Realization of High-Performance Confidential Data Transmission Based on FPGA

Yu Fan[1,a], Chen Wei[2,b*] , Wang Zhongsheng[3,c]
[1,2,3]School of Computer Science and Engineering,
Xi'an Technological University, Xi'an 710021, P.R.C
[a]yffshun@163.com, [b]408315706@qq.com, [c]59483672@qq.com

Abstract-This paper uses advanced encryption standard (AES) to implement encryption algorithm and FPGA devices to achieve hardware encryption. AES commonly used to provide several security services such as data confidentiality. However, it is a challenge to design efficient hardware architectures with small hardware resource usage. The system is implemented in hardware environment using Verilog HDL. The method has the advantage of full hardware circuitry and can update its own cryptographic algorithm module. Hardware encryption system is based on cryptographic devices and appropriate software program. FPGA can modify the hardware programming repeatedly. The logic circuit module of cryptographic algorithm has a high flexibility in design, because users can define a specific structure of the cipher algorithm module.

Key words- AES; FPGA; hardware encryption

I. INTRODUCTION

Now, whether national or personal, has a number of important confidential data and privacy. It requires reliable data transmission. Otherwise, these vast amounts of information once lost, will bring information infringement, information penetration, and even information crimes. Information encryption technology is the most basic core technology measure and theoretical basis to deal with this risk.

Encryption implemented method includes software encryption and hardware encryption. Software encryption methods fully meet the requirements for personal use. But if the case for the military, financial institutions, high security, large amount of data, shortcomings will be exposed as follows: software encryption will take up more PC resources , the program is easy to track, manage key is not easy, virus software especially backdoor Trojans pose a threat and so on. So now some dedicated encryption chips designed for security-sensitive environments. Hardware encryption has the following advantages: fast encryption speed, simple operation, good physical isolation, and can effectively prevent the reverse break. Most hardware encryption products in China are based ASIC.

In security applications, an encryption algorithm can't always guarantee safety. Encryption algorithm used in the application can constantly update, and ASIC does not meet this requirement, thereby greatly increasing its cost. Field Programmable Gate Array, FPGA, is very well suited for high speed cryptography as they can provide the required performance without the excessive cost and it is an integrated circuit that can be bought off the shelf and reconfigured by designers themselves. With each reconfiguration, which takes only a fraction of a second, an integrated circuit can perform a completely different function. Cryptographic algorithms chips of hardware encryption system commonly used anti-fuse FPGA or ASIC implementation, it is difficult to break or change [1-4].

II. SYSTEM ARCHITECTURE AND IMPLEMENTED METHOD

The system is in the middle position between PC and USB storage media, and system accomplishes hardware encryption process as the data path. The main parts of the AES are hardware encryption and decryption module, CY7C68013 module and CH376 module, power supply circuit, downloads circuit and other circuits [5-6].

Realization of hardware encryption mainly has the following two methods:

A. Implementation based on microprocessor or DSP

The way to realize based on Microprocessor or DSP is essentially a software implementation. Algorithm has a great flexibility in the design and it's very easy to update and modify. However, this method finishes cryptographic operation based on a microprocessor or DSP to perform cryptographic operation, the capability of microprocessor or DSP directly affects the password encryption card. Besides, cryptographic algorithm code will be saved in the program memory, once the encryption card is stolen or lost, unauthorized users can easily get the firmware algorithm code. Since the code has legibility, cryptographic algorithm has the risk of leak.

B. Implementation based on FPGA

Cryptographic algorithm module based on FPGA uses the way of hardware circuit only. The speed of cryptographic operations is not as fast as a dedicated chip cryptographic algorithm, but the requirements of the encryption algorithm can be achieved. And due to its programmability, it can update its own cryptographic algorithm module, so it's more convenient to evaluate new algorithms.

The system design is based on the FPGA design.

III. AES ALGORITHM PRINCIPLE AND IMPLEMENTATION

AES encryption algorithm as a new generation of data encryption standard brings strong security, high performance, high efficiency, ease of use and flexible benefits together. AES is designed with three key lengths: 128, 192, 256 bits. On contrast, AES-128 is stronger than DES-56. AES encryption algorithm mainly includes three aspects: round transformation, turns and key expansion.

A. AES Algorithm Structure

AES is a grouping key, 128-bit data input, the key length is also128-bit. AES is an iterative type password: round depends on the key length. If the key length is 128-bit, then N_{r} = 10; If the key length is 192-bit, then N_{r} = 12; If the key length is 256-bit, then N_{r} = 14. AES algorithm inputs and outputs can be regarded as 0, 1 sequences, the key as Fig. 1 shows a Rijndael encryption process for 128-bit plain text data string and a 128-bit key, with the 10 rounds. These numbers are used throughout this paper, including for our hardware implementation. The input data is arranged as a 4×4 matrix of bytes. The primitive functions SubBytes, ShiftRows and MixColumns are based on byte-oriented arithmetic, and AddRoundKey is a simple 128-bit XOR operation.

