논문 2005-42SD-10-5

Twofish 암호알고리즘의 처리속도 향상을 위한 MDS 성능개선에 관한 연구

(A Study on the MDS performance improvement for Twofish cryptographic algorithm speed-up)

이 선 근*, 김 환 용*

(Seon Keun Lee and Hwan Yong Kim)

요 약

본 논문은 Rijndael 암호알고리즘에 비하여 알고리즘 자체가 간결하며 구현의 용이성이 좋지만 처리속도가 느린 단점을 가진 Twofish 암호알고리즘의 속도를 향상시키기 위하여 MDS 블록을 새롭게 설계하였다. 설계된 MDS 블록은 Twofish 암호시스템의 critical path를 점유하게 되는 블록으로서 처리과정중의 병목현상으로 인한 속도저하의 문제점이 존재하였다. 이러한 MDS 블록에서 연산자로 사용되는 곱셈연산을 LUT 연산과 modulo-2 연산을 사용하여 MDS 자체에 대한 속도저하 및 병목현상을 제거하였다. 이러한 결과로 새롭게 설계된 MDS 블록을 포함하는 Twofish 암호시스템은 기존 Twofish 암호시스템에 비하여 10%정도 처리속도의 향상을 가져옴을 확인하였다.

Abstract

Treatise that see designed MDS block newly algorithm itself is concise and improve the speed of Twofish cryptographic algorithm that easy of implement is good but the processing speed has slow shortcoming than Rijndael cryptographic algorithm. Problem of speed decline by a bottle-neck phenomenon of processing process existed as block that designed MDS block occupies critical path of Twofish cryptographic system. Multiplication arithmetic that is used by operator in this MDS convex using LUT arithmetic and modulo-2 arithmetic speed decline and a bottle-neck phenomenon about MDS itself remove. Twofish cryptographic system including MDS block designed newly by these result confirmed that bring elevation of the processing speed about 10% than existing Twofish cryptographic system.

Keywords: Twofish, Cryptographic algorithm, Cryptosystem, MDS, VLSI Design

I. 서 론

1977년부터 암호알고리즘 표준안으로 제정되어 사용되던 DES(Data Encryption Standard) 암호알고리즘이 플랫폼의 발달과 네트워크 환경의 다변화로 인하여 더 이상 암호 표준안으로의 기능이 유명무실하게 되었다. 따라서 NIST는 AES(Advanced Encryption Standard)를 공모하여 2000년도에 Rijndael 암호알고리즘을 최종 표준안으로 제정하였다.

그러나 AES의 최종 후보들인 다섯 개의 후보 알고리즘(Twofish, Rijndael, MARS, RC6, Serpent) 중 암호의 안전성과 성능에서 인정을 받은 것이 Twofish 암호알고리즘이다. 이와 대조적으로 2000년도 이후 채택된 Rijndael 암호알고리즘은 암호/복호화를 동시에 수행할 수 없으며 구현상의 어려움이 많기 때문에 아직까지 Rijndael 암호알고리즘은 활발하게 사용되지 못하고 있다.[1][4][6]

Twofish 암호알고리즘은 Rijndael 암호알고리즘에 비하여 처리속도는 다소 낮지만 구현상의 용이성 및 암

* 정회원, 원광대학교 전기전자및정보공학부
(Department of Electrical Electronic and Information Engineering, Wonkwang University)
※ 본 연구는 정보통신부에서 지원하는 기초기술연구 지원사업의 연구결과입니다.
접수일자: 2005년5월31일, 수정완료일: 2005년9월7일

호/복호화가 동시에 수행될 수 있다는 장점 때문에 Rijndael 암호알고리즘에 비하여 폭넓게 연구되고 있으며 사용되고 있다.

그러므로 본 논문에서는 이러한 Twofish 암호알고리즘의 처리속도를 향상시켜 Rijndael 암호알고리즘의 처리속도에 비슷한 속도를 가질 수 있도록 하였다.[1][3]

Ⅱ. Twofish 암호알고리즘

대칭키 블록 암호 알고리즘인 Twofish 암호 알고리즘은 블록 크기 128 비트에 대하여 키 길이를 128, 192, 256 비트로 다양한 길이를 가질 수 있는 가변 블록 대칭형 암호알고리즘이다. 그림 1은 Twofish 암호 알고리즘의 기본 구조를 나타낸다. Twofish 암호 알고리즘은 입/출력의 whitening 과정을 포함하며, F 함수의 출력이 한 비트씩 순환된다는 점을 제외하면 16라운드 Feistel 네트워크 구조와 유사하다.[1]

