides message bit b into the k^{th} LSB layer of a Fibonacci representation of an image pixel color component, includes two steps as follows:

Input: decomposed color component with 12 bit-planes into Fibonacci representation c = (c_{11}, c_{10}... c_0)
Output: decomposed color component into Fibonacci representation with a secret bit embedded
Step 1: Embedding the message bit b into the k^{th} LSB layer of c
if (c_k==0&&b==1) { // case 1
set c_k = b;
set all the bits (c_{k-1}, c_{k-2},..., c_1) to 0
    if (c_{k+1}==1) {
    set c_{k+1}=0;
    set all the bits (c_{k-2}, c_{k-4},... ,c_2) to 1 in a such way that c will contain no consecutive 1's
    }
}
else if (c_k==1&&b==0) { // case 2
    set c_k=b;
    set all the bits (c_{k-1}, c_{k-2},... ,c_1) to 0
    set all the bits (c_{k-1}, c_{k-3},... c_1) to 1 in a such way that c will contain no consecutive 1's
}
Step 2: Adjust if the value of color component larger than 255 after embedding
    if (convfib2dec(c)>255){ // case 3
    find the first bit 1 on the left of k and then set to 0 if it is found
    set all the bits (c_{k-1}, c_{k-2},... ,c_1) to 0
    set all the bits (c_{k-1}, c_{k-3},... c_1) to 1 in a such way that c will contain no consecutive 1's
}

Where, convfib2dec(x) is the user's function that convert Fibonacci representation x to decimal.

With the adaptive adjustment, the message bit b=1 can be hidden into the LSB layer of cover pixel's color component, which a neighbor in the previous bit-plane is a value of 1. As a result, the proposed algorithm overcomes the limitation of the Zeckendorf's theorem.

The image pixel color component is represented into Fibonacci domain. After that, the secret bit is selected from the k^{th} LSB layer of decomposed image pixel color component.

RESULTS AND DISCUSSION

In the experiment, the selected cover-images were taken from the USC-SIPI Image Database (Weber, 1997), [12] and an xml file format with various lengths were used as the secret message. The values of PSNR were calculated in order to measure the perceptual transparency of the stego-image. After that, the security performance of secret message was estimated by performing RS analysis, SPA (Sample Pairs Analysis) and an enhanced LSB attack. The last experiment measured the robustness of hidden data into stego-images by calculating the Bit Error Rate (BER). The robustness experiment was done by adding Gaussian and Salt-peper noises to the stego-image and then the embedded message was extracted. The comparison of extracting message with the original message was performed in order to calculate the BER.

Perceptual transparency analysis: As shown in Fig. 3, although the secret bits were embedded into the 5^{th} (AAJFIBO5) and 7^{th} MSB layer (AAJFIBO7) of the pixel color component of