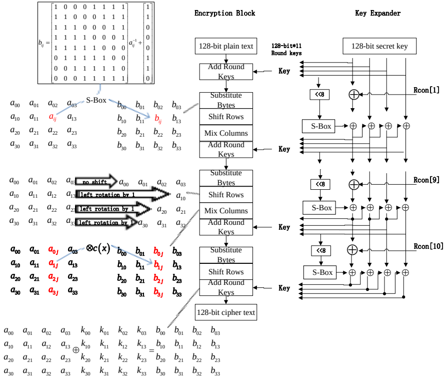

Fig 1.The schematic diagram of AES algorithm

SubBytes is a nonlinear transformation that uses 16 byte S-Boxes. An S-Box is the multiplicative inverse of $GF\left(2^8\right)$ followed by an affine transformation. In the decryption process, the affine transformation is executed prior to the inversion. The irreducible polynomial used by a Rijndael S-Box is:

$$\mathrm{m}(x)=x^8+x^4+x^3+x+1. \tag{1}$$

ShiftRows is a cyclic shift operation of the last three rows by different offsets. MixColumns treats the 4-byte data in each column as coefficients of a 4-term polynomial, and multiplies the data modulo X^4+1 with the fixed polynomial given by:

$$c(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\}. \tag{2}$$

In the decryption process, InvMixColumns multiplies each column with the polynomial:

$$c^{-1}(x)=\{0B\}x^3+\{0D\}x^2+\{09\}x+\{0E\}. \tag{3}$$

and InvShiftRows shifts the last three rows in the opposite direction from ShiftRows.

As we know, the input and the output of Rijndael cipher algorithm are all 128-bit. So we use X to represent the input and Y to represent the output. The follow equation can express Rijndael cipher algorithm.

$$Y{=}O_{K_{r+1}}\circ T\circ\Gamma\circ O_{K_r}\circ\Pi\circ T\circ\Gamma\circ O_{K_{r-1}}\circ\cdots\Pi\circ T\circ\Gamma\circ O_{K_1}(X). \tag{4}$$

$$O_{K_i}:X\in F_2^{128},O_{K_i}(X)=X\oplus K_i. \tag{5}$$

The input of T is:

$$X=\left(X_{00,}X_{01,}X_{02,}X_{03,}X_{10,}X_{11,}X_{12,}X_{13,}X_{20,}X_{21,}X_{22,}X_{23,}X_{30,}X_{31,}X_{32,}X_{33}\right). \tag{6}$$

The output of T is:

$$Y=T(X)=\left(X_{00,}X_{01,}X_{02,}X_{03,}X_{10,}X_{11,}X_{12,}X_{13,}X_{20,}X_{21,}X_{22,}X_{23,}X_{30,}X_{31,}X_{32,}X_{33}\right). \tag{7}$$

The input of Π is:

$$X=\left(X_{00,}X_{01,}X_{02,}X_{03,}X_{10,}X_{11,}X_{12,}X_{13,}X_{20,}X_{21,}X_{22,}X_{23,}X_{30,}X_{31,}X_{32,}X_{33}\right). \tag{8}$$

The output of Π is:

$$Y=\Pi(X)=\left(Y_{00,}Y_{01,}Y_{02,}Y_{03,}Y_{10,}Y_{11,}Y_{12,}Y_{13,}Y_{20,}Y_{21,}Y_{22,}Y_{23,}Y_{30,}Y_{31,}Y_{32,}Y_{33}\right). \tag{9}$$

Among these:

$$\begin{bmatrix}Y_{0i}\\Y_{1i}\\Y_{2i}\\Y_{3i}\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&02\end{bmatrix}\begin{bmatrix}X_{0i}\\X_{1i}\\X_{2i}\\X_{3i}\end{bmatrix}. \tag{10}$$

The matrix is a hexadecimal value, for example: 02 is a bit string of 00000010.

Γ : a constitute of 16 S-boxes juxtaposition, $S=L\circ F$.

F : a multiplicative inverse of a finite field named F_{2^8} :

$$F(X)=X^{-1},L(X)=AX+b. \tag{11}$$

$$A=\begin{bmatrix}1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\\1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\end{bmatrix},b=\begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix}. \tag{12}$$

Decryption process is the inverse of the encryption process.

