Improvement S Box of AES Algorithm Based on FPGA

Huan-qing XU, Yu-ming ZHANG and Jun YANG

School of Information Science and Engineering, Yunnan University, KunMing, China

Keywords: FPGA, AES encryption, Chaotic neural network, Secure processor model.

Abstract. AES is the most mainstream and common encryption standard in the 21st century. It has the advantages of high efficiency, good stability and strong flexibility. It is widely used in e-commerce, encryption hard disk and network transmission encryption. However, in recent years, some AES algorithm has been attacked, which is exposing its S-box simple, single-key and other defects, so the traditional AES algorithm ought to be further improved. Based on FPGA technology, this paper designs and designs a secure processor model of chaotic neural network, realizes the chaotic characteristics of S-box and improves the ability of anti-attack system. The system has the advantages of good reconfiguration, simple circuit structure, low resource consumption and fast running speed, and has good practicability and good application prospect in security encryption and other fields.

Introduction

In this paper, the AES encryption system is designed for the traditional AES algorithm, and the structure of the S-box is improved by using the chaotic neural network. The nonlinear chaotic relationship between the round keys is realized. Ciphertext crack difficulty. In the design process, first of all, we use VHDL hardware language, FPGA-based design and implementation of AES encryption and decryption system, according to the top-down ideas, the system is divided into encryption and decryption module, key expansion module and control scheduling module, In order to solve the simple deflection of the S-box, this paper improves the structure of the S-box by using the chaotic neural network. According to the training samples and logistic functions, the chaotic neural network is used to improve the structure of the S-Function which is determined the weight of the neural network, and the output sequence to replace the traditional S-box; again, this paper will be integrated into a safe processor model, the system space is mapping to allocate, and the use of Nios II processor is to achieve the associated call as well as computer communication control. Finally, we use the Quartus II 13.0 to integrate the cabling and simulation tests. The test results show that the model can meet the requirements of AES encryption and decryption system, and to a certain extent, which is enhancing the security performance and calculation rate of the algorithm.

S Box Transform Module (Sub-byte) and Inverse S Box Transform Module (Inv_Sub-byte)

S transform box also known as byte transform, is the safety grade of the algorithm plays a decisive role in the only nonlinear operation, in order to fully reduce the iterative correlation finally results with the plaintext, S transform box for each byte conversion. The byte substitution mainly has two kinds of design methods, one is based on algebraic operations, each input byte to the initial Status matrix with mathematical calculation of finite field (multiplicative inverse and affine transformation), the final completion of the transformation, which is the transformation function; the second method is through the specific replacement table for the replacement, the input of each according to the finite domain byte substitution table lookup table operation through mapping transformation, the input data as the address, the corresponding unit content as output, to achieve the same byte substitution effect, substitution obtained after the output values are stored in the matrix, called [52] for subsequent iterations. Since the second methods have advantages in programming, hardware implementation and computing speed, we choose the following second methods to achieve S box transformation.
In order to make the lookup table operation is faster, the logic resource utilization rate is higher, we will RAM unit replacement table is stored in the FPGA chip in the encryption process according to the length of the input data to find RAM in the replacement table to complete the fast computation of each byte. The design of the S box with reconfigurable function, improve its security level and application scope, reduce power consumption, can better resist differential attack. According to the description of the second chapter, we use the multiplicative inverse and affine transformation of the finite field to construct the following S box module, as shown in Figure 1, the S box is also suitable for the S box transform module (Inv_Sub-byte).

![Figure 1. S box module diagram.](image)

In order to improve the space utilization ratio, we use the same RAM logic unit for the inverse S box transformation module, and only one port signal is needed to control the encryption and decryption process. Port signal settings are shown in table 1.

<table>
<thead>
<tr>
<th>Serial number</th>
<th>port</th>
<th>length</th>
<th>direction</th>
<th>describe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>clk</td>
<td>1 bit</td>
<td>in</td>
<td>Clock signal, High potential effective</td>
</tr>
<tr>
<td>2</td>
<td>Address_a</td>
<td>8 bit</td>
<td>in</td>
<td>Input data in encrypted mode, Encrypted bytes to be replaced</td>
</tr>
<tr>
<td>3</td>
<td>Address_b</td>
<td>8 bit</td>
<td>in</td>
<td>Decryption mode input data, Decrypted byte to be replaced</td>
</tr>
<tr>
<td>4</td>
<td>q_a</td>
<td>8 bit</td>
<td>out</td>
<td>Encrypted output data</td>
</tr>
<tr>
<td>5</td>
<td>q_b</td>
<td>8 bit</td>
<td>out</td>
<td>Decrypted output data</td>
</tr>
</tbody>
</table>

### Improvement of S Box Based on Chaotic Neural Network

In order to further improve the system security level, strengthen the cyclic nonlinear characteristics of round keys between us, the chaotic neural network to improve the S box based on chaotic neural network with chaotic characteristics due to maintain good, and has strong spatial and temporal complexity and randomness, so we used the output sequence of the chaotic neural network to replace the traditional S box lookup table the key difficulty of crack, enhance the system and improve the level of security.

Chaotic neural network is designed in this paper has 250 S box as the training samples, they are produced by 256 input nodes and Logistic chaotic mapping function in each iteration, the nonlinear S box with differential uniformity of weight ratio, which can satisfy the characteristics of chaos God S box through the network, and gets the design the formula of S box. Part of the training sample data as shown in table 2, after the 150th iteration of the weight distribution map shown in figure 2.

