Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HARDWARE IMPLEMENTATION METHOD OF SECRET DATA SECURITY ON FPGA BASED ON ZIG-ZAG MAP ENCRYPTION AND STEGANO ALGORITHMS

BAYU KUMORO YAKTI[1], SUNNY ARIEF SUDIRO[2], SARIFUDDIN MADENDA[3], AND SURYADI HARMANTO[4]

[1,3,4] Doctoral Program in Information Technology Gunadarma University, Jakarta – Indonesia

[2] Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K, Jakarta – Indonesia

E-mail: [1]bayuyakti@staff.gunadarma.ac.id, [2]sunnyariefsudiro@ieee.org, [3]sarif@staff.gunadarma.ac.id, [4]misdie@staff.gunadarma.ac.id

ABSTRACT

The internet has grown so rapidly that most of the individuals prefer to use the internet as the main medium for transferring data. Data protection is very important especially when sending secret data from one place to another. This paper proposes a zig-zag mapping transposition encryption-decryption algorithms and Least Significant Bit (LSB) steganography algorithm for data security improvement. The hardware implementation methods of these algorithms into Field Programmable Gate Array (FPGA) for data protection in real-time communication are also offered. The FPGA Intellectual Property core (IP core) resulting employed minimal LUTs resources. The encryption and steganography algorithms occupy 107 LUTs and take 1.821 ns for each 64 bits data processing. Whereas steganalysis and decryption algorithms needed108 LUTs and 2.172 ns processing time.

Keywords: *Encryption-decryption, LSB steganography, LUTs FPGA resources, Zig-zag Mapping Transposition*

1. INTRODUCTION

Various methods of data security have been developed and will continue to grow. A method can be implemented in the form of software or hardware. Software uses programming language instructions, and their embodiments are implemented using devices (programs) designed to allow computers to read those instructions, while hardware is a device where functions are implemented to produce computations [1]. Software applications, applications must be installed first which refers to layer 7 Open Systems Interconnection model (OSI) or application layer. Hardware applications are implemented at OSI layer 1 or physical layer. The physical layer is more difficult to hack because the data is represented as electrical quantities. The speed of the process carried out at the physical layer is also faster than the software application layer [2].

Cryptography and Steganography are methods for encrypting and hiding data. Cryptography uses a reversible algorithmic process to convert a plain text message (readable text) into a ciphertext message (readable text) based on an algorithm known to the sender and receiver. Ciphertext messages can be restored to their original, plain text form. Messages that have been encoded cannot be read by anyone except the intended recipient [3]. The most frequently used methods to encrypt a given data are substitution and transposition techniques [4] [5]. Rail Fence cipher is one of the basic transposition ciphers [6]. The advantage that the Rail Fence cipher has over other transposition ciphers (such as sawtooth) is the variable spacing between successive letters. That is, the letters do not need to be arranged in fixed vertical descending columns but can also be arranged in a zigzag manner [7]. The weakness of the Rail Fence cipher comes from the cipher's key. The number of keys is small enough for a cryptanalyst to break. These ciphers basically offer no communication security and are quite visible, making them easily vulnerable to attacks. It cannot be used to encrypt images that contain large areas of a single color. Although Rail Fence ciphers are weak, rail fence ciphers can be mixed

with other cryptographic algorithms such as substitution ciphers to enhance security [8].

Zigzag Cipher is one of many classical cryptographies that uses character permutations in its transposition technique. The original message will remain unread unless the person has the decryption key. Zigzag Cipher is a classic cryptographic algorithm that is not secure. In the Zigzag Cipher algorithm, the transposition used during encryption must be the same as that used during decryption to obtain the same plaintext result. Based on a linear graph, the length of the plaintext is proportional to the processing time, which means that the longer the plaintext, the longer it takes to encrypt [9].

Based on the explanation of the transposition technique, this research was conducted using a mapping key or index mapping. Input data, both in terms of form and amount of data, are flexible. This research is intended so that all forms of data can be used as input data for the transportation cipher.

Steganography is a technique to hide the existence of secret information in other objects, especially in an image [10]. Altaay conducted a comparative study between steganography and cryptography methods [11]. The advantage of steganography over cryptography is that confidential data or information is hidden or not physically visible, so it does not attract attention as an object of surveillance. Messages encrypted by cryptography are physically visible, thus attracting attention to being hacked. This shows that steganography is more secure than cryptography when it comes to sending confidential information. The perfect Steganography technique embeds data into the carrier image with a high level of security [12].

Research by Jatin Chaudhari and K.R.Bhatt in [13] proposed an implementation of steganography with X-Box tables. the drawback of this implementation method is that the resources obtained are large (2411 slices) because the method must call the X-BOX table data repeatedly. Research by Kunjan Pathak and Manu Bansal in [14] proposed an implementation of the steganography method using LSB using grayscale images by ignoring the encryption-decryption process on the input data. Research by Chiung-Wei Huang, Changmin Chou, Yu-Che Chiu, and Cheng-Yuan Chang in [15] proposed of implementing a steganography method similar to that of the Kunjan Pathak researcher with a difference in the implementation of LSB in the last 3 bits.

Encryption-decryption and steganography methods can be used to increase security. The approaches from previous studies that can be used to optimize FPGA-based encryption and steganography in this study are:

1. Many FPGA implementations use only one method, steganography, but previous researchers still use grayscale images for steganography.
2. The FPGA resources used are still very large.

