International Journal of Engineering & Technology, 7 (2.9) (2018) 82-85

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Tiny encryption algorithm and pixel value differencing for enhancement security message

Robbi Rahim [1] *, Danadyaksa Adyaraka [2], Sulfikar Sallu [3], Eri Sarimanah [4], Muhammad Mihram Rahman [5], Nuke L. Chusna [6], Sitti Hartinah [7], Endang Kusuma Astuti [8], Nuning Kurniasih [9]

[1] *School of Computer and Communication Engineering, Universiti Malaysia Perlis, Kubang Gajah, Malaysia*
[2] *Komunitas Kolaborasi Publikasi Indonesia, Medan, Indonesia*
[3] *Faculty of Information Technology, Universitas Sembilanbelas November Kolaka, Kolaka, Indonesia*
[4] *Faculty of Teaching and Education, Universitas Pakuan, Bogor, Indonesia*
[5] *Iai Ddi Polewali Mandar, Polewali Mandar, Indonesia*
[6] *Department of Informatics, Universitas Krisnadwipayana, Jakarta, Indonesia*
[7] *Universitas Pancasakti, Tegal, Indonesia*
[8] *Universitas Darul Ulum Islamic Center, Semarang, Indonesia*
[9] *Faculty of Communication Science, Library and Information Science Program, Universitas Padjadjaran Bandung, Indonesia*
**Corresponding author E-mail: usurobbi85@zoho.com*

Abstract

A combination of algorithms to improve text security is possible, Tiny Encryption Algorithm and Pixel Value Differencing are two possible combinations of algorithms. Cryptography and steganography processes can be done to secure messages with two stages, encryption for the first stage and the second one for steganography. Using this two-stage make it difficult for irresponsible parties to know information.

Keywords: *Tiny Encryption Algorithm; Pixel Value Differencing; Strengthen Security, Enhancement Security*

1. Introduction

Data security and confidentiality issues are one of the most critical aspects of Information Systems [1]–[6], information will no longer be useful if it has been intercepted or hijacked by others. Security demands are becoming increasingly sophisticated, mainly when the data is transmitted, and the data is highly confidential data, so it must be safeguarded not to be hijacked by others [7]–[13].

There are various ways used to protect data such as the provision of a password, but this way can be hacked by the hijackers because the user can create the possibility of the word used as a password by the party who locked it [14]–[19]. Another way is with ciphertext, in this way the data to be stored encoded first, but this way can attract suspicions by other parties, so the user will try to decode the coding so that the data can be hijacked [20]–[22]. Therefore it takes a way that can make the pirates unsuspecting, and the user does not immediately know that there is data stored and that way is Steganography.

Steganography is one method that can be used to secure information [23]–[25]. Steganography is different from cryptography or other information security methods, and this method is to hide information or messages into other media such as digital images, text, sound or video so as not to cause suspicion of others. Steganography requires two properties, information and cover media. Cover media is used to hide information, i.e., digital images. The embedded of information on the digital image media is performed on the pixel bits contained in the image. The use of a digital image as a cover media has advantages because the senses of human vision have limitations to the color so that with such limitations humans are difficult to distinguish the original digital image with a digital image that has been inserted a secret message [23], [26].

Pixel value differencing (PVD) method is one method that can be used in making steganography. This method offers a larger message storage capacity, with better image quality compared to other methods[23]. To increase the level of security of information to be inserted into the image, steganography can be combined with encryption, so the inserted information will not be easy to read by irresponsible people. One of the encryption that can be used is TEA (Tiny Encryption Algorithm) [27], [28].

Tiny Encryption Algorithm (TEA) is a password algorithm created by David Wheeler and Roger Needham from Computer Laboratory, Cambridge University, England in November 1994. This algorithm is a block cipher encryption algorithm designed for minimal memory usage with process speed maximum, this combination it can produce reliable security message so no other party will know the message quickly.

