SIMPLE ENCRYPTION ENCODING FOR DATA HIDING AND SECURITY APPLICATIONS

MD. ANWAR HUSSAIN[1]

Abstract. Effective and light weight, in terms of computational requirements robust encryption algorithms are preferable for data hiding and data security applications. Steganography data hiding, watermarking for copy right protection, secure transmission of biometric data to local or remote sites for validation and recognition, and secure data storage in a digital system are such few applications. We report in this paper an encryption algorithm which is simple and light weight in implementation but robust in its capability to hide or secure the message data. The algorithm requires small block and code based substitution of binary message data before embedding, storage or transmission. We report our work on steganography and watermarking using our proposed encryption algorithm and then simple LSB encoding, and data storage and data transmission applications with our encryption algorithm on the data. We observe higher performance compared to almost all LSB encoding techniques. The encryption algorithm thus provides both data security and data hiding capability and may be applied on all types of binary data processing as an optional and additional step.

Keywords: LSB encoding, watermarking, Biometrics, Security

1.1 Introduction

Almost every information today, be it text, audio, music, image, video or multimedia are dealt with digital techniques for storage, transmission, or protection. Modern communication systems or storage systems are heavily dependent on computer systems and internet and hence are prone to attack. Also as most of the user information is in digital domain, digital signal processing for storage, compression, hiding and security are most prevalent now a day. The message information often needs encryption like processing techniques before they are put to sto-

[1] MD. A. Hussain (✉)
North Eastern Regional Institute of Science & Technology, India
e-mail: bubuli_99@yahoo.com

rage, transmission, covertly communicated or used for copy right protection. Steganography technique hides [1, 2] secret message in a cover audio, image or video file without attracting attention of others and thus covertly communicated. Similarly the technique is used to protect a copy right of a digital product by digital watermarking a logo and thus prohibiting illegal use. Simple LSB encoding technique [3, 5] and its variants [8] and also block based DCT, SVD and DWT [4, 10] based data hiding techniques are mostly used for steganography and watermarking applications. In certain other applications like sending low volume biometric data [12] to local or distant sites for scrutiny, validation and recognition needs to be secured before transmission, from attack. Likewise data stored in computer systems are to be secured before storage. As most of today's data generated by users or an application all over the world are digital multimedia and mostly exchanged between users or user sites or stored in digital systems, data processing techniques should have inbuilt protection mechanisms.

In all the above said applications, as the concern is the security of the user data, encryption technique seems to be the first choice. Conventionally there are a number of encryption techniques like DES, TDES, AES, IDEA, Blowfish, RC5, CAST-128 and many others for encrypting user data. Now all of them are key based whose strength depends on the key size and also the executions of such techniques are computationally intensive. Moreover they convert plaintext to cipher text, thus attracting the attention of attackers with more seriousness [6, 7]. The conventional encryption techniques require message block sizes of 64 bits or higher and requires intensive mathematical operations which make it less vulnerable to crypto analysis.

In this paper, we report a simple data encryption technique which is based on small blocks of 4 bits binary data, and substituting it by another block of same size using different codes. As 4 bits data makes 16 different combinations, we use a set of 16 different codes to choose from to substitute a given 4 bit block by another 4 bit block, the code used for a specific 4 bit block is not used again for any other 4 bit block. Thus the user binary data segmented into consecutive 4 bit blocks are substituted by the respective 4 bit codes. The decryption algorithm is also simple and follows the reverse of the encryption technique. The simple encrypted data are then used for its particular applications of steganography, watermarking, storage, transmission or compression.

1.2 Methodology

We assume here that the plaintext user data is in binary form and arranged as row or column or rectangular array and an integer multiple of four. The plaintext user data may originate from simple text, audio, music, image, video or multimedia ap-

plications. The image or video data which may be colored or gray scale in any format is assumed here to be in gray scale for simple explanation of our algorithm. The user plaintext data may be required to hide in a cover data which may be audio, music, image or video and then transmitted as covert communication, or be marked to protect it from illegal use, or low volume biometric data of a specific application and being transmitted to a distant or local site for scrutiny, validation or recognition, or data stored in a computer system to be used in certain applications.