This research raises the topic of implementing transposition algorithm with key mapping, and steganography with RGB image into FPGA hardware with minimal resources. The Steganography section adopts the LSB insertion model for each image pixel. This study aims to develop algorithms and prototypes as a solution to the problems and shortcomings of the techniques that have been carried out by previous researchers that can overcome the optimization of FPGA hardware.

2. LITERATURE REVIEW

2.1 Cryptography

The National Institute of Standards and Technology (NIST) defines the term computer security as the protection afforded to automated information systems for the purpose of maintaining the integrity, availability, and confidentiality of information security resources [5]. The three main goals of network security include:

1. Confidentiality: Only authorized users can access the submitted information.
2. Integrity: Only authorized users can make changes to submitted information and programs.
3. Availability: Authorized users can use all services provided by the system and permission should not be denied.

A security attack is an attempt to modify, disable, steal information or even gain unauthorized access to exploit data sent over a communication channel. The substitution technique is a cryptographic technique that involves replacing the given plaintext letters to other letters, numbers, or symbols. The plain text

bit pattern is replaced with the ciphertext bit pattern. While the transposition technique involves a kind of permutation on plaintext letters [4]. Table I is the result of technical research that has been carried out.

Table 1: Comparison Of Substitution And Transposition [4]

Aspect	Substitution Technique	Transposition Technique
Principle	Changes its identity but retain its position	Changes its position but retain its identity
Complexity	Simple process	More complex than substitution technique
Security	Easy to crack the code	Difficult to crack the code
Access	Unauthorized users can easily access the data	Difficult for intruders to access the information
Completion time	The time complexity of encryption and decryption is less	The time complexity of encryption and decryption is high
Example	Caesar cipher	Columnar cipher

2.2 Steganography

Steganography maintains data integrity, this means that there will be no modification in the information content during communication. An overview of steganography can be seen in Figure 1. The input data is a cover image and confidential information which can be in the form of text or images. Steganography blocks generate stego images, steganalytic models take stego images as input data to detect and possibly extract confidential information. In some methods the input data image is normalized or the stego image is given as the output data. Text data, color images or grayscale are usually used as secret media [12].

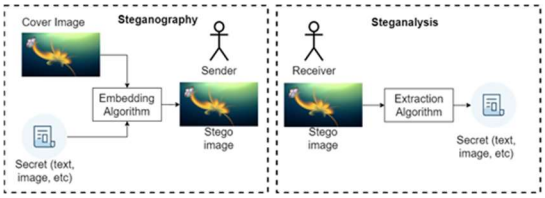


Figure 1 : Basic Overview Of Steganography

2.3 FPGA

The Security in the hardware area and its integration between different security services in embedded systems still requires methods, techniques and most importantly prototypes that are efficient in performance and power consumption. For this reason, it is still possible to create, improve and develop new methodologies and expand the use of this emerging technology in applications in fields such as education and industry, using programmable circuits and other embedded system platforms [16].

FPGA provide reconfiguration as well as robustness for image processing. FPGA functions can develop alternative approaches based on embedded FPGA systems for image processing. Without requiring hardware replacement, the use of FPGA-type Devices can extend product life by updating data stream files. In FPGA-based designs, the hardware area used is provided in the form of LUTs or Configurable Logic Block (CLB) slices. However, for the comparison of designs based on similar FPGA devices, all resources must be considered. Some of the most commonly used FPGA resources are number of LUTs and number of slices [17] [18]

FPGA have the ability to hold the entire system on a single chip and also allow in-platform testing and system debugging. In addition, it offers the opportunity to leverage hardware/software co-design to develop high-performance systems for various applications by including processors, on-chip buses, memory, and hardware accelerators for specific software functions. The FPGA was chosen for its ability to be reconfigured [19].

3. PROPOSED METHOD

The research was conducted by combining development algorithms and development methods that can optimize FPGA performance. Thus, the method for the desired result is the use of minimal components so that the resources used are smaller than previous studies. The object of this research is an RGB image in the form of 8-bit binary as a cover image to which binary data will be inserted. Figure 2 is the process flow from transposition encryption to steganography and Figure 3 is the process flow of steganalysis (data extraction) to transposition decryption.

