A Secure Steganography Scheme Using LFSR

Debalina Ghosh, Arup Kumar Chattopadhyay, Koustav Chanda and Amitava Nag

Abstract Steganography is a technique to hide the secret data inside some other cover files (like text, image, audio, etc.) in such a way that it prevents the detection of the secret data within the cover files. In this paper, a new scheme of concealing secret data has been introduced where the locations of hiding the secret bits will be generated by linear feedback shift register (LFSR). Using LFSR, we will compute random numbers (within a given range) which will determine the specific pixels where secret bits of the secret data will be concealed. We use least significant bit (LSB) technique where the secret bit will be stored at the least significant bit position of the pixel. Choosing the cover pixels from cover image randomly will strengthen the security of LSB technique.

Keywords Steganography · LSB · LFSR · Grayscale image · Secret text

1 Introduction

The term steganography [1] came from Greek words "*stegos*" meaning "cover" and "*graphia*" meaning "writing" that is covered writing. It is a technique to hide information in a cover medium like text, image, audio or video file. There is always a comparison between cryptography and steganography. In cryptography, the secret data will be encoded in some unintelligible form (meaningless) with some specific key. An attacker without having the key does not gain any knowledge about the secret data. But the entire secrecy of the information depends on the key. Once the key is exposed, encrypted data are not secured any more. But in steganography, it hides

D. Ghosh (✉) · A. K. Chattopadhyay
Institute of Engineering & Management, Kolkata, India
e-mail: debalinag1986@gmail.com

K. Chanda
Academy of Technology, Kolkata, India

A. Nag
Central Institute of Technology, Kokrajhar, India

© Springer Nature Singapore Pte Ltd. 2020

J. K. Mandal and D. Bhattacharya (eds.), *Emerging Technology in Modelling and Graphics*, Advances in Intelligent Systems and Computing 937,
https://doi.org/10.1007/978-981-13-7403-6_62

the existence of the secret data in the cover media. The secret is concealed in such a manner that the existence of the secret data remains unknown to the observer or attacker. In this paper, we will use LSB steganographic technique. Among steganography methods, LSB [2] is a very well-known technique. In LSB, the least significant bits of the cover image pixels are used to hide the secret data. The traditional LSB technique is simple, but not very secure. In some modified LSB schemes like [3], a few bits from the most significant side determine the position at least significant side where to hide the secret bit.

To increase the security of LSB steganography, we use the concept of LFSR (a random number generator) [4] in steganography. Linear feedback shift register (LFSR) is a shift register where the input bit is a linear function of the previous state. The initial value of the LFSR is called seed. The operation of the shift register is deterministic. If the current state is known, then the next sequence of values can be determined. LFSR having a well-chosen feedback function can generate a large sequence of random bits. The bit stream generated by LFSR is pseudo-random and also satisfies the cryptographic randomness criteria. Two main parts of LFSR are the shift register and the feedback function. The task of a shift register is to shift the contents of the register to their adjacent places in one direction, such that one position at the other end becomes empty. That position remains empty unless a new content will be entered into the register. The new content will be generated by a linear function. The inputs are the contents of the filled positions. There is an exception in LFSR—if all the contents of the shift register are zeros, then the next state cannot be generated.

In traditional LSB technique, consecutive bytes are used to store the secret. In our scheme, we will use LFSR-based random number generator [5], and these random numbers will determine the pixels within the cover image, where the secret bits will be hidden. As a result, the sequence of the pixels that hide the secret is a random sequence.

The rest of the paper is divided into following major parts: in Sect. 2, we have discussed a few previous works done in the domains of steganography and LFSR; Sect. 3 comprises the proposed algorithm, and in Sect. 4 we conclude the paper.

2 Related Study

Steganography has been used in variety of domains like transform domain, spread domain, image domain, etc. In image steganography, the secret message will be hidden in some cover image or images. The cover image selection is also very important for a few algorithms, because at the time of fetching secret data, the recovery of the secret should be lossless. There are different types of methods available in digital image steganography to hide secret data within the cover images. One of them is least significant bit (LSB) technique. Kumar et al. [6] discussed about file format in steganography commonly used for cover medium like text, image, audio, video, protocol, etc. They have also performed a comparative study of steganography al-

gorithms where image is the cover object. Most significant among these techniques are least significant bit (LSB) and discrete cosine transform (DCT). The detailed analysis of these two techniques is performed in [7].