The plaintext binary data is segmented into 4 bit blocks which may be represented as $B_i = (b_0, b_1, b_2, b_3)$ as row, or $B_i = (b_0, b_1, b_2, b_3)'$ as a column, or $B_i = (b_0, b_1; b_2, b_3)$ as a rectangular block, where the index i refers to its decimal equivalent in the set {0, 1, 2, 3, ….., 15} referring to the particular 4 bit combination out of 16 possible combinations. The 4 bit substitution block is indexed as codes in the code set C = {C0, C1, C2, ….. , C15} where a specific 4 bit code is used to substitute a specific 4 bit user data only, any code may be used for any 4 bit user data but not to be assigned again to any other 4 bit user data. As an example, we take the following 16 possible combinations of 4 bit block of user data.
$B_0 = (0\ 0\ 0\ 0)$, $B_1 = (0\ 0\ 0\ 1)$, $B_2 = (0\ 0\ 1\ 0)$, $B_3 = (0\ 0\ 1\ 1)$, $B_4 = (0\ 1\ 0\ 0)$, $B_5 = (0\ 1\ 0\ 1)$, $B_6 = (0\ 1\ 1\ 0)$, $B_7 = (0\ 1\ 1\ 1)$, $B_8 = (10\ 0\ 0)$, $B_9 = (1\ 0\ 0\ 1)$, $B_{10} = (1\ 0\ 1\ 0)$, $B_{11} = (1\ 0\ 1\ 1)$, $B_{12} = (1\ 1\ 0\ 0)$, $B_{13} = (1\ 1\ 0\ 1)$, $B_{14} = (1\ 1\ 1\ 0)$, $B_{15} = (1\ 1\ 1\ 1)$. Assuming that all possible combinations of user 4 bit blocks are ordered in their decimal equivalent, the example codes of 4 bits used for substituting user 4 bit blocks may be represented as a code set *C_sub = {C4, C6, C9, C1, C5, C10, C2, C14, C3, C0, C11, C7, C12, C8, C15, C13}* where the respective 4 bit codes are as follows.
C4=(0 0 1 0), C6=(0 1 0 1), C9=(1 0 0 1), C1=(1 0 0 0), C5=(1 1 0 0), C10=(0 1 1 0), C2=(0 1 0 0), C14=(1 1 0 1), C3=(0 0 0 1), C0=(0 0 0 0), C11=(0 1 1 1), C7=(0 0 1 1), C12=(1 0 1 1), C8=(1 0 1 0), C15=(1 1 1 1), C13=(1 1 1 0), the naming of a particular code is arbitrary. Hence the code taken to substitute a 4 bit user data is the code in the code set *C_sub* in the position equal to the decimal equivalent of the 4 bit user data block plus 1. For example if the present 4 bit user data block is (0 1 1 0) then the code used to substitute is *C2,* and if the next user 4 bit block is (1 0 1 0) the code used to substitute it is *C12.* The decimal equivalent for the 4 bit blocks of 4x1 column or 2x2 rectangular vector blocks is simply taking equivalent row vector arranging the bits serially row by row. The substitution based encryption algorithm may be represented as follows.

1. *User data segmented in 4 bit blocks B_i as row, column or rectangular block.*
2. *For the present 4 bit block, take decimal equivalent value D_i of B_i, i being a member of the set (0,1,2,…, 15).*
3. *Choose the code CX in the position (D_i + 1) of C_sub, X is a member of the set (0, 1, 2, ..., 15).*
4. *Substitute the present 4 bit user data block B_i by the 4 bit code CX.*

5. *Continue the process of substituting 4 bit blocks till the last 4 bit block is reached.*

For decoding or decrypting the coded data, the reverse process is applied with 4 bit segmentation of the received data. Now every 4 bit is a code from *C_sub* and hence to be decrypted as the 4 bit user data block for which the code was used. For example if the present 4 bit received code is *C7* then the 4 bit decrypted data is (*1 0 1 1)* as the code *C7* is in the 12th position of *C* and decimal equivalent of 4 bit user data is *11=12-1.* We further show an example of 64 bits 16x4 rectangular array block with its 4x4 codes array below in Fig.1, substitution codes *CX* are derived considering consecutive row data.

$$\begin{bmatrix} 0100001000111111 \\ 1100100101010001 \\ 1101100010110111 \\ 1110000010100110 \end{bmatrix} \Longrightarrow \begin{bmatrix} C5 & C9 & C1 & C13 \\ C12 & C0 & C10 & C6 \\ C8 & C3 & C7 & C14 \\ C15 & C4 & C11 & C2 \end{bmatrix}$$

Fig.1 Binary data 16x4 array converted to 4x4 code array

The methodology is shown in the following figures, Fig. 2 (a) and Fig. 2 (b).

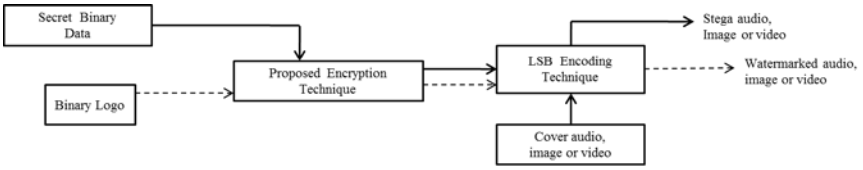


Fig. 2 (a) Encryption and LSB encoding for Steganography and Watermarking applications