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

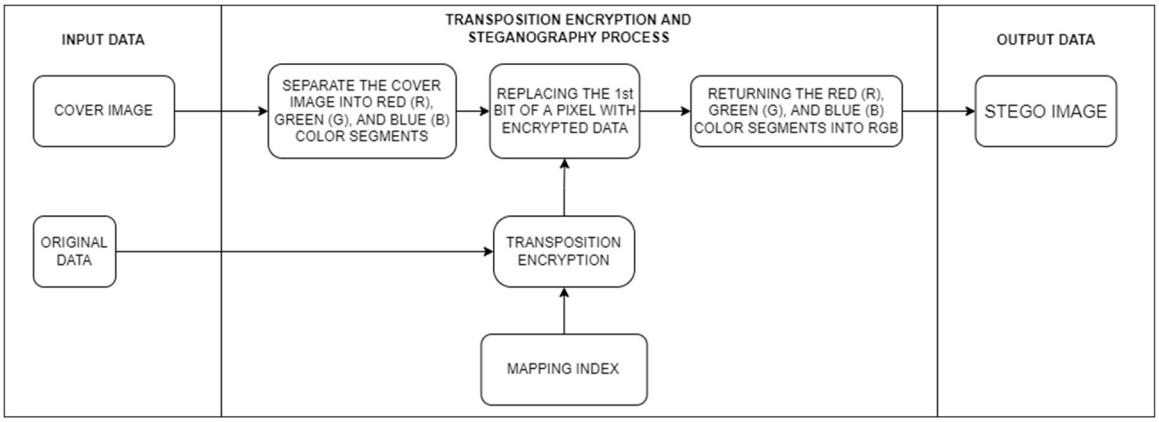


Figure 2 : Process flow of mapping transposition encryption and steganography

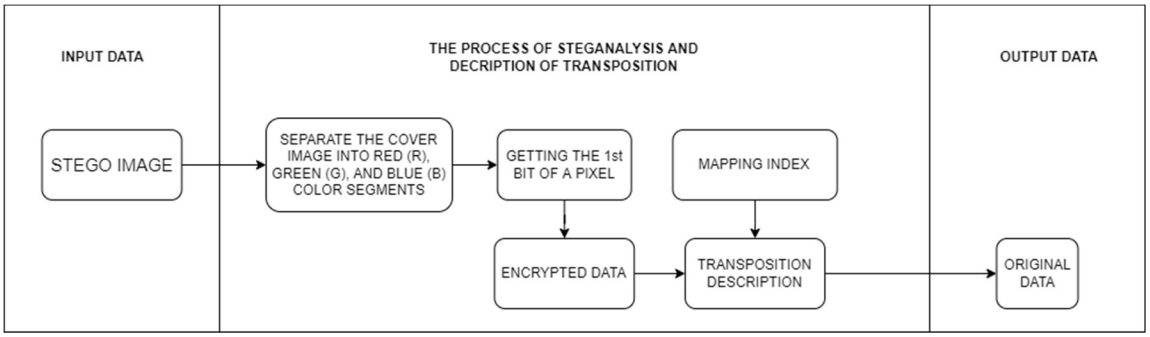


Figure 3 : The Process Flow Of Steganalysis And Decryption Of Mapping Transposition

The secret data is pasted on the cover image with a key that has been created on the FPGA hardware to become a stego image. The key used for transposition encryption is the same as the key for transposition decryption. The mapping key used is a matrix that has dimensions of 8x8 which contains decimal values from 1 to 64 [20] [21]. Encryption in the mapping transposition method is the process of transferring data based on addresses or mapping index parameters. The data resulting from the transfer is referred to as encrypted data. So, the mapping index is the position adjusted to the key.

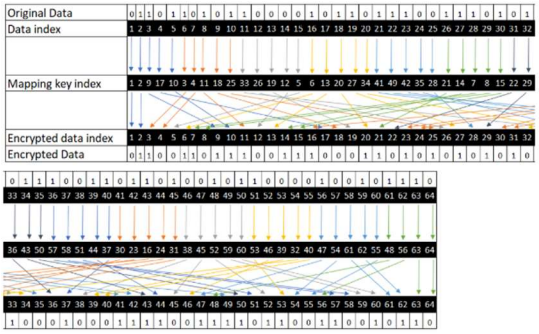


Figure 4 : Transposition Encryption Illustration

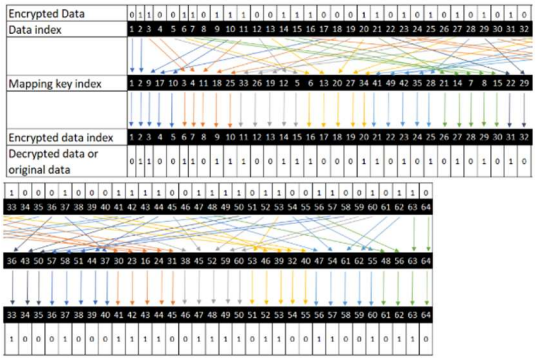


Figure 5 : Transposition Decryption Illustration

Figure 4 is an illustration of data that has been transposed using a mapping index. Decryption in the mapping transposition method is the process of returning the data position to its original position using the same mapping index key as encryption. Figure 5 is an illustration of data that has been transposed using a mapping index. The matrix is done in a zig-zag pattern and converted into vector form to produce the key shown in Figure 4 and Figure 5 of the 4th row.

LSB steganography is done from the first pixel up to the amount of data. For example, if the number of data is 64, then the data insertion is carried out on the 1st pixel up to the 64th pixel. If

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the amount of data is 128, then data insertion is carried out on the 1st pixel up to the 128th pixel and so on. The pixels are converted to 8-bit binary for the embedding process. Extraction is carried out by taking binary at the 1st position of the LSB of the image pixels from each pixel shown in Figure 6. The stego image in the extraction process will take the data that has been inserted and the decryption process is carried out with the key that has been created. Then, stego image will be data and cover image. The encrypted data in steganography is divided into 3 groups shown in Figure 7. Each group of encrypted data consists of 22 binary data bits. Group 1 is the 1st encrypted data to the 22nd encrypted data. Group 2 is the 23rd to 44th encrypted data. Group 3 from the 3rd to the 22nd data is the 43rd encrypted data to the 64th encrypted data. The 1st to 2nd data in group 3 are filled with data value of 0.