2.1 Least Significant Bit (LSB) Method

In this method, the secret data are concealed at least significant bit position of the consecutive pixels of the cover image. Following is the algorithm [8] to embed secret text message using grayscale image as cover.

Method of Embedding of Secret
Step 1: Read the secret message and the cover image.
Step 2: Convert the secret message to binary sequence.
Step 3: Find out the LSB of each pixels in the cover image.
Step 4: Replace the least significant bit of each cover pixel in the cover image with a secret bit (from the secret message) one by one.
Step 5: Generate the stego-image.

Method of Extraction of Secret
Step 1: Read the stego-image.
Step 2: Calculate the least significant bits from each pixel of the stego-image.
Step 3: Retrieve the LSBs and convert each eight bits into a character. Arrange the characters in sequence to reconstruct the secret message.

2.2 A Brief Review of LFSR

An LFSR can be considered as a finite automaton. A finite automaton contains finite number of states. Instead of states, LFSR consists of a number of stages. We are considering a LFSR of length L over $\mathbb{F}_q$. LFSR having length L contains L stages, where stage $S = S_t,\ t \geq 0$ and generates semi-infinite sequence of elements of $\mathbb{F}_q$. It must satisfy the linear recurrence relation of degree L over $\mathbb{F}_q$:

$$s_{t+L} = \sum_{i=1}^{L} c_i s_{t+L-i} \quad \forall t \geq 0.$$

where $c_1, c_2, \ldots, c_L$ are the feedback coefficients of LFSR. The LFSR of length L consists of L delay cells called stages. The contents of L stages are $s_t, s_{t+1}, \ldots, s_{t+L-1}$ which form one state of LFSR. Let the initial content be $s_0, s_1, \ldots, s_{L-1}$.

An external clock is used to control the LFSR. In each clock pulse, the bits will be shifted toward right and the content of rightmost stage S_t is the output. Now the new content at the leftmost stage is the feedback bit s_{t+L}, and it is computed as:

$$s_{t+L} = \sum_{i=1}^{L} c_i s_{t+L-i}.$$

The main weakness of traditional LSB steganography is that the secret bits are stored in consecutive pixels of the stego-image. For strengthening the security of LSB method, randomness in selection of cover pixels is important. LFSR is a random number generator and popularly used in many modified steganography schemes [9–11] and their applications. Random numbers are of two types, one is "truly random" and the other one is "pseudo-random." The random numbers generated using some mathematical algorithms are of the pseudo-random type. Digital random number generators are used in different cryptography applications. The authors in [12–14] have proposed different types of 8-bit and 16-bit LFSR-based random number generators for high-secured multipurpose operations.

In [12], a polynomial modulator has been proposed to avoid the predictability in random numbers. Because periodic and predictable random sequences will help the intruder to find the secret. The authors in [9] implemented steganography with random number generator. They have introduced an algorithm where we found four stages. First stage generates pseudo-random sequences (pseudo-random sequences are generated by linear feedback shift register and standard chaotic map), permutation and XORing using pseudo-random sequences conducted in second stage. Encryption using Rabin cryptosystem and steganography using the improved diagonal queues are the last two stages. In [10], the authors proposed the use of random bit sequences for the purpose of steganography. The random bit sequence has been generated by linear feedback shift registers (LFSRs), and a new factor named beta factor has been created. The task of beta factor is to select cover from the database of covers.

3 Proposed Method

Let the cover image be I_c (of size $w \times h$) and the secret text be T_s which is in binary form of any given length l, $T_s \in \{0, 1\}^l$. The steps in the construction of stego-image are as follows.

3.1 Embedding of Secret Bits

3.1.1 Initialization

Use padding technique (add enough rows and columns of zeros) such that cover-image matrix I_c can be divided into blocks of $n \times n$ matrices (n must be chosen such that $n = 2^k$ *where* $k \geq 2$ and $n \leq min\{\frac{w}{2}, \frac{h}{2}\}$). The number of such matrices will be as follows:
Number of matrices row-wise, $row\_count = \lceil \frac{w}{n} \rceil$
Number of matrices column-wise, $column\_count = \lceil \frac{h}{n} \rceil$
Total number of matrices, $N = row\_count \times column\_count$. Let the matrices are represented as $M_{(i,j)}$ *where* $1 \leq i \leq row\_count$ *and* $1 \leq j \leq column\_count$.

