FPGA Implementation and Design of a Hybrid Chaos-AES Color Image Encryption Algorithm

Cheng-Hsiung Yang 1,2,* and Yu-Sheng Chien 1

1 Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; M10612913@mail.ntust.edu.tw
2 Center of Automation and Control, National Taiwan University of Science and Technology, 43 Section 4, Keelung Road, Taipei 10607, Taiwan
* Correspondence: yangch@mail.ntust.edu.tw

Received: 20 December 2019; Accepted: 13 January 2020; Published: 22 January 2020

Abstracts: In this paper, we propose an image encryption algorithm based on four-dimensional chaotic system to generate key and improve advanced encryption standard. The encryption algorithm is optimized by using the pipeline and parallel computing features of Field Programmable Gate Array (FPGA). First, the chaotic system is used as a key generator for the encryption algorithm. Next, in the improved advanced encryption standard, ShiftRows and SubBytes are modified with Spin-Sort and Cubic S-Box, and the round of encryption is reduced. We implement the encryption algorithm and the wired image transmission system to the ARM-based SoC-FPGA. The HPS software runs on Linux and is used to control the FPGA encryption algorithm and image transmission. Finally, the results from the encryption security analysis show that the proposed image encryption algorithm is safe and effective.

Keywords: four-dimensional chaotic system; S-Box; sort; image encryption; FPGA

1. Introduction

Recently, with the rapid development of network communication technology, the opportunities for global users to access the Internet have become more and more popular, and the dissemination of information and data has become more frequent. Multimedia and visual content already exist and are widely used in many domains, such as sharing military and medical personal information. Encryption methods are not only used in image multimedia, but also important for user security information (such as credit card numbers), cloud usage, etc. A good encryption scheme to protect information transmission security is of great significance. Therefore, encryption technology is receiving more and more attention. The image encryption scheme based on chaotic system has been widely studied, and its inherent characteristics such as initial sensitivity, unpredictability, and pseudo-randomness [1-3], thus observing the close relationship between chaotic systems and cryptography.

In 1989, Matthews first proposed an encryption scheme based on chaotic systems [4]. The first paper devoted to image encryption was published by Friedrich in 1998 [5]. A random number generator is used to generate a sequence of random numbers for encryption. When the generated number is more random, the encryption effect is better. Chaotic-based random number generator design is one of the common areas of chaotic systems used for encryption [6,7]. A permutation image encryption algorithm based on Logistic mapping and image segmentation is proposed in [8], in which the initial values and parameters of the Logistic map are used as secret keys. The diffusion and confusion methods is proposed by Shannon in [9]. Confusion causes the position of the pixel to be randomly disturbed, Cao et al [10] using breadth-first search as the confusion algorithm. The
confusion process has good encryption effect, but the pixel values do not change. It causes the histogram of the encrypted image to be similar to the original image. Therefore, its security is threatened by statistical analysis. All pixels have modified values during the diffusion process. The diffusion may result in higher security than confusion, but the encryption effect is not good. After these two operations, the normal image cannot be recognized correctly. S-Box performs replacement, which is one of the most important parts of the block encryption algorithm. Therefore, because they bring nonlinearity to the encryption process, using a powerful S-Box structure helps to achieve secure encryption.

The main reasons for the destruction of the image cryptosystem are: (1) the encryption key stream is only related to the secret key, but unrelated to the plain images; (2) the encryption method itself has a defect, so that partial information of cipher images is easily cracked and causes the entire encryption scheme to fail. In order to resist the choice of plaintext and known plaintext attacks, [11,12] adopts the method of generating key stream according to plaintext, that is, different original images will produce completely different key streams. High-dimensional chaotic maps have more variables and parameters. It usually exhibits good hyper-chaos properties and is more suitable for encryption. Therefore, we propose a chaotic encryption algorithm, combined with high-dimensional chaotic mapping and improved confusion and diffusion of AES, implemented on SoC FPGA. To avoid the defect of Shannon's entropy. In [13,14], another scheme is proposed to verify the randomness of the image, called “Shannon's local entropy.” We also use Shannon local entropy analysis to validate our proposed scheme [15–28].

The rest of the paper is organized as follows. Section 2 presents our designed four-dimensional chaotic system and proposed image encryption scheme. Section 3 explains how to implement the wire image transmission encryption system on the FPGA. The security analysis and evaluation are presented in Section 4. Finally, in Section 5 some conclusions are drawn.

2. Algorithm Design

In this section, we introduce a newly designed four-dimensional chaotic system. First, the characteristics of chaotic systems using various techniques including phase portrait and Lyapunov exponent are verified. Second, an improved chaotic encryption algorithm is proposed based on advanced encryption standard (AES) and key generator. Finally, the encryption scheme of this paper is represented by a flow chart.