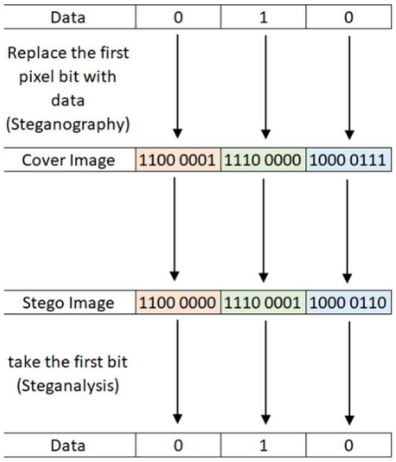


Figure 6 : Illustration Of The Steganography And Steganalysis Process

Steganalysis is carried out by taking binary at the 1st position of the LSB of the image pixels from each pixel. The results of the extraction in steganalysis are grouped first according to the color of the extracted pixels shown in Figure 8. The grouping process is carried out, namely: the 1st red pixel is placed in the R extract data group at position 22 (far right), the 1st green pixel is placed in the G extract data group at 22 position, the 1st blue pixel is placed in the data group. extract data B at position 22, the 2nd red pixel is placed in the data group R extract data at position 21, the 2nd green pixel is placed in the data group data extract G at position 21, the 2nd blue pixel is placed in the data extract data group B at position 21 and so on.

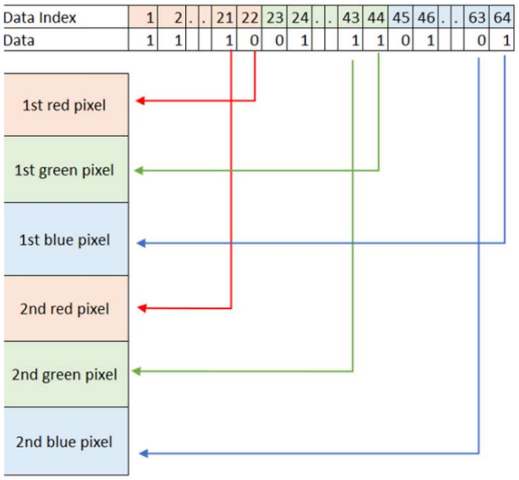


Figure 7 : Illustration of the insertion laying process

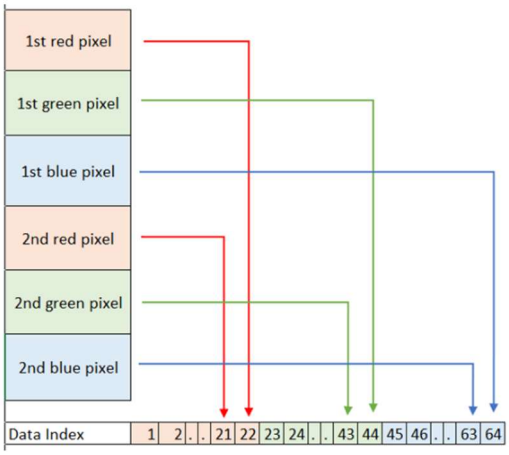


Figure 8 : Illustration Of The Extraction Laying Process

4. RESULTS AND DISCUSSION

The input data used in Xilinx was {0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 0 1 1 1 0}. Clock 1 was the beginning of the process, so the data value was '0'. Clock 2 was the beginning of incoming data, namely binary data. The beginning of the data was marked by an increase in the 'we' signal. Binary data was entered up to Clock 8. The data was transposed at clock 9 and the result was {0 1 1 0 0 1 1 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 0 10 1 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 0 1 1 0 0 1 0 1 1 0}. The results of the encryption were then pasted into pixels. Figure 9 was the result of the integration of transposition encryption and steganography. Clock 10 to clock 24 had input data binary pixels of red, green, and blue cover images. The results of the stego image were also issued at clock 3 to clock 31.

