High-speed Hardware Design for the Twofish Encryption Algorithm

Choong-Mo Youn and Beom-Geun Lee, *Member, KIMICS*

Abstract—Twofish is a 128-bit block cipher that accepts a variable-length key up to 256 bits. The cipher is a 16-round Feistel network with a bijective F function made up of four key-dependent 8-by-8-bit S-boxes, a fixed 4-by-4 maximum distance separable matrix over Galois Field(GF (2^8)), a pseudo-Hadamard transform, bitwise rotations, and a carefully designed key schedule. In this paper, the Twofish is modeled in VHDL and simulated. Hardware implementation gives much better performance than software-based approaches.

Index Terms—high speed, Twofish encryption algorithm

I. INTRODUCTION

With the recent advances in communication technology and the increasingly universal use of the Internet, information exchange through the computer communication network has become very active. Advances in networked computing have enabled the transfer of much amounts of information and have made human life easier. However, the absence of efficient security in the early version of protocols made networks vulnerable to illegal accesses and, hence, the data security of the data stored or being transferred became a critical issue.

In 1977, IBM and the National Institute of Standards & Technology (NIST) developed the Data Encryption Standard (DES). Later, DES became the American National Standards Institute (ANSI) standard and gets re-authentication every five years. However, DES with only 56 bits was decoded as computer hardware technology advanced, and many problems in security have been presented. To solve these problems, the NIST publicly collected the algorithm for the next-generation encryption standard in January 1997. The NIST suggested the standards for the new encryption algorithm such that it should use 128-bit block with 128-, 192-, and 256-bit key lengths.

In this paper, the Twofish algorithm, which can guarantee the security of the encrypted data, is implemented in VHDL and modeled to be verified by simulation.

II. TWOFISH ALGORITHM

Manuscript received October 10, 2005.

Choong-Mo Yun is with the School of Information Technology, Seoil College, Seoul, 131-701, Korea (Tel: +82-2-490-7408, Fax: +82-2-490-4783, Email: 5477choong@hanmail.net)

Beom-Geun Lee is with the Department of Electronic Engineering, Kyunghee University

Twofish is a 128-bit block encryption algorithm with symmetric key type, which can support various key lengths from 128 bits to 256 bits[1].

Twofish has a structure similar to that of the feistel network and is composed of 16 rounds and includes a whitening process for input and output. The difference from the pure feistel network is that its structure allows the output of the F function and is end-around shifted by 1 bit. In this paper, it is implemented for only the 128-bit key length.

(Fig. 1) shows the block diagram of the whole structure of the Twofish encryption algorithm.

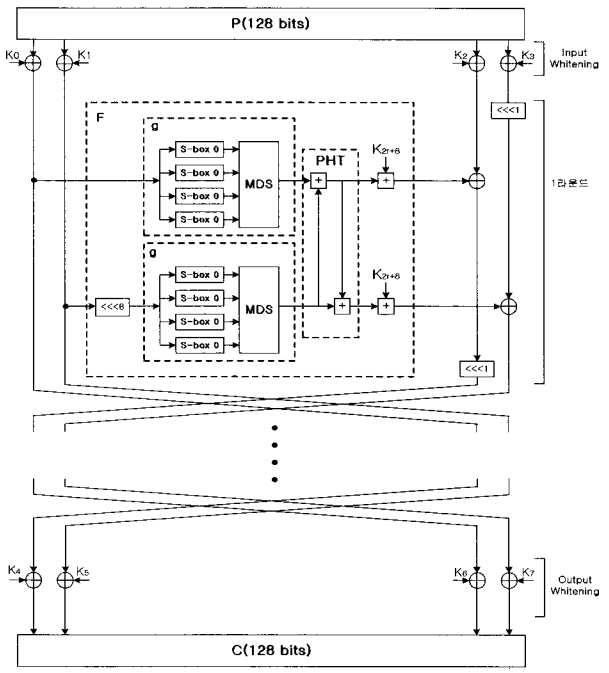


Fig. 1 Block Diagram of the Twofish encryption algorithm

Two outputs of the F function are end-around shifted by 1 bit after XOR operation with the two words on the right. Two outputs of the F function after the XOR operation with two words on the left get the position exchange operation for the next round.