<table>
<thead>
<tr>
<th>Serial number</th>
<th>Nonlinear mean</th>
<th>Mean of difference uniformity</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>108</td>
<td>123</td>
<td>0.88</td>
</tr>
<tr>
<td>2</td>
<td>109</td>
<td>121</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>106</td>
<td>129</td>
<td>0.82</td>
</tr>
<tr>
<td>4</td>
<td>105</td>
<td>138</td>
<td>0.76</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>166</td>
<td>0.65</td>
</tr>
<tr>
<td>6</td>
<td>104</td>
<td>141</td>
<td>0.74</td>
</tr>
<tr>
<td>7</td>
<td>107</td>
<td>124</td>
<td>0.86</td>
</tr>
<tr>
<td>8</td>
<td>108</td>
<td>142</td>
<td>0.76</td>
</tr>
<tr>
<td>9</td>
<td>106</td>
<td>156</td>
<td>0.68</td>
</tr>
<tr>
<td>10</td>
<td>103</td>
<td>137</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Design of S Box Based on Chaotic Neural Network

We can design the S box based on the chaotic neural network by the S box formula which is obtained by the training samples of the last section. In the design process, the function is used in the Logistic chaotic map function:

\[ x_{n+1} = \lambda x_n (1 - x_n) \]

Control parameter: \( \lambda \in (0, 4) \), \( x_n \in [0, 1] \)

Detailed design of the S box algorithm is as follows:

(a) the integer sequence about \( \{0, 1, 2, ..., 255\} \) will arbitrarily permute, any sequence of the resulting floating point sequence \( D \). Transform function is \( D_k = (I_k + 0.1) / 256, k = 1, 2, 3, ..., 256 \), among \( D_k \) and \( I_k \) are corresponding floating point sequences. \( D \) and the sequence \( I \) are used as the input sequence of neural network.

(b) Define integer array \( S \), the initial state of the array is empty.

(c) Set the parameters of the neural network. Iteration Logistic chaotic mapping function \( M \) times, to eliminate transient effects, which \( M \) are constant. Then, the Logistic chaotic map function is iterated and set \( X(M + 1), X(M + 2), ..., X(M + 4096) \) as the weight.

\( W_{1,1}, W_{1,2}, ..., W_{1,16}, W_{2,1}, W_{2,2}, ..., W_{2,16}, W_{256,16} \) are respectively \( X(M + 4097), X(M + 4098), ..., X(M + 4112) \) set deviation \( B_1, B_2, ..., B_{16} \). In the output layer, it will set the \( X(M + 4113), X(M + 4114), ..., X(M + 4240) \) as the weight \( W_{1,1}, W_{1,2}, ..., W_{1,8}, W_{2,1}, W_{2,2}, ..., W_{2,8}, W_{16,1}, W_{16,2}, ..., W_{16,8} \) \( X(M + 4241), X(M + 4242), ..., X(M + 4248) \) and is set to a deviation \( B_{O1}, B_{O2}, ..., B_{O8} \).

(d) In the input layer, the input sequence is transformed \( D \) into the output sequence \( C \) as:

\[ C_j = f^\tau \left( \text{mod} \left( \sum_{j=1}^{256} D_j \times W_{ij} + B_i, 1 \right) \right) \]

The transfer function of the \( f \) is logarithmic, \( \tau \) is the number of iterations \( i = 1, 2, ..., 16 \)

(e) In the output layer, the data sequence \( C \) is transformed into output data by calculating the following equation,

\[ \text{Out} = [\text{Out}_1, \text{Out}_2, ..., \text{Out}_s] \]

\[ \text{Out}_i = f^\tau \left( \text{mod} \left( \sum_{j=1}^{256} C_j \times W_{Oij} + B_{Oi}, 1 \right) \right) \]

The transfer function of the \( f \) is logarithmic, \( \tau \) is the number of iterations \( i = 1, 2, ..., 16 \)
(a) according to the formula, extract the integers $S$ between 0 and 255.

$$S = \sum_{i=0}^{2} \text{mod}(Out_i + 0.5, 1) \times 2^i$$

(b) If $S$ do not in $S_i$, it will be add the $S$ to the $S_i$. If the $S_i$ have 256 data, that $S_i$ will be converted to the 8-8 about $S$ box, the algorithm is complete. Otherwise, swap $D_i$ and $D_{i+1}$ go to step 4 (the $x$ is the number of data in the representation about $S_i$)

(c) if not, it will be added to the. If there are 256 data that will be converted to the 8-8 box, the algorithm is complete. Otherwise, swap and go to step 4 (the number of data in the representation).

The algorithm gets the S box as shown in Figure 3, will be converted to decimal sixteen S box, as shown in figure 4.

![Figure 3. Decimal S box.](image)

| BE | C1 | CB | E9 | 29 | E1 | F1 | FA | C0 | B9 | 7 | AA | AF | C | F | FF | BB | FE | 0 | F6 | ED | 9C | 55 | 72 | 39 | SF | 62 | 4C |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---
When the weight of the improved S box is changed, the look-up table also changes, and the mapping relationship between input and output is changed.

![Figure 5. Simulation of S box based on Chaotic Neural Network.](image)

**Conclude**

Simulation test and performance analysis. Firstly, the AES encryption and decryption subsystem, the improved S neural network and the security processor model are tested and simulated. Secondly, for the improved S box, the performance indexes such as nonlinear, avalanche and difference approximation probability are evaluated. Finally, the performance of the whole platform is analyzed by comparing with other schemes. Analysis shows that the hardware resource consumption and higher this scheme has less throughput than, and the comprehensive performance is very good, can effectively improve the utilization rate of the security processor model of resources, can be widely used in the field of security encryption.

**Acknowledgement**

This research was financially supported by the Yunnan University foundation project.

**References**