2.1. Chaotic System

The three-dimensional chaotic Rössler system [18] is used for image encryption. It can be described as Equation (1).

$$\begin{cases}
\dot{x} = -y - z \\
\dot{y} = x + ay \\
\dot{z} = b + z(x-c)
\end{cases}$$

(1)

Base on Rössler system, we design a 4D chaotic dynamic system as Equation (2).

$$\begin{cases}
\dot{x} = -y - z + dw \\
\dot{y} = x + ay \\
\dot{z} = b + z(x-c) - a(y-w) \\
\dot{w} = az - w
\end{cases}$$

(2)

where the $x$, $y$, $z$, and $w$ are the states and $a$, $b$, $c$, and $d$ are parameters of the system.

To use the fourth-order Runge–Kutta method in MATLAB, the parameters are chosen as $a = 0.4$, $b = 0.6$, $c = 3$, and $d = 0.8$ and designed the initial values $x$, $y$, $z$, $w = [1, 1, 1, 1]$. We obtained phase portraits of the new 4D chaotic system. Phase portrait is one of the direct observation methods to
determine chaotic system. The trajectory of chaotic motion is a never-closing curve. The reciprocating movement in the orbit curve is confined to a bounded area and forms a strange attractor [20]. The Figure 1 shows the two-dimensional phase portraits of the new 4D chaotic system.

![Figure 1. Chaos system x-y (a) and x-z (b) dimension phase portrait.](image)

Figure 2 shows the diagrams of Lyapunov exponents by changing the parameter $d$ of the new chaotic system between 0 and 1. When $\lambda < 0$, the orbit is attracted to a stable periodic motion or fixed point, and the system is dissipative. If $\lambda = 0$ is called Lyapunov stability, its orbit is a neutral fixed point. It means that the system is conservative. If $\lambda > 0$, the system is sensitive to the initial value. It means that the system is chaotic and unstable. Figure 2 exhibits that the new chaotic system has chaotic characteristic values between 0 and 1.

![Figure 2. D chaotic system with parameter $d$.](image)

2.2. Purpose of New Chaotic Encryption Algorithm

AES is a well-known block cipher that has several advantages in data encryption. It is a fast encryption scheme and AES algorithm more robust against hacking. For 128-bit key, about $2^{128}$ attempts are needed to crack. However, it is not suitable in multimedia data because multimedia data has high redundancy. Only using AES encrypted images cannot achieve good encryption. In order to solve the shortcomings of AES encrypted images, we combine the features of chaotic systems, such as initial value sensitivity, ergodicity and generation of pseudo-random numbers, and modification of confusion and diffusion in the AES. It enhances the complexity of encryption and greatly increases security.

Each round of AES encryption (except for the last round) consists of four steps [16].
i. AddRoundKey: Each byte in the matrix uses XOR to manipulate the round key; a subkey is derived from the main key using Rijndael’s key schedule.

ii. SubBytes: Change each byte using an 8-bit substitution box. This step provides the nonlinear substitution of the replacement function.

iii. ShiftRows: In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The shifted number of places each byte is depended on the number of rows.

iv. MixColumns: In order to properly mix linear operations in a matrix, this step uses a linear transformation to mix four bytes per line. The last encryption loop omits the MixColumns step and replaces it with another AddRoundKey.

In Ref. [11], it presents Cubic S-Box for diffusion. We use Cubic S-Box to replace AES SubBytes. The Figure 3 show the block of AES. Stack block I, II, III, and IV as z axis. It enhances the original S-box become a three-dimensional matrix named by Cubic S-box as shown in Figure 4.