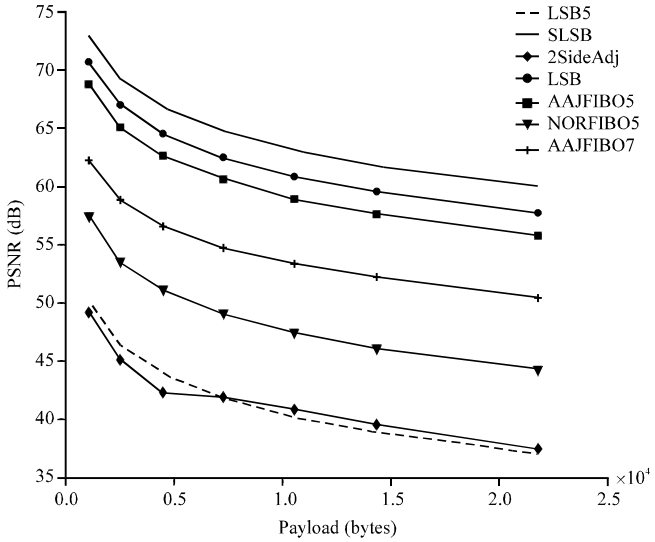

Fig. 3: PSNR comparison of stego-images

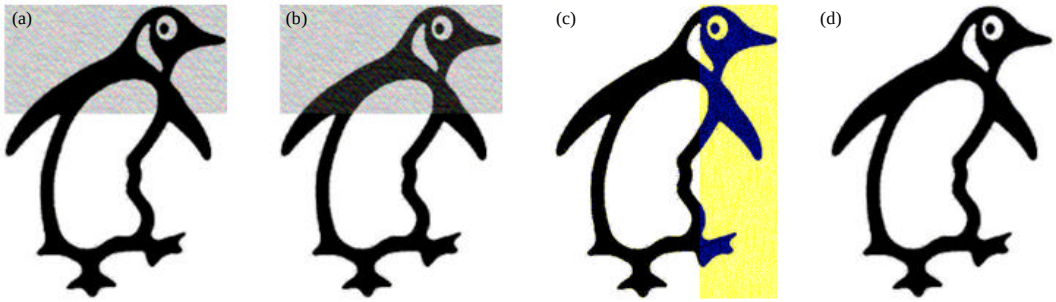

Fig. 4(a-d): Result images from the enhanced LSB attack

cover-images, the values of PSNR of stego-images created by AAJFIBO still higher than the other stego-images from 2SideAdj, normal Fibonacci algorithm (Alharbi, 2013) and classical LSB algorithm with secret bits embedded into the 5th MSB layer. The values of PSNR of stego-images with secret embedded into the 5th MSB layer of the pixel color component are close to the values of stego-images measured from the classical LSB algorithm (with secret bits embedded into LSB layer).

Security analysis: In this experiment, the enhanced LSB attack was done by xsteg secret (Munoz, 2007) and the generated images are shown in Fig. 4.

The change of LSB cannot be distinguished by the human eye because the binary value (0, 1) of LSB on a 256 values range does not produce any visible color. Therefore, the enhanced LSB attack enhances these LSBs by modifying a 1 to 255 but keep a 0 as original. The LSBs of the stego-image is visible.

In general, the value of LSB of the pixel color component is not random in an image without message embedded (there are some uniform color areas in an image). Since the

message bits are embedded into cover-image by classical LSB algorithm, the LSB of pixels in uniform color areas create a pattern which can see easily by the enhanced LSB attack.

Figure 4 illustrates the images produced by the 2SideAdj algorithm (a), the classical LSB algorithm (b) and SLSB (c), by which the produced images have a pattern after the LSBs of these images were enhanced. From the pattern of the generated images, it's possible to estimate the length of hidden message. In contrast with the generated images, the image generated by the proposed algorithm (d) has no pattern after LSB enhance process because the embedding process does not modify the LSBs of the cover-image.

Nevertheless, the enhanced LSB attack method is effective only when it is applied on an image that contains the uniform color (white or black).

From this fact, the RS analysis and SPA are performed in the image set that contains a stego-image with a secret message (21779 bytes) embedded by the 5 steganographic algorithms. With 2SideAdj, proposed algorithm (AAJFIBO) and normal Fibonacci algorithm (Alharbi, 2013), the secret message bits are embedded in the 5th MSB layer. The results of RS analysis and SPA are listed in Table 1.

As shown in Table 1, the estimated length of hidden message in the stego-image created by the classical LSB algorithm and SLSB are close to the length of the embedded message, whereas these values of the stego-image produced by 2SideAdj and AAJFIBO are significantly higher than the length of secret message. However, the length of hidden message was estimated by measuring the proportion of regular and singular groups of spatially adjacent pixels in RS analysis and the change of characterize of sample pairs in SPA. Thus, the higher value of estimated length, the higher possibility of the existence of hidden message. The estimated length of hidden messages embedded by AAJFIBO indicates that the possibility of the existence of hidden message detected by the RS analysis and SPA compared with previous algorithm is decreased.

Furthermore, the security of hidden data against extracting attack is enhanced using the generated key stream to select the image pixel used in data hiding. Thus, if the hacker figures out the existence of the secret message in the stego-image, the secret message still cannot be extracted.

Robustness analysis: The Gaussian noise and Salt-Pepper noise were added to the stego-image in order to measure the robustness of hidden data embedded into the stego-image. The mean of Gaussian noise is zero and variation varies between 0.1 and 0.35, whereas the density of Salt-Pepper noise varied between 0.0001 and 0.0006.

After that, 21779 bytes of hidden message were extracted and compared with an original XML file (21779 bytes of length) to calculate the bit error rate.