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

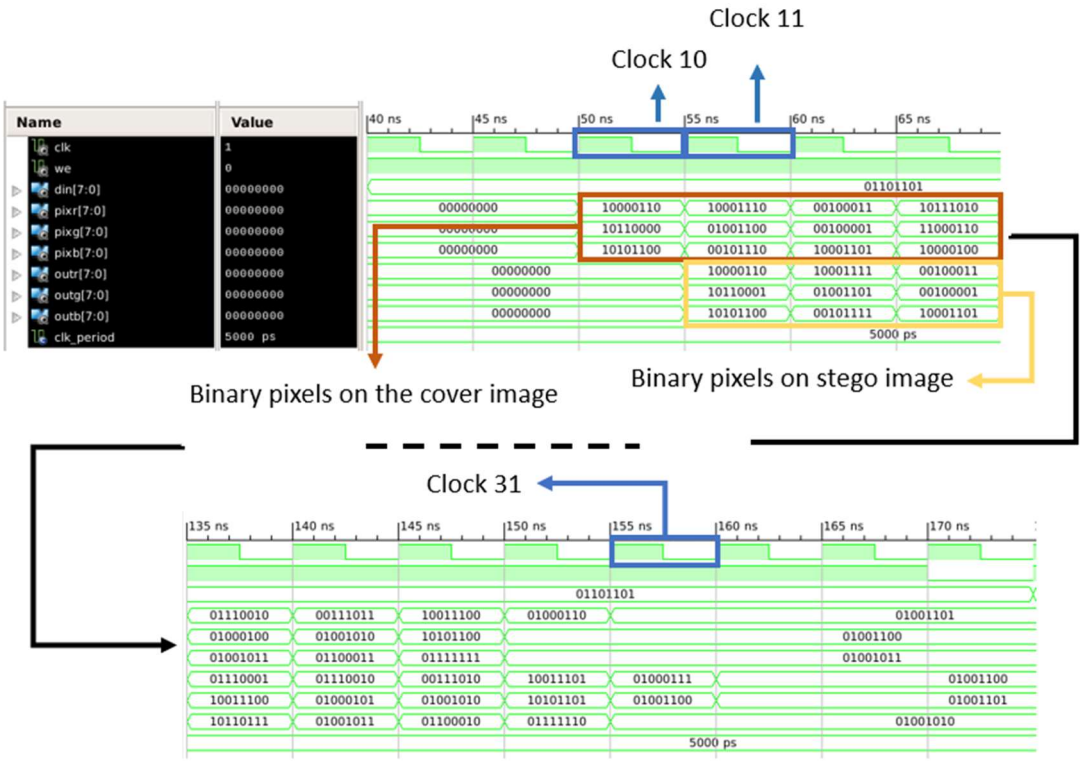
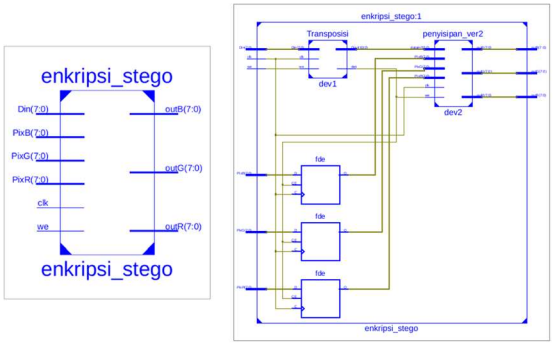


Figure 9 : The Results Of The Integration Of Transposition Encryption And Steganography On ISIM

Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	247	126,800	1%
Number used as Flip Flops	246		
Number used as Latches	1		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	107	63,400	1%
Number used as logic	38	63,400	1%
Number using O6 output only	31		
Number using O5 output only	0		
Number using O5 and O6	7		
Number used as ROM	0		
Number used as Memory	0	19,000	0%
Number used exclusively as route-thrus	69		
Number with same-slice register load	69		
Number with same-slice carry load	0		
Number with other load	0		
Number of occupied Slices	46	15,850	1%
Number of bonded IOBs	55	210	26%

Figure 10 : Device Summary Integration Of Transposition Encryption And Steganography

Figure 10 was a device summary of the integration that had been done. The integration of transposition encryption and steganography used 46 occupied slices, and 107 LUTs. The integration process of transposition encryption and steganography used 10 latencies and 31 cycles. Figure 11a and Figure 11b were an RTL 1 and RTL 2 schematic integration of transposition encryption and steganography on Xilinx. Figure 12 was a timing summary on integration. The integration of transposition encryption and steganography took 1,821 ns or worked at a maximum frequency of 450,572 MHz.



(a) (b)

Figure 11 : (A) RTL 1 Schematic Integration Of Transposition Encryption And Steganography On Xilinx, (B) RTL 2 Schematic Integration Of Transposition Encryption And Steganography On Xilinx

```
Speed Grade: -3

   Minimum period: 2.219ns (Maximum Frequency: 450.572MHz)
   Minimum input arrival time before clock: 1.181ns
   Maximum output required time after clock: 0.640ns
   Maximum combinational path delay: No path found
```

Figure 12 : Timing Summary Of Transposition Encryption And Steganography Integration