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63</td>
<td>7c</td>
<td>77</td>
<td>7b</td>
<td>f2</td>
<td>6b</td>
<td>6f</td>
<td>c5</td>
<td>30</td>
<td>01</td>
<td>67</td>
<td>2b</td>
<td>fe</td>
<td>d7</td>
<td>ab</td>
</tr>
<tr>
<td>1</td>
<td>ca</td>
<td>82</td>
<td>c9</td>
<td>7d</td>
<td>fa</td>
<td>59</td>
<td>47</td>
<td>f0</td>
<td>ad</td>
<td>d4</td>
<td>a2</td>
<td>af</td>
<td>9c</td>
<td>a4</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>b7</td>
<td>fd</td>
<td>93</td>
<td>26</td>
<td>36</td>
<td>3f</td>
<td>f7</td>
<td>cc</td>
<td>34</td>
<td>a5</td>
<td>e5</td>
<td>f1</td>
<td>71</td>
<td>d8</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>04</td>
<td>c7</td>
<td>23</td>
<td>c3</td>
<td>18</td>
<td>96</td>
<td>05</td>
<td>9a</td>
<td>07</td>
<td>12</td>
<td>80</td>
<td>c2</td>
<td>eb</td>
<td>27</td>
<td>b2</td>
</tr>
<tr>
<td>4</td>
<td>09</td>
<td>83</td>
<td>2c</td>
<td>1a</td>
<td>1b</td>
<td>6e</td>
<td>5a</td>
<td>a0</td>
<td>52</td>
<td>3b</td>
<td>d6</td>
<td>b3</td>
<td>29</td>
<td>e3</td>
<td>c3</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>d1</td>
<td>00</td>
<td>ed</td>
<td>20</td>
<td>fc</td>
<td>b1</td>
<td>5b</td>
<td>6a</td>
<td>cb</td>
<td>be</td>
<td>39</td>
<td>4a</td>
<td>4c</td>
<td>58</td>
</tr>
<tr>
<td>6</td>
<td>d0</td>
<td>ef</td>
<td>aa</td>
<td>fb</td>
<td>43</td>
<td>4d</td>
<td>33</td>
<td>85</td>
<td>45</td>
<td>f9</td>
<td>02</td>
<td>7f</td>
<td>50</td>
<td>3c</td>
<td>9f</td>
</tr>
<tr>
<td>7</td>
<td>51</td>
<td>a3</td>
<td>40</td>
<td>8f</td>
<td>92</td>
<td>9d</td>
<td>38</td>
<td>f5</td>
<td>bc</td>
<td>b6</td>
<td>da</td>
<td>21</td>
<td>10</td>
<td>ff</td>
<td>f3</td>
</tr>
<tr>
<td>8</td>
<td>ed</td>
<td>0c</td>
<td>13</td>
<td>ec</td>
<td>5f</td>
<td>97</td>
<td>44</td>
<td>17</td>
<td>c4</td>
<td>a7</td>
<td>7e</td>
<td>3d</td>
<td>64</td>
<td>5d</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>60</td>
<td>81</td>
<td>4f</td>
<td>dc</td>
<td>22</td>
<td>2a</td>
<td>90</td>
<td>88</td>
<td>46</td>
<td>ec</td>
<td>b8</td>
<td>14</td>
<td>de</td>
<td>5e</td>
<td>0b</td>
</tr>
<tr>
<td>a</td>
<td>e0</td>
<td>32</td>
<td>3a</td>
<td>0a</td>
<td>49</td>
<td>06</td>
<td>24</td>
<td>5c</td>
<td>c2</td>
<td>d3</td>
<td>ac</td>
<td>62</td>
<td>91</td>
<td>95</td>
<td>e4</td>
</tr>
<tr>
<td>b</td>
<td>c7</td>
<td>e8</td>
<td>37</td>
<td>6d</td>
<td>8d</td>
<td>d5</td>
<td>4e</td>
<td>a9</td>
<td>6c</td>
<td>56</td>
<td>f4</td>
<td>ea</td>
<td>65</td>
<td>7a</td>
<td>ae</td>
</tr>
<tr>
<td>c</td>
<td>ba</td>
<td>78</td>
<td>25</td>
<td>2c</td>
<td>1e</td>
<td>a6</td>
<td>b4</td>
<td>c6</td>
<td>e8</td>
<td>dd</td>
<td>74</td>
<td>1f</td>
<td>4b</td>
<td>bd</td>
<td>8b</td>
</tr>
<tr>
<td>d</td>
<td>70</td>
<td>3e</td>
<td>b5</td>
<td>66</td>
<td>48</td>
<td>03</td>
<td>f6</td>
<td>0e</td>
<td>61</td>
<td>35</td>
<td>57</td>
<td>b9</td>
<td>86</td>
<td>c1</td>
<td>1d</td>
</tr>
<tr>
<td>e</td>
<td>e1</td>
<td>f8</td>
<td>98</td>
<td>11</td>
<td>69</td>
<td>d9</td>
<td>8e</td>
<td>94</td>
<td>9b</td>
<td>1e</td>
<td>87</td>
<td>e9</td>
<td>ce</td>
<td>55</td>
<td>28</td>
</tr>
<tr>
<td>f</td>
<td>8c</td>
<td>a1</td>
<td>89</td>
<td>0d</td>
<td>bf</td>
<td>c6</td>
<td>42</td>
<td>68</td>
<td>41</td>
<td>99</td>
<td>2d</td>
<td>0f</td>
<td>b0</td>
<td>54</td>
<td>bb</td>
</tr>
</tbody>
</table>