2. Methodology

Pixel Value Differencing (PVD) scheme uses the value of the difference between two consecutive pixels in the block to determine how many secret bits must be embedded. There are two types of tables of quantization ranges in Wu and Tsai methods, the first based on selecting the range width [8, 8, 16, 32, 64, 128], to provide a large capacity. The second is based on selecting the range width [2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64], to provide high imperceptibility [23], [29]. Most of the related studies focus on capacity building using LSB and insertion processes, so the approach is too aligned with the LSB approach. This study provides a new perspective that if choosing the right width for each range and using the proposed method, then better picture quality and higher capacity can be obtained [26].

The Tiny Encryption Algorithm (TEA) is a password algorithm created by David Wheeler and Roger Needham from Computer Laboratory, Cambridge University, England in November 1994. This algorithm is a block cipher encryption algorithm designed for minimal memory usage with process speed maximum. The TEA encryption system uses a Feistel network process by adding a mathematical function of addition and subtraction as an inverting operator other than XOR. It is intended to create non-linearity properties. Two-way shifts (left and right) cause all key bits and data to mix repeatedly [27], [28].

TEA processes 64-bit inputs at a time and generates 64 bits of output. TEA stores 64-bit inputs into L0 and R0 respectively 32 bits. L0 is a Left variable for storing 32-bit input and R0 is the Right variable to store 32-bit keys. While 128 key bits are stored into k [0], k [1], k [2], and k [3] each containing 32 bits, K [0] to K [3] are the variables for key storage. It is assumed that this technique is enough to prevent the use of exhaustive search techniques effectively. The output results will be stored in L16 and R16, L16 and R16 are Left and Right variables to store the output[30].

The delta number is derived from the golden number, used delta = (√5 -1) 231. Different multiple delta numbers are used in each round so that no bits of the multiplication does not change regularly. In contrast to the Feistel structure which initially operated only one side, i.e. the right side with an F function, in the TEA algorithm both sides are operated with a similar function.

3. Results and discussion

To perform encryption, the process begins with 64-bit bright-bit input of the text. Then 64-bit bright text is divided into two parts, i.e., the left side (L0) as much as 32-bit and the right side (R0) as much as 32-bit. Each piece of bright text will be operated independently. R0 (z) will be shifted left four times (4) and added with the key k [0]. Meanwhile, z is coupled with sum (delta) which is a constant. This addition result is XOR with the previous addition. Then it is XOR with the result of the addition of z which is shifted to the right by five (5) times with the key k [1]. The result is then added with L0 (y) which will be R1.

The left side will experience the same process with the right side. L0 (y) will be shifted left four times (4) then added with k key [2]. Meanwhile, Y plus the sum (delta). The result of this addition is XOR with the previous addition. Then it is XOR with the result of the addition of Y being shifted to the right by five (5) times with the key k [3]. The result is then added with R0 (Z), which will be L1. The encryption process is as follows:

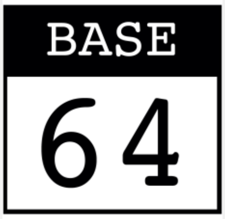
Plaintext = rembulan
Key = RIDHO ADI PUTRAA
For plaintext into 2 (two) parts
R = REMB
L = ULAN
After that for the 4 (four) key part so as below
K [0] = RIDH
K [1] = OspasiAD
K [2] = IspasiPU
K [3] = TRAA
After that change the plaintext into binary as shown in Table 1.

Table 1: Plaintext Binaries and Key

ASCII	BINER
rembulan	R(01110010) E(01100101) M(01101101) B(01100010) U(01110101) L(01101100) A(01100001) N(01101110)
RIDHO ADI PUTRAA	01010010 01001001 01000100 01001000 01001111 00100000 01000001 01000100 01001001 00100000 01010000 01010101 01010100 01010010 01000001 01000001