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

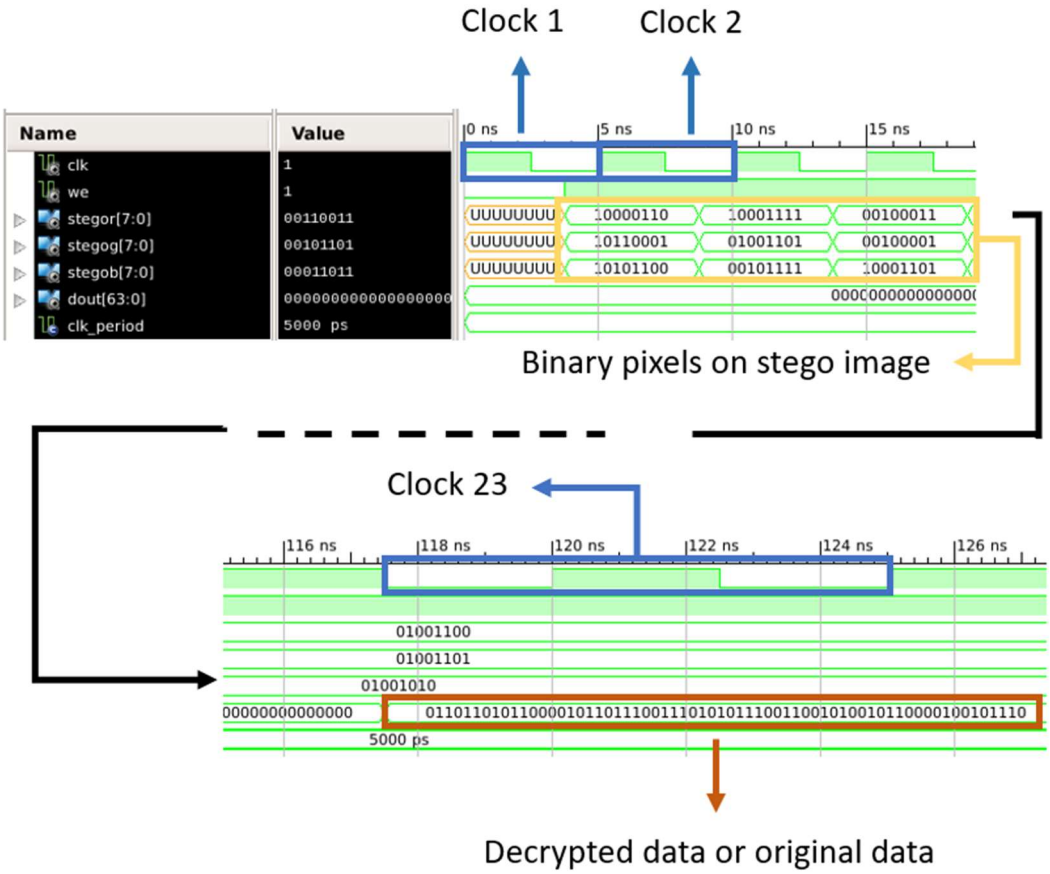


Figure 13 : The Results Of The Integration Of Transposition Decryption And Steganalysis On Isim

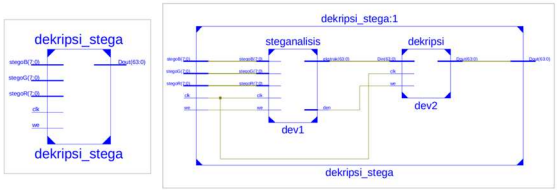
Figure 13 was the result of the integration of decryption and steganalysis on Isim. The results on 'dout' were the same as the original data on the integration of encryption and steganography. Clock 2 to clock 22 had input data of binary pixels of red, green and blue stego images. The results of the decrypted data were issued at clock 22. After the stego image data entered, the next clock had carried out the steganalysis process so that at clock 22 the steganization process was completed. Figure 14 was a device summary on the integration of transposition decryption and steganalysis. The integration used 77 occupied slices, and 108 LUTs. Figure 15a and Figure 15b showed the RTL 1 and RTL 2 schematic on the integration of transposition decryption and steganalysis. Figure 16 was a timing summary on integration. Integration took 2,172 ns or could work at a maximum frequency of 517,411MHz. The integration process used 22 latencies and 22 cycles.

Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	202	126,800	1%
Number used as Flip Flops	202		
Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	108	63,400	1%
Number used as logic	104	63,400	1%
Number using O6 output only	98		
Number using O5 output only	0		
Number using O5 and O6	6		
Number used as ROM	0		
Number used as Memory	0	19,000	0%
Number used exclusively as route-thrus	4		
Number with same-slice register load	4		
Number with same-slice carry load	0		
Number with other load	0		
Number of occupied Slices	77	15,850	1%
Number of bonded IOBs	70	210	33%

Figure 14 : Device Summary Integration Of Decryption And Steganalysis

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a) (b)

Figure 15 : (a) RTL 1 schematic on the integration of transposition decryption and steganalysis, (b) RTL 2 schematic on the integration of transposition decryption and steganalysis

```
Speed Grade: -3

   Minimum period: 1.933ns (Maximum Frequency: 517.411MHz)
   Minimum input arrival time before clock: 1.532ns
   Maximum output required time after clock: 0.640ns
   Maximum combinational path delay: No path found
```

Figure 16 : Timing summary of integration of decryption and steganalysis

The experiment on the FPGA board was carried out with the Nexys A7-100T series board. The test was carried out by utilizing the 7-Segment contained in the Nexys A7-100T FPGA as the output of the encryption and decryption results. The experiments carried out display the results of mapping transposition encryption and mapping transposition decryption. The representation of 7-Segment on FPGA hardware was a hexadecimal number. The concept of steganographic encryption and decryption steganalysis testing was shown in Figure 17 and Figure 18.

The data used in the steganographic encryption test was "4321uyab". The key used was the same as in the previous trial, namely the zig-zag pattern lock. Figure 19a showed the results of the steganographic encryption test on the Nexys A7-100T FPGA hardware. The encryption result obtained was "2E85836AC862789E". Figure 19b showed the results of the decryption steganalysis test on the Nexys A7-100T FPGA hardware. The decryption result obtained was "3433323175796162" or if it was converted back to text, it was "4321uyab" (same as the original data).