**Figure 3.** Block of S-box.

**Figure 4.** Cubic S-box.
For a given 1-byte data \( d \), transform \( d \) to binary form 
"\( d_7d_6d_5d_4d_3d_2d_1d_0 \)" where 
\( d_i \in \{0,1\}, \quad (i=0,1,...,7) \). Any 1-byte value is represented by hexadecimal value "\( xyz \)," where 
\( x = d_7d_6d_5, \quad y = d_4d_3d_2, \) and 
\( z = d_1d_0 \). Substitute values by querying columns, rows, and depth in Cubic S-box. For example, input 
\( d = (58)_{16} \), transfer to binary \( d = (00111010)_{2} \). Then, \( x = (001)_2, \quad y = (110)_2, \) and \( z = (10)_2 \). Substituted values can be obtained by third block, second row and seventh column. The output \( d' = (90)_{16} \). There are six different options for express 1-byte date \( d \). The example \( d = "xyz" \) is one option. We also can express 
\( d = "xzy," "yxz," "zxy," "zxy," and "zyx." \) Compared with S-box, Cubic S-box is more suitable to generate two different numbers from one 1-byte data. The security of the cryptosystem can be enhanced. We use two types of "\( yxz \)" and "\( zyx. \)" The input 128-bit chaotic sequence key is used as a parameter to select the type.

Since ShiftRows of AES has only one form of diffusion. A 16-form of diffusion is proposed. It used to replace the ShiftRows of AES. Name by Spin-Sort. Its steps are as follows:

Step i. In the Spin-Sort, we can express the plaintext as a \( 4 \times 4 \) matrix in Figure 5a. Where \( x_0, x_1, x_2, \ldots, x_{15} \) are 8-bit values.

Step ii. Mark the matrix as an outer and inner blocks as Figure 5b.

Step iii. Take a vertex of the outer(inner) block as the root.

Step iv. Sorted by the root, clockwise (counterclockwise) from the outer (inner) block to the inner (outer) block.

According to step i to step iv, obtain 16 sequences as Equation (3). Figure 6 shows a sorting process of Spin-Sort.

Figure 5. (a) Plaintext; (b) mark inner and outer blocks.

Figure 6. Spin-Sort (a) start from outer-top-left vertex and clockwise; (b) start from outer-top-left vertex and counterclockwise; (c) start from inner-top-left vertex and clockwise; (d) start from inner-top-left vertex and counterclockwise.
Each 24-bit image file can be express as Figure 7a. A pixel includes 3 byte data, respectively R/G/B. To enhance the complexity of the encryption algorithm, before performing algorithm, add a diffusion method called diversion of pixel. All pixels of image into RGB channels. As shown in Figure 7b, reconstitute a new image.

![Figure 7](image.png)

Figure 7. (a) Pixel composition of the image. (b) Diversion of pixel.

2.3. Key Generator Design

The encryption key stream is only related to the secret key, but unrelated to the plain images. This is the main reason for destroying the image cryptosystem. To make the cryptosystem depend on the plain image, the encryption key uses the initial values \(x_0, y_0, z_0\), and \(w_0\) in the 4D chaotic system and features of image pixel. After a certain number of iterations, the initial 128-bit key is generated. Every block encryption finish will iterate again to generate a new encryption key. Therefore, the
encryption key generated by the 4D chaotic system is not only related to the initial value, but also related to the plaintext. That means different plaintexts will produce completely different key streams. The following is the process of key generator:

Step. i. Select the proper values to be the initial conditions for the 4D chaotic system.

\[
\text{initial condition} = \{x_0, y_0, z_0, w_0\}
\]  

(4)

Step. ii. Execute pre-iteration \(n_r\) rounds, where \(n_r\) is 2000 rounds.

\[
\text{pre iterated condition} = \{x', y', z', w'\}
\]

(5)

Step. iii. Calculate summation of image pixel, according to Equation (6).

\[
\text{Sum}_{\text{pixel}} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \text{pixel}(i, j)\]

(6)

where \(m\) and \(n\) are the image dimension, \(\text{Pixel}(i, j)\) is the \(i\) row and \(j\) column pixel unsigned values of plain image.

Step. iv. Use \(\text{Sum}_{\text{pixel}}\) as parameter to iterate \(n_p\) rounds as in Equation (7).

\[
\text{n}_p = \left(\frac{256 \times \text{Sum}_{\text{pixel}}}{1000}\right)
\]

(7)

Step. v. The initial 128-bit key generate after iterate \(n_p\) rounds in 4D chaotic system as Equation (8).

\[
\text{initial 128bit key} = \{(\text{CHAOS}_X)_1, (\text{CHAOS}_Y)_2, (\text{CHAOS}_Z)_3, (\text{CHAOS}_W)_4, \}
\]

(8)

Step. vi. Iterate once every 16-byte block encryption.