A. F function

F function has the permutation specific to the 64-bit key value. The R_0, R_1 word on the left each having 32 bits inputted for each round are used for the two inputs for the g function in F function and the one inputted R_1 word is end-around shifted to the left during inputting. At this time, the two outputs of the g function is combined with the pseudo-Hadamard transform (PHT) and two-part keys are computed with 2^{32} modular addition

The functions below shows the T_0,T_1,F_0 and F_1 Operation of the F function.

$$T_0 = g(R_0)$$
$$T_1 = g(ROL(R_1,8)) \quad \text{(Eq 1)}$$

$$F_0 = (T_0 + T_1 + K_{2r+8}) \bmod 2^{32}$$
$$F_1 = (T_0 + 2T_1 + K_{2r+9}) \bmod 2^{32} \quad \text{(Eq 2)}$$

From the equation (2), T_0 and T_1 are computed from equation (1) and (F_0,F_1) is the result of the F function.

(Fig. 2) is the block diagram of the single-round F function.

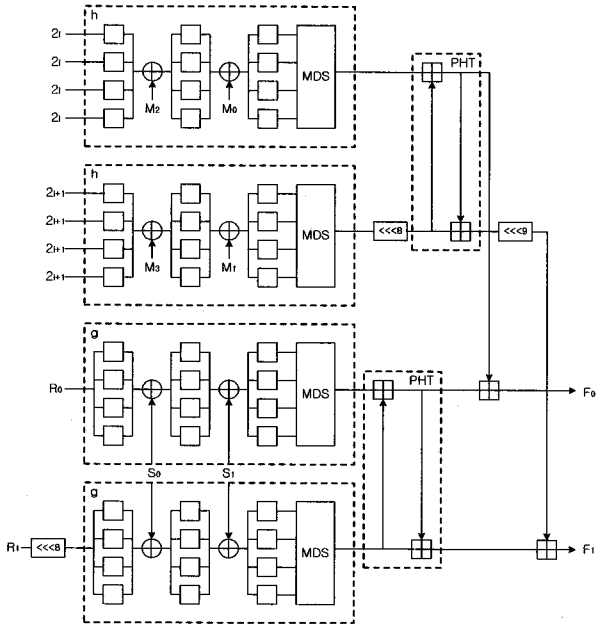


Fig. 2 Single-round F function

(Fig. 2) shows that the 128-bit plaintext is divided by 4 32-bit words and each word is processed by whitening process with 4-part keys of the 32 bits and XOR operation.

B. g function

The g function is composed of S-box specific to the key value of four 8x8-bit matrices and maximum distance separable (MDS) matrix multiplier. The output of the two g functions is combined by the PHT and the two-part keys are added by the 2^{32} modular addition.

And the g function is composed of S block of 4 bytes. The inputted word X is divided into 4 bytes and each byte is executed through the S block specific to its key value. The 4x4-byte MDS matrix multiplier in the g function is computed using Galois Field (GF (2^8)).

Since all the operations of the Twofish are done with the unit of byte, all the variables are computed in a finite field, GF(2^8), and, if there is an eight-degree primitive polynomial whose coefficients are only 0 or 1 and which are elements of GF(2^8), all elements excluding 0 of the GF(2^8) can be expressed as the root of the polynomial. The primitive polynomial is denoted as *v(x)*.

$$v(x) = x^8 + x^6 + x^5 + x^3 + 1$$
$$x_i = [X / 2^{8i}] \bmod 2^8 \qquad i = 0,\cdots,3$$
$$y_i = s_i[x_i] \qquad i = 0,\cdots,3 \quad \text{(Eq 3)}$$

From equation (3), X is an input work and S_i is a key-dependent S-box.

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = (MDS) \bullet \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$
$$MDS = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix} \quad \text{(Eq 4)}$$

In equation (4), MDS is a given hex value 4 x 4 matrix

$$Z = \sum_{i=0}^{3} z_i \bullet 2^{8i} \quad \text{(Eq 5)}$$

In equation (5), Z is the result of the g function.

C. Key schedule

Key schedule provides extension keys from K_0 to K_{39}. K is defined as N/64 and N denotes the key length. For example, N can be selected by the user between 128, 192 and 256 as the key length.