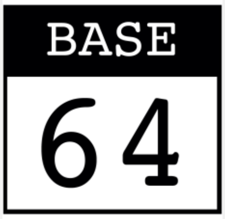
Cipher R (Z) = 01110010 01100101 01101101 01100010
Cipher L(Y) = 01110101 01101100 01100001 01101110
K [0]: 01010010 01001001 01000100 01001000
K [1]: 01001111 00100000 01000001 01000100
K [2]: 01001001 00100000 01010000 01010101
K [3]: 01010100 01010010 01000001 01000001
After that, Z cipher will experience a shift bit to the left as much as [4] bits and to the right as much as 5 bits, the following results.
Initial Condition Cipher R (Z) = 01110010 01100101 01101101 01100010
After the shift left and right to:
Zsl (Z shift left): 00100110 01010110 11010110 00100111
Zsr(R shift right): 01001100 10101101 1010110001001110
Zsl is added with the value of K [0] as below:Zsl: 00100110 01010110 11010110 00100111K [0]: 01010010 01001001 01000100 01001000
01110110 01010111 11010110 01110111Zsr is added with the value of K [1] so as belowZsr: 01001100 10101101 10101100 01001110
K [1]: 01001111 00100000 01000001 01000100
01001111 10101101 11101101 01001110Then Cipher R (Z) does not experience a bit shift added with a delta number, where the delta numbers are constantly used, i.e., F9A3B4E7 or in binary 11111001 10100011 10110100 11100111.
R (Z): 01110010 01100101 01101101 01100010
Delta: 11111001 10100011 10110100 11100111
11111011 01100111 01111101 11100111
Then in XOR it is with Zsl cipher plus K [0]:
Results R (Z) + Delta: 11111011 01100111 01111101 11100111
Results Zsl + K [0]: 01110110 01010111 11010110 01110111
Sum Result: 11111111 01110111 11111111 11110111
Next is to do XOR of Zsr cipher plus K [1]
Results R (Z) + Delta: 11111111 01110111 11111111 11110111
Results Zsr + K [1]: 01001111 10101101 11101101 01001110
Sum Result: 11111111 11111111 11111111 11111111
For the L (Y) cipher, the process is necessarily same as the R (Z) cipher, i.e., the L (Y) cipher is also shifting a bit to the left as much as 4 bits and to the right as much as 5 bits. Cipher L (Y): 01110101 01101100 01100001 01101110, the value of the cipher is shifted by 4 bits and 5 bits, and it will produce the value of shift as below
Ysl: 01010110 11000110 00010110 11100111
Ysr: 10101101 10001100 00101101 11001110Lsl is added with k [2]:
Ysl: 01010110 11000110 00010110 11100111
K [2]: 01001001 00100000 01010000 01010101
01011111 11100110 01010110 11110111
Ysr is added with k [3]:
Ysr: 10101101 10001100 00101101 11001110
K [3]: 01010100 01010010 01000001 01000001
11111101 11011110 01101101 11001111
Ciphers L (Y) that do not experience shifts added to delta
Binary L (Y): 01110101 01101100 01100001 01101110Binary Delta: 01011111 11001111 01011111 11110111
01111111 11101111 01111111 11111111
Then the value of the above calculation coupled with the value of K [2]
Result L (Y) + Delta: 01111111 11101111 01111111 11111111
Results Ysl + K [2]: 01011111 11100110 01010110 11110111
Result: 11111111 11111111 11111111 11111111
Next is to do the XOR of the Ysr cipher plus K [3]
Result L (Y) + Delta: 11111111 01110111 11111111 11110111
Results Ysr + K [3]: 11111101 11011110 01101101 11001111
Result: 11111111 11111111 11111111 11111111
The result of the R (Z) cipher is added with an unshifting L (Z) cipher, which the result will be the L1 (Y1) cipher for the next round. Similarly, the end product on the L (Y) cipher will be added with the non-shifting R (Z) cipher to be the R1 (Z1) cipher in the next round:
R (Z): 01110010 01100101 01101101 01100010
L (Y): 01110101 01101100 01100001 01101110