2.4. Flow Chart of Encryption

We describe the entire encryption process via a flow chart. Figure 8 shows the flow chart of encryption algorithm. It contains the steps of AES, Spin-Sort, Cubic S-Box, Diversion of pixel, and uses the chaotic key generator. Standard AES uses nine rounds with an initial round and a final round. In this paper, four rounds are used, an initial round and a final round. The decryption algorithm is the reverse process of encryption algorithm.

![Figure 8. Flow chart of encryption in this proposal method.](image)
3. FPGA Implementation

In this section, the entire experiment is conducted in Altera DE10-standard. The DE10-Standard use CycloneV system-on-a-chip (SoC) is composed of two distinct portions a dual-core Arm Cortex-A9 hard processor system (HPS) and an FPGA. A LXDE Linux program in the SD card and running on the ARM processor. Next, describe how to implement Euler’s method chaotic system on the FPGA and hardware implementation of our encryption system. Wired image transmission system showed in Figure 9.

Figure 9. Wired image transmission system.

3.1. Chaos System Discretization

In order to implement the chaotic system on FPGA, we use Euler’s method to implement the discretization. The chaotic system of Equation (2) can be rewritten according to our system as shown in Equation (9).

\[
\begin{align*}
    x(n+1) &= x(n) + \Delta t[-y(n) - z(n) + dw(n)] \\
    y(n+1) &= y(n) + \Delta t[x(n) + ay(n)] \\
    z(n+1) &= z(n) + \Delta t[b + z(n)(x(n) - c) - a(y(n) - w(n))] \\
    w(n+1) &= w(n) + \Delta t[a z(n) - w(n)]
\end{align*}
\]

where \( a = 0.4, b = 0.6, c = 3, d = 0.8 \), the initial values \( x, y, z, w = [1, 1, 1, 1] \) and the step size \( \Delta t \) is 0.001.

According to Equation (9), our system contains addition, subtraction, and multiplication. We use the floating point IP-CORE ALTFP_ADD_SUB and ALTFP_MULT, provided by Altera, as shown in Table 1. The two IP-CORE follows the IEEE-754 Standard. The IP-CORE ALTFP_ADD_SUB can run add or subtract operation and ALTFP_MULT handles floating-point multiplication. There are different latency times in these two IP-CORE, ALTFP_ADD_SUB is 7 clocks and ALTFP_MULT is 5 clocks. We need to wait for these clocks to make sure the calculations are correct.

In hardware design, we split every step of the operation in Equation (9), first. Figure 10 shows the entire chaos system operations. Our chaotic system requires six steps to get the next iterations. Then we use the adder, subtractor, and multiplier in the IP-CORE of Altera’s floating-point arithmetic unit to implement Equation (9). The system’s clock for the entire architecture is 50MHz. Actually, we need to wait one more clock in each operation, since the system’s clock is a positive edge trigger. The latency time of ALTFP_ADD_SUB is 8 clocks, ALTFP_MULT is 6 clocks. It can be calculated that the operation time of entire architecture is 44 clocks.

For the entire hardware, the chaotic system is just one of the modules. It uses the signals of CHAOS_STEP, CHAOS_RESET, and CHAOS_DONE to control this module. The signal CHAOS_STEP is the start signal. When 44 clocks are passed, an iteration is completed to generate new values \( x(n+1), y(n+1), z(n+1) \) and \( w(n+1) \). At this time, the signal CHAOS_DONE triggers. The signal CHAOS_RESET is used to reset the chaotic system, and the iteration value will be reset back to initial values \( x_0, y_0, z_0, w_0 \).
Table 1. Time latency of floating point IP-CORE.

<table>
<thead>
<tr>
<th>IP-CORE</th>
<th>Time Latency (Clock, 50 MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTFP_ADD</td>
<td>8</td>
</tr>
<tr>
<td>ALTFP_SUB</td>
<td>8</td>
</tr>
<tr>
<td>ALTFP_MULT</td>
<td>6</td>
</tr>
</tbody>
</table>

Figure 10. Operation of chaotic system.

3.2. Encryption System Hardware Implementation

The Figure 11 shows the entire system of image encryption. It contains FPGA encryption, HPS software, SDRAM, VGA display, and wired transmission. There are two boards that are connected to a router. The router is not connected to the Internet. Use the router’s dynamic host configuration protocol (DHCP) to dynamically assign different IP addresses to the two boards. A computer and two DE10 boards form an internal network.