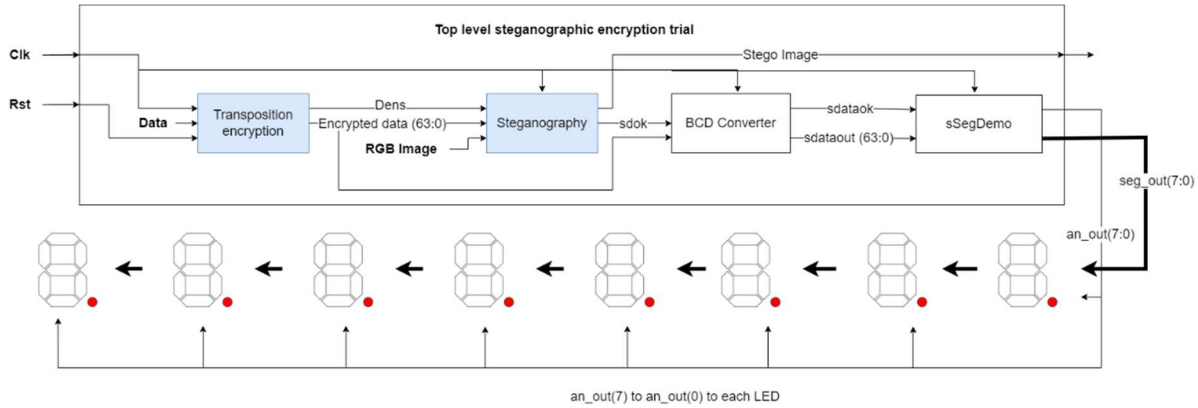


Figure 17 : The Concept Of Testing Of Encryption Steganography

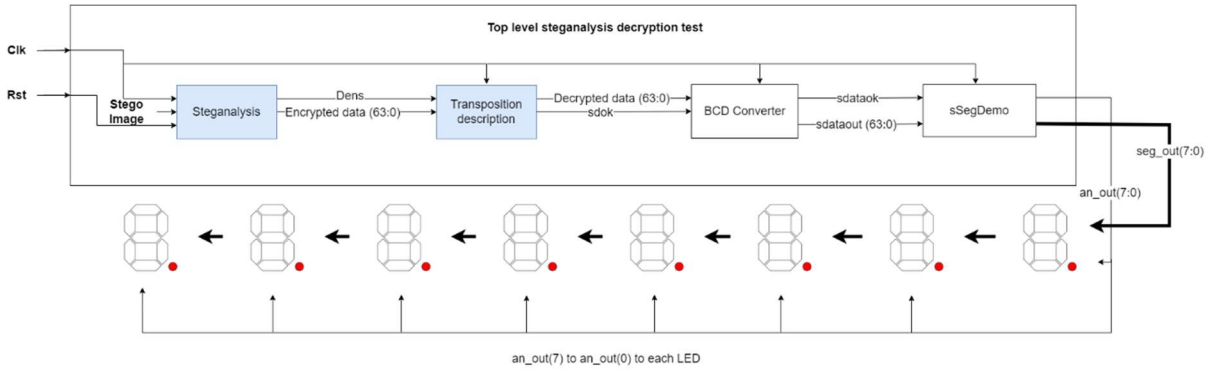
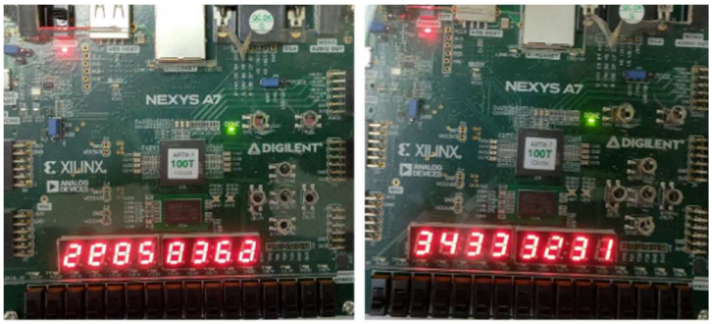


Figure 18 : The Concept Of Testing Of Decryption Steganalysis

The number of binary data bits that could be inserted into the cover image was the total pixels of the cover image. For example, if the cover image had a size of 256x256 so that the number of binary data bits that could be inserted was 16,777,216. The smallest image size that could be used was an image with a total of 64 pixels because the number of keys in this study was 64.



(a) (b)

Figure 19 : Test Results (A) Transposition Encryption On The FPGA Board And (B) Transposition Decryption On The FPGA Board

The Chaudhari et al. method, the Huang et al. method and the Pathak et al. method only implement LSB steganography so that the performance that was compared was only the steganographic method. The experiment conducted by Chaudhari used an image with a size of 512*512 RGB and the amount of data was not presented. The study was also compared with the same device using the Huang method and the Pathak method for steganographic resources. The experiment conducted by Huang used an image with a size of 256*256 grayscale and the amount of data inserted was 32,768 and 524,288, respectively. The experiment conducted by Pathak used an image with a size of 128*128 grayscale and the amount of data inserted was 16,384. The steganographic method obtained in this study uses the same device as the previous researcher. The resources used are smaller than the Huang method (270 slices and 340 LUTs with Spartan 3E devices) and Pathak methods (160 slices and 322 LUTs with Spartan 6 series XC6SLX45T devices).

Integration of transpose encryption and steganography requires 46 slices and 107 LUTs, and integration of steganalysis and transpose decryption requires 77 slices and 108 LUTs. The total slice of this integrated method is smaller than the Chaudhari method (2,411 slices), the Huang method (270 slices), and the Pathak method (160 slices). Table 2 shows a performance comparison between the different device methods used in the previous study and the results of this study.

Table 2:Resource Comparison

No	Name	Device	Slice	LUT
1	Chaudhari and Bhatt method [13]	N/A	2411	N/A
2	C.-W. Huang, C. Chou, Y.-C. Chiu and c. Chang method [15]	Spartan 3E	270	340
3	K. Pathak and M. Bansal method [14]	Spartan 6 seri XC6SLX45T	160	322
4	Proposed method	Spartan 3E	69	56
		Spartan 6 seri XC6SLX45T	30	39

5. CONCLUSIONS

The component (IPCore) of transposition and steganography process were obtained and has been optimized. These results were optimized because of 1 bit substitution in insertion and key stored in a table as mapping index. The transposition encryption and steganography process took 1,821 ns and the FPGA resource needed was 46 occupied slices and 107 LUTs. The transposition steganalysis and decryption process took 2,172 ns and FPGA resources needed was 56 occupied slices and 103 LUTs. These results showed that this research objective was fulfilled which were used smaller resources compared to other methods.