01110111 01101101 01101101 01101110 → L1 (Y1)
L (Y): 11111111 11111111 11111111 11111111
R (Z): 11111111 01110111 11111111 11110111
11111111 11111111 11111111 11111111 → R1 (Z1) the end result is as follows:
L1 (Y1) = 01110111 01101101 01101101 01101110
R1 (Z1) = 11111111 11111111 11111111 11111111Process the decryption of the TEA algorithm as well as the encryption process. It's just the difference in the key scheduling is the encryption process for cipher R that has shifted bits to the left as much as 4 bits used the key k [0] in the process description used key k [1], for cipher R that shift to the right as much as 5 bits use key [1] in process description using key k [0]. So is the case with L's cipher, in the encryption process for L cipher which has a left shift of 4 bits using the k key [2] in the description process used key k [3]. For the L cipher, having a right shift of 5 bits the key k [3] is used in the description process using k key [2].
Example of steganography process, if a known message to be inserted in the form of binary = 01110111 01101101 01101101 01101110 11111111 11111111 11111111 11111111, the next stage is to take the pixel value of an image, assuming an image with the name base64.bmp, the pixel value obtained by using mat lab software.



45	102	157
66	103	165
153	169	192

After getting the pixel value from the picture and perform embedded using Pixel Value Differencing algorithm, the pixel in the image will be changed into below:



96	71	167
76	113	175
163	179	202

4. Conclusion

The combination of 2 different algorithms with different processes can improve the security of peers from irresponsible parties, TEA and PVD are algorithms that can be combined and used well where the image media is used as the cover of the message storage you want to hide. The development of this algorithm is possible by using Double Pixel Value Differencing with other cryptography algorithms.