A. SDRAM access

In order to achieve our FPGA encryption system and VGA display, the image data need to be stored in a memory that HPS and FPGA can access. The FPGA 64MB SDRAM is a suitable memory.
The image data is relatively large, so we use HPS-to-FPGA Bridge to connect SDRAM and HPS. The HPS software uses mapped address of the SDRAM to transfer the image data into the SDRAM. Since the frame reader will read R/G/B/A 4 bytes at a time, it needs to fill in an empty byte for each pixel. The HPS uses the control signal of the frame reader to display image through VGA.

The SDRAM is a 16-bit data bus, it means only read 2 bytes at a time. The FPGA uses a 50 MHz clock but the SDRAM uses a 100 MHz clock. To synchronize data originating from different clock domains, the SDRAM controller uses two FIFOs (first in, first out) to store read data and write data respectively. Since the encryption system represents 16 byte data as a $4 \times 4$ matrix, the encryption process requires 16 bytes of data. We use the pipeline feature to read 2 bytes from FIFO at a time, stored in a 128-bit register until it meets 128 bits. Then trigger a control signal to start the encryption module. When encryption is complete, store data in another 128-bit register. The encrypted 128-bit data will also be written to the write FIFO in sequence.

B. Encryption system

In the previous section, we mentioned Cubic S-Box, Spin-Sort, and MixColumn, AddRoundKey of AES. These encryption methods require four rounds of execution. Using FPGA parallel computing architecture, four rounds can be completed in four clocks. The step of encryption system on FPGA design as follows:

Step. i.  Posedge clock, count a register $dcnt$ to confirm if the number of rounds is 4.
Step. ii. Posedge clock, module $aes\_key\_expand\_128$ generate new round key.
Step. iii. Posedge clock, check whether is initial round. If is initial round, registers $sam,n$ are divided 128-bits input data XOR first round key, Else, assign by $sam_n\_{next}$ which is $sam_n\_mc$ XOR round key,
Step. iv. Wire connected module $aes\_Csbox$, $spin\_sort$ and $mix\_col$. The module $aes\_Csbox$ produced $sam_n\_cs$, the module $spin\_sort$ produced $sam_n\_ss$, the module $mix\_col$ produced $sam_n\_mc$.
Step. v.  Posedge clock, the output data is assigned $sam_n\_ss$ XOR round key.

where $m$ is 0, 1, 2, 3 and $n$ is 0, 1, 2, 3. It is worth noting that all steps are performed simultaneously, instead of step by step. The Figure 12 shows the flow chart of encryption system.

![Diagram](image)

Figure 12. Flow chart of encryption system on FPGA design.
4. Security Analysis

In order to analyze our encryption performance, we use histogram analysis, correlation analysis, differential attack analysis, and information entropy analysis to verify the security of our proposed encryption algorithm. For test images, we chose 24-bits BMP that include Lena, Airplane, Pepper, and Mandrill.

A. Histogram Analysis

The histogram of the image reflects the distribution of pixel values of an image. A color image can be used to treat the three channels of red, green, and blue as a special gray image, and the one-dimensional histogram method of gray image is used to analyze the performance of the encryption algorithm. The ciphertext generated by a secure encryption algorithm should be nearly evenly distributed in the histogram, and has significant difference compared with the histogram of the original image. The Figure 13 is histogram test of Lena, and (a) and (b) are the RGB histograms of the plaintext and its ciphertext encrypted by our proposed chaotic encryption. It can be seen that the histogram of the encrypted image is nearly completely consistent and has a significant difference from the original image. So our chaotic encryption algorithm are safe in histogram analysis. Figures 14 and 15 shown the Correlation of each channel in plain image Lena and in cipher image Lena, respectively.

![Figure 13. Histogram analysis of Lena plaintext (a) and its ciphertext encrypted (b) by our proposed chaotic encryption.](image)
Correlation analysis is used to measure the relationship between two adjacent pixels in an image. The adjacent pixels of the original image have high correlation in the horizontal, vertical, and diagonal directions. In order to improve the resistance of statistical attacks, an ideal image encryption scheme should reduce the correlation between the two adjacent pixels [22]. The correlation coefficient can be calculated by
\[ r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \]  
\[ D(x) = \frac{\sum_{i=1}^{n} (x_i - E(x))^2}{n} \]  
\[ E(x) = \frac{\sum_{i=1}^{n} x_i}{n} \]  
\[ \text{cov}(x, y) = \frac{\sum_{i=1}^{n} (x_i - E(x))(y_i - E(y))}{n} \]  

where \( x \) and \( y \) are values of two adjacent pixels in the image, \( n \) is the number of pixel pairs, \( D(x) \) is the variance, \( E(x) \) is the mean and \( \text{cov}(x, y) \) is the covariance.