3.1.2 Selection of Blocks for Bit Insertion

We consider $M_{(i,j)}$ as cover-matrix, and the secret bits will be inserted at the least significant position of each pixel belongs that block.
The block selection method is as follows:

for $i = 1$ *to row_count* $- 1$ **do**
  for $j = 1$ *to column_count* $- 1$ **do**
    select the matrix $M_{(i,j)}$ *as cover matrix*
  end
end

We intentionally do not use any of the matrices from row_count^{th} row and $column\_count^{th}$ column as due to padding those may be filled with zeros.

3.1.3 Selection of Different Seeds for LFSR at Different Blocks

To ensure randomness to choose the cover pixels in different blocks, we compute the seed for each matrix $M_{(i,j)}$ as follows.
Initialize the first seed of r bits (r depends on the polynomial used for LFSR) from MSB part of the first pixel ($P_{[(1,1),(1,1)]}$) of first matrix ($M_{(1,1)}$). The seed for rest of blocks or matrices will be generated as follows.

$seed_{prev} = 0$
for *for* $i = 1$ *to row_count* $- 1$ **do**
  for *for* $j = 1$ *to column_count* $- 1$ **do**
    select the matrix $M_{(i,j)}$ *as cover matrix*
    select the r bits from the MSB of pixel $P_{[(i,j),(1,1)]}$ *as* $seed_{new}$
    $seed = seed_{prev} \oplus seed_{new}$
    $seed_{prev} = seed$
  end
end

3.1.4 Insertion of Length of Secret Bit Stream for the Secret Text

The length of the bit stream for the secret text which is l will be inserted into the first block, i.e., $M_{(1,1)}$. The l is converted into binary form, and the bits will be inserted into the least significant position of the cover-pixels selected by the order generated by the LFSR algorithm. The LFSR generates $2 \times k$ bits at a time, where first k bits decide the x-coordinate (p) and second k bits decide the y-coordinate (q); the cover pixel hence chosen is $P_{(p,q)}$ of matrix $M_{(i,j)}$.

3.1.5 Selection of Cover Pixel from $M_{(i,j)}$ for Bit Insertion

Consider a flag matrix $FM_{(i,j)}$ corresponding to each $M_{(i,j)}$, which stores the value one or zero. We initialize all flag matrices with value zero. From LFSR, we generate $2 \times k$ random bits. The first k bits will determine the row number (p), and next k bits will determine the column number (q) within the matrix and presented as pixel $P_{[(p,q),(i,j)]}$. The secret bit from T_s will be inserted at LSB of pixel $P_{[(p,q),(i,j)]}$ (pixel $P_{(p,q)}$ of matrix $M_{(i,j)}$). Each time a secret bit is inserted at the least significant position of the pixel at matrix position (p, q), the value at corresponding position at flag matrix will be changed to one. The flag matrix keeps the trace of which pixels are already used to store secret bits.

If the same pixel selected more than once as a cover pixel (a collision occurs), we utilize *linear probing technique*. In *linear probing technique*, if the cover pixel is already utilized, we scan the matrix for next unused pixel to insert the secret bit. The process will be continued until either we utilized all the pixels of the same matrix or we are exhausted with secret bits. In case all the pixels of the same matrix are already used, the process will be repeated for the next matrix.

3.1.6 Construction of Stego-Image

Once we are exhausted with all the secret bits (all the secret bits are already inserted into cover image I_c), we generate the stego-image I_{stego}.

3.2 Extraction of Secret Bits

The stego-image I_{stego} is considered as input.

3.2.1 Initialization

Generate the block matrices $M_{(i,j)}$ of size $n \times n$ (where $1 \leq i \leq row\_count$ and $1 \leq i \leq column\_count$). The *row_count* and *column_count* computation is as discussed in the Sect. 3.1.1.

3.2.2 Selection of Block for Bit Extraction

Select the block matrices $M_{(i,j)}$ sequentially as discussed in Sect. 3.1.2.

3.2.3 Selection of Different Seed for LFSR at Different Blocks

The selection of the seed for LFSR at different blocks as discussed in Sect. 3.1.3.