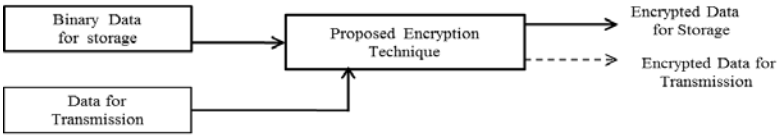


Fig. 2 (b) Encryption of User data for Storage and Transmission applications

As is clear from the above, the encryption and decryption is very simple to implement and very less amount of computations are required. For decrypting the received data, the code set *C_sub* is to be available at the receiving site as 64 bit binary array. Although it looks as if a 64 bit key is used to encrypt or decrypt, it is very different from the conventional encryption-decryption technique using 64 bit keys. In conventional encryption-decryption technique of data security, the 64 bit key is used for any 64 bit block of user data, whereas in our algorithm, a specific 4

bit code is used for a specific 4 bit user data block, and the codes can be chosen randomly. Here the maximum number of different combinations of 64 bit code set *C_sub* to be tried for decrypting the encrypted 64 bit original data is equal to 16x15x14x ……x2x1=2.0923×10^{13} , instead of $2^{64}=1.8447 \times 10^{19}$. Hence the light weight encryption algorithm seems to be very robust and provides security to the user data as expected.

1.3 Results and Discussions

We show applications of our proposed algorithm in LSB based steganography and digital watermarking, secured storage of important data in a computer system and secured transmission of biometric application data. In all cases we take human iris [9] gray scale image data of size 120x200 pixels as user or application generated data for the above said applications. We use the JPG formatted image of *baboon* of size 512x512 pixels as the cover image. For LSB encoding of the cover image, we consider a random binary stream of size 1x262000 bits, for watermarking a digital product, which is here the *baboon* gray scale image, we consider a partial rectangular human iris of size 120x100 pixels gray scale image. For encrypting a stored data in a computer system, we consider one eigeniris [11] gray scale image of size 120x200 pixels, and for transmission example of a biometric application data we consider the 120x200 pixels size rectangular human iris for encryption. We apply the encryption algorithm we propose here on the application data for encryption, all the application data are in non-compressed form although the encryption technique can be applied on compressed data also. For watermarking a product image, we apply LSB encoding technique, and are open to any other technique of watermarking. A detail study of our encryption technique in various applications shall be reported very soon. While applying LSB encoding on the cover *baboon* image, we obtained the 512x512 8th bit rectangular array, segmented it to 64x64 block array each block of 64 bits, choosing a 64 bit block randomly and then LSB encoding the 4 bit encrypted blocks row wise.

Fig, 3 shows the cover *baboon* jpg image of size 512x512 pixels, the LSB encoded cover image where a random binary stream of 262000 bits are first encrypted and then hidden in the cover image's 8th bit of each pixel is shown in Fig. 4. The capacity of the cover image is 512x512=262144 bits in each bit plane and the payload data is 262000 bits. The cover image after LSB encoding, called stego image has a PSNR value of 54.2 dB. Thus LSB encoding for steganography applications, after encrypting the random data by our technique produces better quality stego image than only LSB encoding the same size cover image [3].

For encrypting important user data in a computer system, we consider the biometric database of eigenirises for iris recognition system [11, 12]. We have 10 eigeni-

rises for each person's identification, recognition or validation applications, and each eigeniris is rectangular size of 120x200 pixels. The eigeniris is first converted to binary stream of size 204152 bits and then encrypted using the proposed algorithm and stored back to the database. Fig. 5 shows one rectangular eigeniris for encryption, and Fig. 6 shows the encrypted eigeniris having a PSNR value of 50.4 dB.

Fig. 3 Cover Image 512x612 pixels, JPG

Fig. 4 Stegoimage with LSB encoded encrypted data

For watermarking a gray scale image product, we consider the partial rectangular human iris of size 120x100 pixels as the logo for watermarking. The iris is first converted to binary data having size 97960 bits, encrypt it using our algorithm and then LSB encode the product image. Fig. 7 shows the logo, and Fig. 8 shows the watermarked product image having a PSNR value of 58.5 dB. Although we consider here partial human iris as logo, the logo may be any other logo and watermarking may be done any other technique like DCT, SVD or DWT block based pixel encoding other than LSB encoding.