As shown in Fig. 5a, although the secret bits were embedded into 5th (AAJFIBO5) of cover-image, the BERs of AAJFIBO are lower than 2SideAdj. With embedding of secret bits into a higher MSB layer (such as 7th MSB-AAJFIBO7), the robustness of hidden data against Gaussian noise addition attack is further increased. The BERs shown in Fig. 5a, however, are higher compared with Fig. 5b because Gaussian noise is random and affects a large number of image pixels; mean while Salt-Pepper only affects a small number of image pixels but with high

Table 1: Estimated length of secret messsage embedded into stego-images

	Classical LSB	2SideAdj	AAJFIBO	Normal FIBO	SLSB
RS	21193	72999	2045	1205	23409
SPA	12664	83706	5113	12394	25204

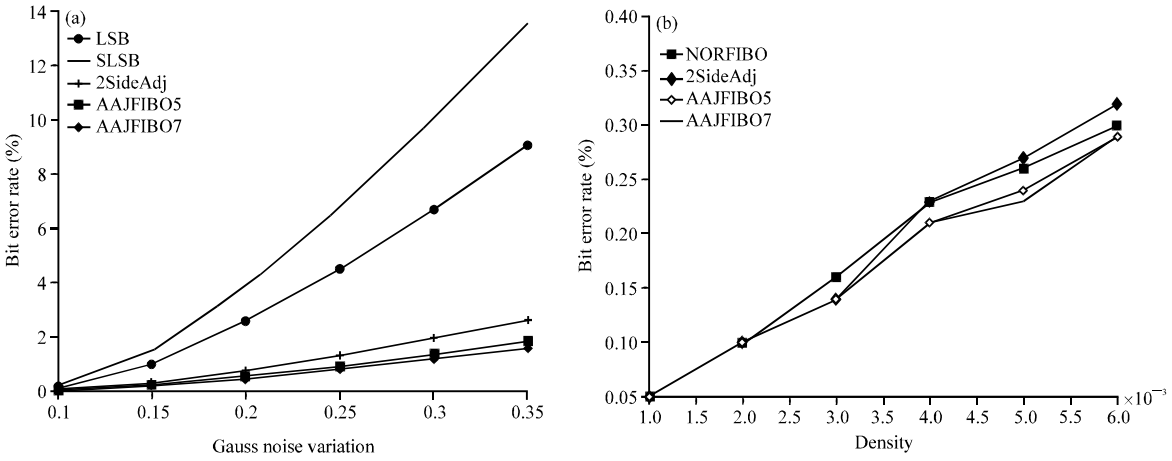

Fig. 5(a-b): BERs estimated from the stego-images after added (a) Gaussian noise and (b) Salt-Pepper noise

Table 2: BERs of stego-images created by the classical LSB algorithm

Density	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006
BER (%)	0.04	49.82	49.83	49.85	49.81	49.82

intensity. This means that this type of noise can affect a large number of bit-planes of image pixels. As a result, the robustness of hidden data against Salt-Pepper noise addition attack is enhanced slightly compared with 2SideAdj. The BERs of the stego-image with the message length 21779 bytes embedded by the classical LSB algorithm was shown in Table 2. Because the values of BER are very high, hence, these BERs cannot be illustrated in the same graph with BERs from the 2SideAdj and proposed algorithm.

Table 2 illustrates the significant effect of Salt-Pepper noise on the stego-image, which contains the secret message embedded by classical LSB algorithm. With the value of density from 0.0002-0.0006, the values of BER are nearly 50% of the total number of the embedded secret bits.

CONCLUSION

This study proposes a secure adaptive steganographic algorithm based on cellular automata and Fibonacci bit-plane decomposition. Fibonacci representation introduces more available bit-planes which can be used in data-hiding. The secret bits can be embedded into the higher LSB layers with low distortion on the cover-image. The adaptive adjustment is operated in order to further minimize the distortion of the cover-image. The results show that the higher LSB layers can be used to embed secret bits and the robustness against modification attacks (Gaussian and Salt-Pepper noise addition) is improved. Furthermore, the proposed algorithm enhances the performance of security of hidden data against visual and statistically attacks.

ACKNOWLEDGMENT

We wish to acknowledge the support of the Khon Kaen University Publication Clinic, Research and Technology Transfer Affairs, Khon Kaen University, for their assistance.

REFERENCES

Abdul Hameed, M.U. and A.M.B. Eldin, 2007. A cellular automata random number generator for cryptographic applications. Proceedings of the International Conference on Computer Engineering and Systems, November 27-29, 2007, Cairo, pp: 101-105.