ACKNOWLEDGMENT

This work is supported by Research Program and the Kemendikbudristek Indonesia in Hibah Penelitian Disertasi Contract Number: 064/SP2H/LT/DRPM/2021 dated 18 March 2021 and Yayasan Pendidikan Gunadarma Jakarta Indonesia, Contract Number: 05A.24/LP/UG/IV/2021, 5 April 2021.

REFERENCES:

[1] W. Duncan, "Ontological distinctions between hardware and software", *Applied Ontology*, vol. 12, pp. 1-28, 2 2017.

[2] L. Zhao and D. Lie, "Is Hardware More Secure Than Software?", *IEEE Security & Privacy*, vol. 18, no. 5, pp. 8-17, Sept-Oct 2020.

[3] M. Annalakshmi and P. A. Padmapriya, "Zigzag Ciphers: A Novel Transposition Method", in *IJCA Proceedings on International Conference on Computing and Information Technology*, IC2IT(2):8-12, December 2013.

[4] P. Poonia and P. Kantha, "Comparative Study of Various Substitution and Transposition Encryption Techniques", *International Journal of Computer Applications* (0975 – 8887), vol. 145, no. 10, pp. 24-27, July 2016.

[5] K. Devi and G. Harshini, "Analysis and Comparison of Substitution and Transposition Cipher", *IJRAR- International Journal of Research and Analytical Reviews*, vol. 6, no. 2, pp. 549-555, June 2019.

[6] S. Godara, S. Kundu and R. Kaler, "An Improved Algorithmic Implementation of Rail Fence Cipher", *International Journal of Future Generation Communication and Networking*, vol. 11, no. 2, pp. 23-32, 12 March 2018.

[7] Saiful Islam, "Information Encryption By Zigzag Rule With Dynamic Block And Key", Master Thesis, Dhaka University of Engineering & Technology, Gazipur, Gazipur, 2011.

[8] A. P. U. Siahaan, "Rail Fence Cryptography in Securing Information", *International Journal of Scientific & Engineering Research*, vol. 7, no. 7, pp. 535-538, July 2016.

[9] M. A. Budiman, Amalia and N. I. Chayanie, "An Implementation of RC4+ Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security", in *2nd International Conference on Computing and Applied Informatics*, Medan, Indonesia, 2018.

[10] O. Fouad, A. A. M. Khalaf, A. Hussein and H. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques", *IEEE Access*, vol. 9, pp. 31805-31815, 2021.

[11] A. A. J. Altaay, S. B. Sahib and M. Zamani, "An Introduction to Image Steganography Techniques", *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pp. 122-126, 2012.

[12] N. Subramanian, O. Elharrouss, S. Al-ma'adeed and A. Bouridane, "Image Steganography: A Review of the Recent Advances", *IEEE Access*, vol. 1, pp. 23409-23423, 2021.

[13] J. Chaudhari and K. R. Bhatt, "FPGA Implementation of Image Steganography: A Retrospective", *International Journal of Engineering Development and Research*, vol. 2, no. 2, pp. 2117-2121, 2014.

[14] K. Pathak and M. Bansal, "A FPGA based Steganographic System Implementing a Modern Steganalysis Resistant LSB Algorithm", Defence Science *Journal*, vol. 67, no. 5, pp. 551-558, September 2017.

[15] C.-W. Huang, C. Chou, Y.-C. Chiu and c. Chang, "Embedded FPGA Design for Optimal Pixel Adjustment Process of Image Steganography", *Mathematical Problems in Engineering*, vol. 2018, pp. 1-8, 3 2018.

[16] E. Ordonez, "Platforms and Applications in Hardware Security: Trends and Challenges", *International Journal of Information and Computer Security*, vol. 7, pp. 289-304, 1 10 2013.

[17] G. Krishna and S. Roy, "Fundamentals of FPGA Architecture" in *Advanced Engineering*, Singapore, Technical and Scientific Publisher, 2017, pp. 12-30.

[18] S. M. Qasim, S. A. Abbasi and B. Almashary, "A review of FPGA-based design methodology and optimization techniques for efficient hardware realization of computation intensive algorithms", in 2009 *International Multimedia, Signal Processing and Communication Technologies*, Aligarh, India, 2009.

[19] S. Baddap, K. Khomane, P. Deshmukh and P. P. Shilpa, "Hardware Implementation of LSB Steganography for Data Security",

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 2, no. 3, pp. 59-63, May 2015.

[20] R. Candra, S. Madenda, S. A. Sudiro and M. Subali, "The Implementation of an Efficient Zigzag Scan", *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2, pp. 95-98, April 2017.

[21] S. Madenda, "Pengolahan Citra dan Video Digital Teori, Aplikasi dan pemrograman menggunakan MATLAB (Digital Image and Video Processing Theory, Application and programming using MATLAB)", *Jakarta: Erlangga*, ISBN: 9786022985983, 2015.