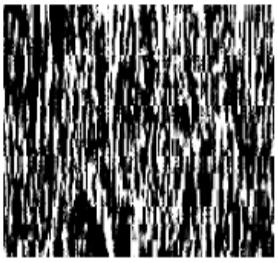


Fig. 5 Rectangular Eigen Iris, 120x200 pixels

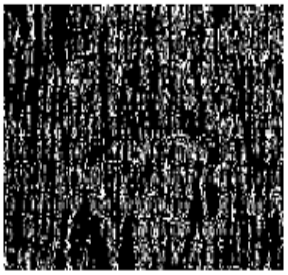


Fig. 6 Encrypted Eigen Iris, 120x200 pixels

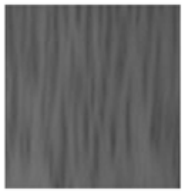
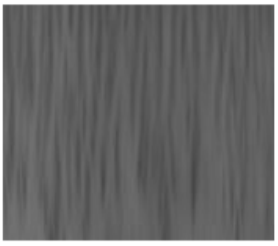


Fig. 7 Partial Iris Logo, 120x100 pixels

Fig. 8 Watermarked cover image, 512x512 pixels

To show encrypted biometric application data transmission to a local or distant server site, for recognition, identification or validation requirement as reported in [12], we consider an uncompressed rectangular human partial iris of size 12x200 pixels. The binary equivalent iris of 167612 bits is encrypted and transmitted, without applying any forward error correcting coding. Fig. 9 shows the partial rectangular iris and Fig. 10 shows the encrypted partial iris after conversion to gray scale image. The PSNR value of the encrypted iris is 69. 7dB.



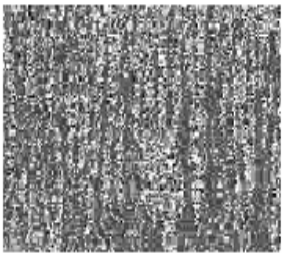


Fig. 9 Rectangular Iris, 120x200 pixels

Fig. 10 Encrypted Rectangular Iris, 120x200 pixels

1.4 Conclusion

We report here a simple encryption encoding technique that can be tied with LSB encoding for various data hiding and security applications. We observe from the results of using our proposed encryption algorithm on the application data for steganography, watermarking, secured data storage, and secured transmission of application data, that the algorithm is simple in implementation, requires moderate

computations with simple mathematical rules of substituting a specific block of user data bits by a specific code of same size from a code set. The algorithm is light weight in terms of computations for encryption and decryption, and effective for data encryption of application data that requires moderate computational burden and yet provides security. We also observe that the LSB encoding after encrypting the data produces better quality stegoimage or watermarked image, as is evident from the respective PSNR values, than the reported LSB encoding techniques in the literature. The encrypted image data that is stored in a database or transmitted through a channel to local or distant server sites also have good quality as is obvious from their PSNR values. We have shown results for steganography, watermarking using encryption and LSB encoding, and secure data storage and secure data transmission using encryption only. The encryption algorithm is expected to be used in such devices where light weight data processing algorithms are preferred because of limited battery power.

1.5 References

1 C. K. Chan, and L. M. Chen, "Hiding Data in Images by Simple LSB Substitution, "Pattern Recognition, Vol. 37, No. 3, March 2004, pp 469-474

2. S. H. Liu, T. H. Chen, H. X. Yao, and W. Gao, "A Variable Depth LSB Data Hiding Techniques in Images, "Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, China, August 2004, pp 3990-3994

3. Chin-Chen Chang, and Wen-Chuan Wu, "A Novel Data Hiding Scheme for Keeping High Stego-Image Quality," Multi-Media Modelling Conference Proceedings, 2006 12th International IEEE conference, 4-6 Jan, pp 225-232

4. B. G. Mobasseri, and R. J. Berger, "A foundation for watermarking in compressed domain, "IEEE Signal Processing Letters, vol. 12, pp. 399-402, May 2005

5. R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm, "Pattern Recognition, vol. 34, no. 3, pp. 671-683, 2001

6. Nan-L Wu, and Min-Shiang Hwang, "Data Hiding: Current Status and Key Issues," International Journal of Network Security, vol. 4, no. 1, pp. 1-9, January 2007

7. Jung-Hee Seo, and Hung-Bog Park, "Data-Hiding Method using Digital Watermark in the Public Multimedia Network," International Journal of Information Processing Systems, Vol. 2, No. 2, pp. 82-87, June 2006

8. Cheng-Hsing Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution, "Pattern Recognition 41, 2008, pp. 22674-2683

9. J. Daugman, "How iris recognition works, "Circuits and Systems for Video Technology," IEEE Transactions, 14 (1), pp 21-30, 2004

10. R. Kalita, S. Majumder, and Md. Anwar Hussain, "Multidimensional Multimetric Novel and Simple Techniques for Iris Recognition System," International Journal of Recent Trends in Engineering [ISSN 1797-9617], pp. 161-166, Vol. 3 No. 3, Academy Publishers

11. Md. Anwar Hussain, "Eigenspace Based Accurate Iris Recognition System," Proceedings of 2010 Annual IEEE India Conference (INDICON), Kolkata, India

12. Md. Anwar Hussain, "Wavelet Processing of Human Iris for Accurate Recognition System," Proceedings of International Conference on Information Technology, Systems and Management (ICIISM-2012), pp. 30-36, March 2012, Dubai, UAE