The correlation coefficient is shown in Tables 2–5. We can find that proposed encryption algorithm has a sufficiently safe performance.

<table>
<thead>
<tr>
<th>Lena</th>
<th>Airplane</th>
<th>Pepper</th>
<th>Mandrill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagonal</td>
<td>-0.000483</td>
<td>0.000348</td>
<td>0.000759</td>
</tr>
<tr>
<td>Horizontal</td>
<td>-0.001001</td>
<td>0.000468</td>
<td>0.000466</td>
</tr>
<tr>
<td>Vertical</td>
<td>-0.001015</td>
<td>0.001086</td>
<td>-0.000803</td>
</tr>
</tbody>
</table>

Table 3. Comparison of correlation coefficients on Lena (red).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagonal</td>
<td>-0.000546</td>
<td>0.000573</td>
<td>0.006000</td>
</tr>
<tr>
<td>Horizontal</td>
<td>-0.000383</td>
<td>0.000538</td>
<td>-0.012700</td>
</tr>
<tr>
<td>Vertical</td>
<td>-0.001287</td>
<td>-0.007058</td>
<td>0.006700</td>
</tr>
</tbody>
</table>

Table 4. Comparison of correlation coefficients on Lena (green).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagonal</td>
<td>-0.003829</td>
<td>-0.001693</td>
<td>-0.007800</td>
</tr>
<tr>
<td>Horizontal</td>
<td>0.001205</td>
<td>0.001186</td>
<td>-0.007500</td>
</tr>
<tr>
<td>Vertical</td>
<td>-0.000961</td>
<td>0.000177</td>
<td>-0.006800</td>
</tr>
</tbody>
</table>

Table 5. Comparison of correlation coefficients on Lena (blue).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagonal</td>
<td>0.002926</td>
<td>-0.000927</td>
<td>0.002600</td>
</tr>
<tr>
<td>Horizontal</td>
<td>-0.003825</td>
<td>-0.002372</td>
<td>-0.000700</td>
</tr>
<tr>
<td>Vertical</td>
<td>-0.000798</td>
<td>0.007818</td>
<td>0.004200</td>
</tr>
</tbody>
</table>

C. Differential Attack Analysis

A differential attack is usually a plaintext attack. The attacker encrypts with a small change in plaintext. Then the corresponding ciphertext is generated for statistical analysis to obtain the key. A secure encryption scheme should be very sensitive to plain images. To measure the impact of this
type of attack, pixel rate of change (NPCR) and uniform mean change intensity (UACI) are calculated as follows:

\[
D(i, j) = \begin{cases} 
0, & \text{if } C_1(i, j) = C_2(i, j) \\
1, & \text{if } C_1(i, j) \neq C_2(i, j)
\end{cases}
\]  

(114)

\[
NPCR = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\%
\]

(15)

where, \( C_1 \) is an encrypted image from original plain image and \( C_2 \) is an encrypted image from plain image with small change, i.e., one pixel difference. And \( i \) and \( j \) are the pixel positions, \( H \) and \( W \) are height and width of the ciphertext image.

\[
UACI = \frac{1}{W \times H} \left( \frac{\sum_{i,j} |C_1(i, j) - C_2(i, j)|}{255} \right) \times 100\%
\]

(16)

where \( C_1, C_2, i, j, W, \) and \( H \) are the same as in Equations (14) and (15).

The test result of the differential attacks analysis NPCR and UACI is shown in Tables 6 and 7. We can find that the NPCR score is close to 100%, and the UACI is close to the theoretical value of 33.33%. It means that the proposed encryption algorithm is sufficient to resist differential attacks.

Table 6. Comparative of NPCR scores.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>99.6040%</td>
<td>99.71%</td>
<td>99.6150%</td>
<td>99.6525%</td>
</tr>
<tr>
<td>Airplane</td>
<td>99.6051%</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Pepper</td>
<td>99.6227%</td>
<td>99.74%</td>
<td>99.6156%</td>
<td>99.6254%</td>
</tr>
<tr>
<td>Mandrill</td>
<td>99.6071%</td>
<td>99.72%</td>
<td>99.6031%</td>
<td>99.6378%</td>
</tr>
</tbody>
</table>