Key scheduling can be divided into two parts. One is for generating 32-bit S_0 and S_1 that are used in S block to make S blocks specific to key values using the inputted 128-bit key. S_0 and S_1 are generated by Reed–Solomon multiplication in GF(2^8) and can be expressed as the following equation.

$$v(x) = x^8 + x^6 + x^3 + x^2 + 1$$
$$\begin{pmatrix} s_{i,0} \\ s_{i,1} \\ s_{i,2} \\ s_{i,3} \end{pmatrix} = (RS) \bullet \begin{pmatrix} m_{8i} \\ m_{8i+1} \\ m_{8i+2} \\ m_{8i+3} \\ m_{8i+4} \\ m_{8i+5} \\ m_{8i+6} \\ m_{8i+7} \end{pmatrix} \quad \text{(Eq 6)}$$
$$RS = \begin{pmatrix} 01 & A4 & 55 & 87 & 5A & 58 & DB & 9E \\ A4 & 56 & 82 & F3 & 1E & C6 & 68 & E5 \\ 02 & A1 & FC & C1 & 47 & AE & 3D & 19 \\ A4 & 55 & 87 & 5A & 58 & DB & 9E & 03 \end{pmatrix}$$

$$S_i = \sum_{j=0}^{3} s_{i,j} \bullet 2^{8j} \quad \text{(Eq 7)}$$

In equation (7), $S = (S_{k-1}, S_{k-2}, ..., S_0)$ when I=0,..., k-1. The remaining is for generating 40 32-bit part keys used in each round, input, output, and whitening. Since the operation in this part is done with the g function in encryption and decryption and fixed circulation additional to the PHT structure, the g function in encryption and decryption and PHT can be used as it is.

$$a' = a + b \bmod 2^{32}$$
$$b' = a + 2b \bmod 2^{32} \quad \text{(Eq 8)}$$

D. Pseudo-Hadamard transforms (PHTs)

PHT can be expressed as for the two inputs as in equation (8).

III. DESIGN STUDY AND IMPLEMENTATION

Twofish is a type that repeats 16 identical rounds in performing encryption and decryption. For implementing this in hardware, one may design one block and reuse it 16 times. This method has the advantage that the hardware loss is small but the state of each round should be controlled and the result should be stored. (Fig. 3) shows the entire block diagram of Twofish.

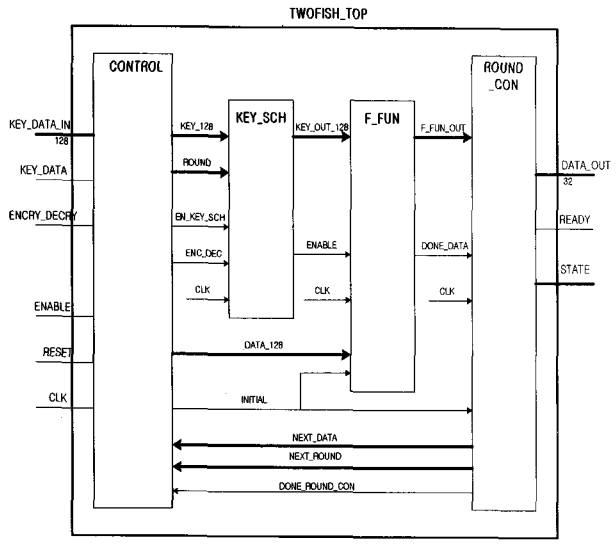


Fig. 3 Block Diagram

In (Fig. 3), TWOFISH_TOP block is the actual encryption and decryption. TWOFISH_TOP block can be mainly divided into CONTROL, KEY_SCH, F_FUN, and ROUND_CON. For hardware implementation, the data input is set to 128 bits and the output is done by four 32-bit sets (dividing 128 bits into four parts) considering the number of input/output pins and encryption processing speed.

If the RESET value is set to 1, TWOFISH_TOP is initialized. KEY_DATA_IN gets the 128-bit key values or 128-bit data values from the external with one 128-bit bus. If the value of KEY_DATA is 0, the value of KEY_DATA_IN bus is KEY; if it is 1, the value of KEY DATA IN is DATA. ENCRY DECRY selects encryption or decryption according to the TWOFISH_TOP signal such that it does encryption for value 1 and decryption for 0. Only when ENABLE is 1, TOP accepts the data in KEY_DATA_IN. DATA_OUT is for outputting the result of encryption or decryption and it outputs the result with four 32-bit sets by dividing 128 bits into four parts. If READY is 1, the result which is encoded in TWOFISH_TOP is outputted. In case of RESET, READY is set to 0. STATE is the 2-bit output signal for showing DATA in DATA_OUT corresponds to which part of the 128-bit data is for outputting.

CONROL block is for controlling the external interface and internal operation and controls the entire TWOFISH_TOP.

KEY_SCH block is for generating KEY values corresponding to each round. When ROUND is "00000," which is the initial state, if the EN_KEY_SCH is set to 1, the KEY value of each round is outputted to KEY_OUT_128 after computing KEY values corresponding to each round with the data in KEY_128.