References

[1] A. Putera, U. Siahaan, and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *Int. J. Secur. It is Appl*., vol. 10, no. 8, pp. 173–180, Aug. 2016.
[2] R. Rahim, "128 Bit Hash of Variable Length in Short Message Service Security," *Int. J. Secur. It is Appl*., vol. 11, no. 1, pp. 45–58, Jan. 2017.
[3] R. Rahim, M. Dahria, M. Syahril, and B. Anwar, "Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression," *World Trans. Eng. Technol. Educ*., vol. 15, no. 3, pp. 292–297, 2017.
[4] H. Hartono, D. Abdullah, and A. S. Ahmar, "A New Diversity Technique for Imbalance Learning Ensembles," *Int. J. Eng. Technol*., vol. 7, no. 2, pp. 478–483, Apr. 2018.
[5] D. Abdullah, Tulus, S. Suwilo, S. Effendi, and Hartono, "DEA Optimization with Neural Network in Benchmarking Process," *IOP Conf. Ser. Mater. Sci. Eng*., vol. 288, no. 1, p. 012041, Jan. 2018.
[6] R. Rahim et al., "Combination Base64 Algorithm and EOF Technique for Steganography," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 012003, Apr. 2018.
[7] R. Rahim, "Man-in-the-middle-attack prevention using interlock protocol method," *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 22, pp. 6483–6487, 2017.
[8] M. Attaran and I. VanLaar, "Privacy and security on the Internet: how to secure your personal information and company data," *Inf. Manag. Comput. Secur.* vol. 7, no. 5, pp. 241–247, 1999.
[9] K. J. Fitzgerald, "Security and data integrity for LANs and WANs," *Inf. Manag. Comput. Secur*. vol. 3, no. 4, pp. 27–33, 1995.
[10] E. Kartikadarma, T. Listyorini, and R. Rahim, "An Android mobile RC4 simulation for education," *World Trans. Eng. Technol. Educ.*, vol. 16, no. 1, pp. 75–79, 2018.
[11] H. Nurdiyanto, R. Rahim, and N. Wulan, "Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement," *J. Phys. Conf. Ser*., vol. 930, no. 1, p. 012005, Dec. 2017.
[12] D. Abdullah, R. Rahim, D. Apdilah, S. Efendi, T. Tulus, and S. Suwilo, "Prime Numbers Comparison using Sieve of Eratosthenes and Sieve of Sundaram Algorithm," in *Journal of Physics: Conference Series*, 2018, vol. 978, no. 1, p. 012123.
[13] R. Rahim, H. Winata, I. Zulkarnain, and H. Jaya, "Prime Number: an Experiment Rabin-Miller and Fast Exponentiation," *J. Phys. Conf. Ser*., vol. 930, no. 1, p. 012032, Dec. 2017.
[14] K. Neeraja, P. Rama Chandra Rao, D. Suman Maloji, and D. Mohammed Ali Hussain, "Implementation of security system for bank using open CV and RFID," *Int. J. Eng. Technol*., vol. 7, no. 2–7, p. 187, Mar. 2018.
[15] N. Srinivasu, O. Sree Priyanka, M. Prudhvi, and G. Meghana, "Multilevel classification of security threats in cloud computing," *Int. J. Eng. Technol*., vol. 7, no. 1.5 Special Issue 5, pp. 253–257, 2018.
[16] Y. H. Kim and G. W. Bang, "Development of security camera combined beacon signal for transmission of disaster and crime situation as well as tracking location," *Int. J. Eng. Technol*., vol. 7, no. 2, pp. 141–144, 2018.
[17] D. Abdullah et al., "A Slack-Based Measures for Improving the Efficiency Performance of Departments in Universitas Malikussaleh," *Int. J. Eng. Technol*., vol. 7, no. 2, pp. 491–494, Apr. 2018.
[18] A. E. S. Kacaribu and Ratnadewi, "Multiplying cipher images on visual cryptography with ElGamal algorithm," *in 2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2015, pp. 159–162.
[19] Ratnadewi, R. P. Adhie, Y. Hutama, A. Saleh Ahmar, and M. I. Setiawan, "Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC)," *J. Phys. Conf. Ser*., vol. 954, no. 1, p. 012009, Jan. 2018.
[20] L. Legito and R. Rahim, "SMS Encryption Using Word Auto Key Encryption," *Int. J. Recent Trends Eng. Res*., vol. 3, no. 1, pp. 251–256, 2017.
[21] R. Rahim, D. Hartama, H. Nurdiyanto, A. S. Ahmar, D. Abdullah, and D. Napitupulu, "Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm," *J. Phys. Conf. Ser*., vol. 954, no. 1, p. 012008, 2018.
[22] R. Rahim et al., "Searching Process with Raita Algorithm and its Application," *J. Phys. Conf. Ser*., vol. 1007, no. 1, p. 012004, Apr. 2018.
[23] H. Nurdiyanto and R. Rahim, "Enhanced pixel value differencing steganography with government standard algorithm," *in 2017 3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 366–371.
[24] R. Bhardwaj and V. Sharma, "Image Steganography Based on Complemented Message and Inverted Bit LSB Substitution," in *Procedia Computer Science*, 2016, vol. 93, pp. 832–838.
[25] M. Ramalingam and N. A. M. Isa, "A steganography approach over video images to improve security," *Indian J. Sci. Technol.*, vol. 8, no. 1, pp. 79–86, 2015.
[26] H. Zhang, Q. Guan, and X. Zhao, "Steganography based on adaptive pixel-value differencing scheme revisited," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8389 LNCS, pp. 32–47.
[27] M. Shoeb and V. K. Gupta, "a Crypt Analysis of the Tiny Encryption Algorithm in Key Generation," *Int. J. Commun. Comput. Technol.*, vol. 01, no. 0138, pp. 5–123, 2013.

[28] Amandeep and G. Geetha, "Implications of bitsum attack on tiny Encryption Algorithm and XTEA," *J. Comput. Sci.*, vol. 10, no. 6, pp. 1077–1083, 2014.

[29] T. Zhang, W. Li, Y. Zhang, and P. Xijian, "Detection of LSB matching steganography based on distribution of pixel differences in natural images," in *IASP 10 - 2010 International Conference on Image Analysis and Signal Processing*, 2010, pp. 548–552.

[30] H. R. Ismaeel, "Apply Block Ciphers Using Tiny Encryption Algorithm (TEA)," *Baghdad Sci. J.*, vol. 7, no. 2, 2010.