3.2.4 Extraction of the Length of Secret Bit Stream for the Secret Text

The length of the secret bit stream is concealed in the first block $M_{(1,1)}$. Using LFSR, we generate the random bit sequence. We combine each $2 \times k$ bits to find the coordinate $P_{(p,q)}$ (where p and q are calculated from first and second k bits) where the secret bit is hidden. Extract the secret bit from the least significant position of the selected pixel. Hence, combine all the secret bits to construct the length (l) of the secret text (in binary form).

3.2.5 Identification of Cover-Pixel from $M_{(i,j)}$ and Bit Extraction

For each block matrix $M_{(i,j)}$, the cover-pixels are identified by the LFSR algorithm. The LFSR is used to generate $2 \times k$ bits, where first k bits construct the p, the x-coordinate and the second k bits construct the q, the y-coordinate. The (p, q) gives us the pixel position. The secret bit is at the least significant bit of pixel $P_{(p,q)}$, which is extracted. If all the l secret bits are extracted, then those can be combined to generate the secret text T_s.

4 Conclusion

In the proposed scheme, we have considered a cover image to conceal a secret text message. The cover image is first segmented into equal size blocks. Then, those blocks are selected one by one to hide secret bits of the text message. Within the block, each pixel stores one bit of secret message. Each secret bit is inserted at the least significant bit position of one of the pixels. The sequence of the cover-pixels within each block is generated by LFSR-based random number generator. It also ensures that the seeds selected for the LFSR are different for different blocks. The use of pseudo-random sequence of the cover pixels to store the secret bits makes the scheme more secure.

References

1. F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey. Proc. IEEE. **87**(7), 1062–1078 (1999)
2. N. Kaur, S. Behal, A survey on various types of steganography and analysis of hiding techniques. Int. J. Eng. Trends Technol. **11**(8), 388–392 (2014)
3. P. Pathak, A.K. Chattopadhyay, A. Nag, A new audio steganography scheme based on location selection with enhanced security, in *First International Conference on Automation, Control, Energy and Systems (ACES)* (IEEE, Hooghy, India, 2014), pp. 1–4
4. W. Mao, Y. Li, C.H. Heng, Y. Lian, Zero-bias true random number generator using LFSR-based scrambler, in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* IEEE, Baltimore, MD, USA (2017), pp. 1–4
5. A. Kaur, H.K. Verma, R.K. Singh, 3D playfair cipher using LFSR based unique random number generator, in *2013 Sixth International Conference on Contemporary Computing (IC3)*. IEEE, Noida, India (2013), pp. 18–23
6. R. Kumar, K. Choudhary, N. Dubey, An introduction of image steganographic techniques and comparison. Int. J. Electron. Comput. Sci. Eng. **1**(3), 1000–1005 (2012)
7. E. Walia, P. Jain, N. Navdeep, An analysis of LSB & DCT based steganography. Global J. Comput. Sci. Technol. **10**(1), 4–8 (2010)
8. R. Garg, T. Gulati, Comparison of Lsb & Msb based steganography in gray-scale images. Int. J. Eng. Res. Technol. **1**(8) (2012)
9. M. Jain, A. Kumar, R.C. Choudhary, Improved diagonal queue medical image steganography using Chaos theory, LFSR, and Rabin cryptosystem. Brain Inform. **4**(2), 95–106 (2017)
10. I.A. Sattar, M.T. Gaata, Image steganography technique based on adaptive random key generator with suitable cover selection, in *2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT)*. IEEE, Baghdad, Iraq (2017), pp. 208–212
11. T. Jamil, A. Ahmad, An investigation into the application of linear feedback shift registers for steganography, in IEEE SoutheastCon 2002. IEEE, Columbia, SC, USA (2002), pp. 239–244
12. P. L'Ecuyer, F. Panneton, A new class of linear feedback shift register generators, in *2000 Winter Simulation Conference Proceedings*. IEEE, Orlando, FL, USA (2000), pp. 690–696
13. M. Sahithi, B.M. Krishna, M. Jyothi, K. Purnima, A.J. Rani, N.N. Sudha, Implementation of random number generator using LFSR for high secured multi purpose applications. Int. J. Comput. Sci. Inf. Technol. **3**(1), 3287–3290 (2012)
14. M. Han, Y. Kim, Unpredictable 16 bits LFSR-based true random number generator, in *2017 International SoC Design Conference (ISOCC)*. IEEE, Seoul, South Korea (2017), pp. 284–285