Adamatzky, A., 2010. Game of Life Cellular Automata. Springer, New York, USA., ISBN: 9781849962162, Pages: 621.

Agaian, S.S., R.C. Cherukuri and R. Sifuentes, 2006. A new secure adaptive steganographic algorithm using fibonacci numbers. Proceedings of the IEEE Region 5 Conference, April 7-9, 2006, San Antonio, TX., USA., pp: 125-129.

Ahmed, J.M. and Z.M. Ali, 2011. Information hiding using lsb technique. Int. J. Comput. Sci. Network Sec., 11: 18-25.

Alharbi, F., 2013. Secure steganography system for rgb images. Int. J. Eng. Adv. Technol., 2: 492-494.

Aroukatos, N., K. Manes, S. Zimeras and F. Georgiakodis, 2012a. Data hiding techniques in steganography using fibonacci and catalan numbers. Proceedings of the 9th International Conference on Information Technology: New Generations, April 16-18, 2012, Las Vegas, NV., pp: 392-396.

Aroukatos, N., K. Manes, S. Zimeras and F. Georgiakodis, 2012b. Data hiding techniques in steganography using sub-fibonacci sequences. Proceedings of the 8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, July 18-20, 2012, Piraeus, pp: 89-93.

Bandyopadhyay, S.K. and B. Datta, 2011. Higher lsb layer based audio steganography technique. Int. J. Electr. Commun. Technol., 2: 129-135.

BrahmaTeja, K.N., D.G.L. Madhumati and K.R.K. Rao, 2012. Data hiding using edge based steganography. Int. J. Emerging Technol. Adv. Eng., 2: 285-290.

Cvejic, N. and T. Seppanen, 2005. Increasing Robustness of LSB audio steganography by reduced distortion LSB coding. J. Univ. Comput. Sci., 11: 56-65.

Dumitrescu, S., X. Wu and Z. Wang, 2003. Detection of LSB steganography via sample pair analysis. IEEE Trans. Signal Process., 51: 1995-2007.

Fridrich, J. and M. Goljan, 2002. Practical steganalysis of digital images-state of the art. Proceedings of the SPIE Photonic West Electronic Imaging 2000 (Security and Watermarking of Multimedia Contents IV), Volume 4675, January 21-24, 2002, San Jose, CA., USA.

Fridrich, J., M. Goljan and R. Du, 2001. Detecting LSB steganography in color and gray-scale images. IEEE Multimedia, 8: 22-28.

Liu, J., K. Zhou and H. Tian, 2012. Efficient least-significant-bits steganography for voip. Int. J. Advancements Comput. Technol., 4: 297-305.

Mammi, E., F. Battisti, M. Carli, A. Neri and K.O. Egiazarian, 2009. Substitutive steganography in the generalized fibonacci domain. Proceedings of the 7th Image Processing: Algorithms and Systems, Volume 7245, January 2009, San Jose, California, USA..

Marsagliam, G., 1995. Diehard battery of tests of randomness. The Diehard Test Suite. http://stat.fsu.edu/~geo/diehard.html.

Munoz, A., 2007. StegSecret. A simple steganalysis tool. http://stegsecret.sourceforge.net/.

Patsakis, C. and E. Fountas, 2010. Extending Fibonacci LSB data hiding technique to more integer bases. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, Volume 4, August 20-22, 2010, Chengdu, pp: V4-18-V4-21.

Protechnix, 2013. Cryptology and data secrecy: The vernam cipher. http://www.pro-technix.com/information/crypto/pages/vernam_base.html

Reuvers, P. and M. Simons, 2013. One-Time Pad (OTP). Sara Toga, July 9, 2013. http://www.scribd.com/doc/152573445/One-Time-Pad-OTP.

Roque, J.J., 2009. SLSB: Improving the *Steganographic algorithm* LSB. Proceedings of the 7th International Workshop on Security in Information Systems, May 6-7, 2009, Milan, Italy, pp: 57-66.

Seredynski, F., P. Bouvry and A.Y. Zomaya, 2004. Cellular automata computations and secret key cryptography. Parallel Comput., 30: 753-766.

Tomassini, M. and M. Perrenoud, 2001. Cryptography with cellular automata. Applied Soft Comput., 1: 151-160.

Weber, A.G., 1997. The USC-SIPI image database version 5. Signal and Image Processing Institute, USC-SIPI Report No. 315, October 1997.