Table 7. Comparative of UACI scores.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>33.4614%</td>
<td>33.45%</td>
<td>33.4893%</td>
<td>33.4331%</td>
</tr>
<tr>
<td>Airplane</td>
<td>33.4503%</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Pepper</td>
<td>33.5029%</td>
<td>33.53%</td>
<td>33.4336%</td>
<td>33.4566%</td>
</tr>
<tr>
<td>Mandrill</td>
<td>33.4969%</td>
<td>33.50%</td>
<td>33.4392%</td>
<td>33.4673%</td>
</tr>
</tbody>
</table>

D. Information Entropy Analysis

Information entropy is a quantitative measure of randomness of a system, proposed by Shannon [9]. The global entropy \( H(m) \) of a message source \( m \) can be calculated as

\[
H(m) = -\sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)}
\]

(17)

where \( p(m) \) means the probability of each symbol appearance. For color images, each color channel(red, green, or blue) is described as 8-bits that have pixel values ranging from 0 to 255. The maximum value of entropy is 8. A good image encryption scheme, the global entropy value of the encrypted image should be close to 8. There are two disadvantages to the global entropy measurement. First, the global Shannon entropy fails to measure the true randomness of the image if the image contains some image blocks with low Shannon entropy. Second, the global information entropy may be affected by different image size, because larger images contain more information. To complement these disadvantages, use local Shannon entropy measures image randomness. The local entropy can be calculated as
where \( k \) is non-overlapping image blocks and \( H(S) \) is the information entropy of block \( i \).

Referring to [15,16], we randomly select 30 non-overlapping \( 44 \times 44 \) image blocks to calculate the average of information entropy. If the average information entropy (called local information entropy) is greater than 7.9, the image randomness is high. The encryption algorithm has sufficient security. Thus, Tables 8 and 9 show the global and local Shannon entropy of the encrypted images. The results of global entropy are close to the theoretical value 8 and the results of local entropy is higher than 7.9. This means that the image encrypted by the proposed encryption scheme has good randomness.

### Table 8. Comparative of global entropy of propose scheme.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>7.99932</td>
<td>7.99729</td>
<td>7.99717</td>
<td>7.99729</td>
</tr>
<tr>
<td>Airplane</td>
<td>7.99928</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>Pepper</td>
<td>7.99928</td>
<td>NaN</td>
<td>7.99707</td>
<td>7.99733</td>
</tr>
<tr>
<td>Mandrill</td>
<td>7.99932</td>
<td>7.99897</td>
<td>7.99700</td>
<td>7.99720</td>
</tr>
</tbody>
</table>

### Table 9. Comparative of local entropy of propose scheme.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>7.90299</td>
<td>7.90210</td>
<td>7.90237</td>
<td>7.90210</td>
</tr>
<tr>
<td>Airplane</td>
<td>7.90285</td>
<td>NaN</td>
<td>7.90212</td>
<td>NaN</td>
</tr>
<tr>
<td>Pepper</td>
<td>7.90255</td>
<td>7.90240</td>
<td>7.90275</td>
<td>7.90240</td>
</tr>
<tr>
<td>Mandrill</td>
<td>7.90244</td>
<td>7.90190</td>
<td>7.90271</td>
<td>7.90230</td>
</tr>
</tbody>
</table>

### 5. Conclusions

In the paper, we design a new chaotic system based on the Rössler system and implemented on the Altera FPGA DE10-standard board. Using the Euler method with the given value as the initial condition and the sum of the image pixels as the iteration parameter, four pseudo-random numbers are generated as the encryption key. Each time a 16-bytes block is encrypted, the new encryption key is generated.

The encryption method uses AES as the basis, reducing the number of encryption rounds to 4 and changing the original shiftRows and SubBytes to be replaced by Spin-Sort and Cubic S-Box. In order to increase the complexity of encryption, 16-form sorting process of Spin-Sort and two types of Cubic S-Box are used. The entire encryption algorithm is implemented on the FPGA, using HPS software to control encryption and wired transmission and display images instantly on the VGA port.

Finally, we use histogram analysis, correlation analysis, differential attack analysis, and entropy analysis to verify the safety and performance of our proposed scheme. In Section 4, all the experimental results show that the algorithm can encrypt the color image effectively and resist various typical attacks. Our proposed scheme is secure and useful for image encryption.

**Author Contributions:** Conceptualization, C.-H.Y.; methodology, Y.-S.C.; software, C.-H.Y. and Y.-S.C.; validation, C.-H.Y. and Y.-S.C.; formal analysis, C.-H.Y. and Y.-S.C.; data curation, C.-H.Y. and Y.-S.C.; writing—original draft preparation, C.-H.Y. and Y.-S.C.; writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.
References