B. *Hardware Implementation of AES Encryption and Decryption Module*

In this paper the design of AES encryption module is implemented in an FPGA. If implement various operation of AES algorithm in the logic circuit directly, will inevitably result in the large chip area and the waste of resources. The entire module structure of AES encryption and decryption is shown in Fig. 2:

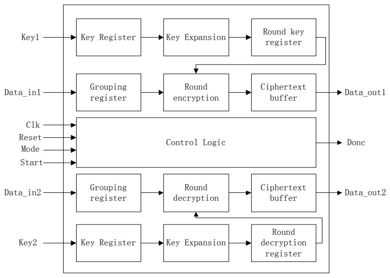

Fig 2. The entire module structure of AES encryption and decryption

AES encryption and decryption module is mainly composed by grouping registers, round encryption and decryption operations unit, the plaintext cipher text buffer, state machine (control logic), the key registers, key expansion units, round key memory and other key unit.

Encryption system involves a large number of process data, so we use pipelining ways. Encryption system uses a state machine approach to achieve control modules, shown in Fig. 3.

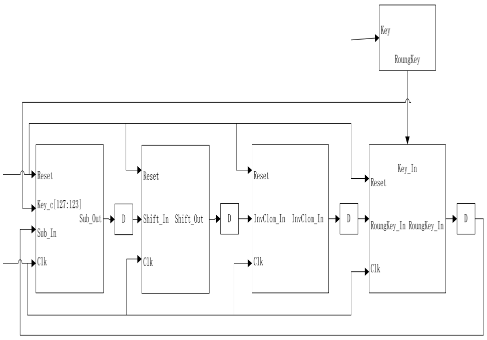

Fig 3. Block diagram of the encryption system

We use five stages to achieve encryption:

a) The start of encryption.

b) The XOR operation of the initial data and the initial key.

c) SubBytes, ShiftRows, MixColumns , AddRoundKey and the iterative encryption of nine operators.

d) The last encryption: SubBytes, ShiftRows and AddRoundKey.

e) End.

IV. Functional Simulation

We give different input plaintexts and keys, observe the simulation results of encryption and decryption and get two functional simulation figures shown as follows.

The input of Fig. 4 is 0, the key is ffffffffffffffffffffffff1ffffff8, the encrypt data is 5a4d404d8917e353e92a21072e3b2305 and the decrypt data is 0.

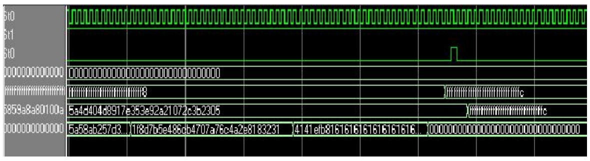

Fig 4. Encryption

The input of shown as Fig. 5 is 8000000000000000000000000000, the key is 0, the encrypt data is 3ad78e726c1ec07bebfe92b23d9ec34 and the decrypt data is 8000000000000000000000000000.

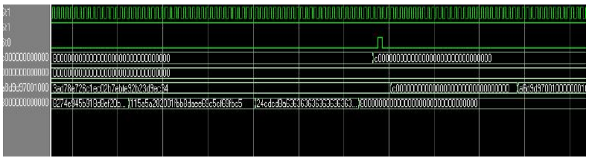

Fig 5. Decryption

After the above verification, you can see the encryption and decryption function correctly.

V. Conclusion

The demand for security device increases because of big data. Now hardware encryption card in the market is mainly based on the PCI (Peripheral Component Interconnect), system encrypts and decrypts data for USB in real time is relatively less. This paper discusses the direction, and has done some simulations. The experiment proved that the system is feasible.

The data encryption system has the characteristics of fast speed, simple operation, reliable, FPGA mode encryption flexibility further extend, and etc. But the system also needs to be improved and perfected, in order to achieve the purposes of complete application and commercial.

References

[1]Deshpande, Mangesh S, Kayatanavar, Devendra N, “FPGA implementation of AES encryption and decryption,” International Conference on Control Automation. (2009)

[2]Yu-xin Wang, “Design and Implementation of USB Image Capture controller Based on FPGA,” Xi'an University of Electronic Science and Technology. (2012)

[3]Fang Ren, Cheng-rui Yang, Lei-hua Chen, 3DES Implementation Based on FPGA, Journal of Xi’an Polytechnic University, vol. 25, 2011, pp. 555-559.

[4]Wei Yuan, The Design Principles of AES Algorithm and The Improvement of Its Key Expanding Algorithm, Jilin University. (2010)

[5]Kakarountas, Athanasios P, Goutis, Costas E, “A high-throughput area efficient FPGA implementation of AES-128 encryption,”

Workshop on Signal Processing Systems, IEEE. 2005, pp. 116-121.

[6]Jian-jun Guo, Chang-ming Liu, Yin Shi, Xia Lu, Jian Zhang, Hardware Implementation Analysis and Design of AES Algorithm. (2012)