F_FUN block performs encryption and decryption with the key value generated in KEY_SCH and data in bus DATA_128.

ROUND_CON block determines the current round in the process of encryption and decryption and outputs the result of F_FUN to the external if the 16 rounds are processed; otherwise, the result is returned to the control block to process the next round.

IV. RESULT AND CONCLUSION

Using the structure suggested in this paper, the encrypted result with 128-bit key (00000000000000000000000000000000 589:HEX) and 128-bit data (01234567890ABCDEF 01234567 89ABCDEF:HEX) is the 128-bit output data (123456789ABC DEF 0123456789ABD232:HEX) as shown in (Fig. 4).

The process of decoding the result by inputting it again is shown in (Fig. 5): The encoded output data (0123456789ABCD EF0123456789ABD232:HEX) is decoded to hexadecimal (01234567890ABCDEF0123456789ABCDEF:HEX).

In this paper, the 128-bit block encryption chip is designed using the Twofish encryption algorithm. While the existing DES uses a fixed table where the S box is fixed regardless of the key, the S box of Twofish is dependent on key value and uses functions that change according to the key schedule. Hence, estimating the attack complexity is very difficult for Twofish since S box is unknown if the key is unknown while such is possible for DES since the S box can be known even if the key is unknown.

In this paper, each block of the 128-bit Twofish is described with the hardware description language VHDL and the Twofish algorithm which is for guaranteeing the security of the encrypted data is modeled and simulated in Max-Plus II.

The result was verified using the test vector given in Twofish algorithm and it is shown that the encryption and decryption is performed stably.

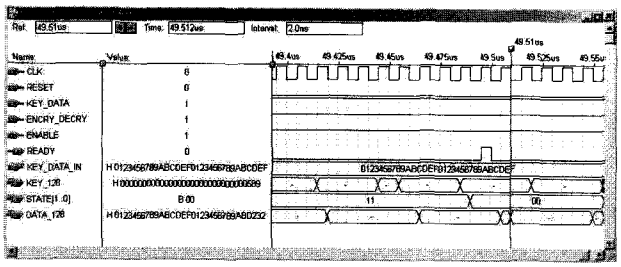


Fig. 4 Result of Encryption

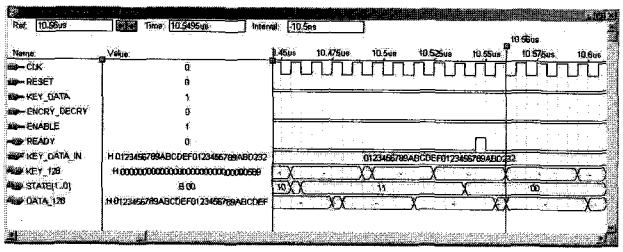


Fig. 5 Result of Decryption

REFERENCES

[1] Feistel, "Cryptography and Computer Privacy", *Scientific American*, May 1973.

[2] Pawel Chodowiec, Kris Gaj, "Implementation of the Twofish Cipher Using FPGA Devices", Technical Report, George Mason University,1999.

[3] Brace Schneier, "Applied Cryptography", John Wiley & Sons Inc., 1996.

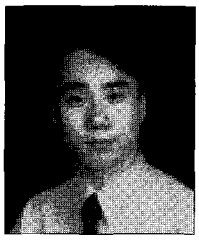
[4] Feistel, "Cryptography and ComputerPrivacy", *Scientific American*, 1973.

[5] C. Paar, "Optimized Arithmetic for Reed–Solomon Encoders", 1997 IEEE International Symposium on Information Theory, 1997.

[6] C. Paar, M.Rosner, "Comparison of Arithmetic Architectures for Reed-Solomon Decoders in Reconfigurable Hardware", Fifth Annual IEEE Symposium on Field-Programmable Custom Computing Machines FCCM '97. 1997. USA.

Choong-Mo Youn

He received the M.S degree in Electronic Engineering from Dankook University in 1990 and Ph.D degree in Electronic Engineering for Chongju University in 2000. From 1992 to present, he is a professor, School of Information Technology Seoil College in Korea. His research interests include Digital system and CAD, Network.



Beom-Geun Lee

He received the M.S degree in Electronic Engineering from Chongju University in 1997. Since 1997 to now, he has been Ph.D course student in Electronic Engineering, Kyunghee University Suwon, Korea. His research interests include Computer system and security.