그림 1과 같이 twofish 암호알고리즘에서 128 비트 평문은 little-endian convention을 사용하여 4개의 32비트 워드로 나누어지고, 각각의 워드는 32비트 부분키(subkey) 4개와 XOR 되어 입력 whitening 과정을 거친다. 각 라운드에서는 좌측 2개의 워드가 F-함수내의 두 g-함수의 입력으로 사용된다. 이때 1개의 입력 워드는 8비트 좌측 순환을 거쳐 입력된다. g-함수는 4개의 8-by-8비트 킷값에 종속된 S-box들과 MDS 행렬 곱셈기로 구성된다. 두 g-함수의 출력은 Pseudo-Hadamard Transform(PHT)을 이용하여 결합되고, 2개의 부분키가 modulo 2 덧셈에 의해 더해진다. 이때 F-함수의 두 출력은 다음 라운드를 위하여 자리바꿈을 수행한다. 16라운드 후에는 마지막 라운드의 결과가 다시 자리바꿈되고, 출력 whitening을 거쳐 128비트 평문에 적용한 little-endian convention과 같은 방법으로 128 비트의 암호문을 생성한다. 복호화 과정은 암호화 과정에서 적용한 40개의 부분키의 순서 및 F-함수의 출력이 우측 2개의 워드와 XOR 및 한 비트 되는 과정을 역으로 적용하여 이루어진다.[3][4]

g-함수내의 4-by-4 바이트 MDS 행렬 곱셈기는 Twofish 내의 주요한 확산 메카니즘으로 사용되며 원시다항식은 식 (1)과 같다.

$$g(x) = x^8 + x^6 + x^5 + x^3 + x + 1 \quad (1)$$

Galois Field(2^8)상에서의 MDS 행렬 곱셈은 식 (2)와 같이 표현된다.

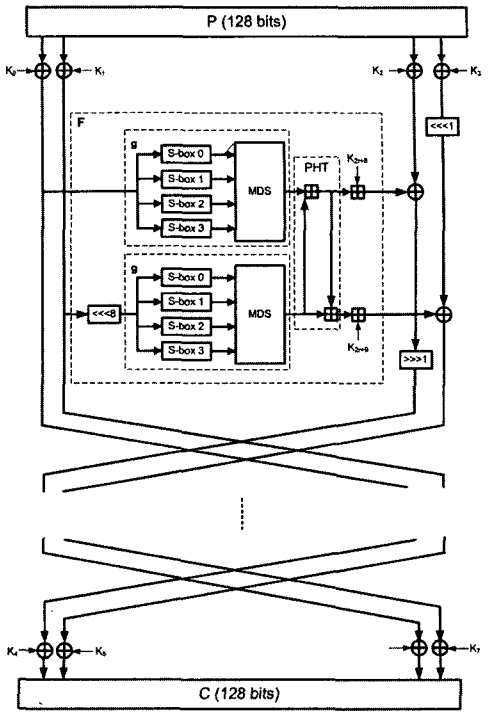

그림 1. Twofish 암호 알고리즘의 블록도
Fig. 1. Twofish cryptographic algorithm block diagram.

$$\begin{vmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{vmatrix} = \begin{vmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{vmatrix} \cdot \begin{vmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{vmatrix} \quad (2)$$

또한, 외부 128비트 키를 입력으로 받아 40개의 32비트 부분키 및 S-box에서 사용되는 S_0, S_1 부분키를 생성하는 키 스케쥴링 부분은 암호/복호 부분의 두 g-함수와 PHT 및 추가적인 고정된 순환, RS-곱셈기 등으로 구성된다.[2]

Ⅲ. 처리속도 향상을 위한 제안된 MDS 블록

Twofish 암호알고리즘을 구현할 때 처리시간이 가장 많이 소요되는 블록이 MDS 블록이다. MDS 블록은 F 함수 내에 존재하며 식 (2)와 같이 곱셈연산을 수행해야 한다. 곱셈연산 수행으로 인하여 Twofish 암호시스템의 가장 중요한 블록인 MDS 블록에서 병목현상이 발생하고 이로 인하여 처리시간의 감소를 가져온다.

본 논문에서는 이러한 곱셈연산에 의한 처리시간의 감소를 억제하고자 Look-Up Table(LUT)과 modulo-2 연산을 사용하여 데이터 처리시간을 높여 Twofish 암호알고리즘의 전체 암호화 수행시간을 향상시키고자 하였다.

기존 Twofish MDS블록의 연산식인 식 (2)를 MDS 연산시간의 향상을 위하여 식 (3)과 같이 변형한다.

$$\begin{aligned} Y_0 &= 01 \cdot X_0 + EF \cdot X_1 + 5BX_2 + 5B \cdot X_3 \\ Y_1 &= 5B \cdot X_0 + EF \cdot X_1 + EFX_2 + 01 \cdot X_3 \\ Y_2 &= EF \cdot X_0 + 5B \cdot X_1 + 01X_2 + EF \cdot X_3 \\ Y_3 &= EF \cdot X_0 + 01 \cdot X_1 + EFX_2 + 5B \cdot X_3 \end{aligned} \quad (3)$$

이때 $01 \cdot X$ 연산의 결과 값은 자기 자신이 되므로 식 (3)에서 $01 \cdot X$의 부분을 생략하여 정리하면 식 (4)와 같다.

$$\begin{aligned} Y_0 &= EF \cdot X_1 + 5BX_2 + 5B \cdot X_3 \\ Y_1 &= 5B \cdot X_0 + EF \cdot X_1 + EF \cdot X_2 \\ Y_2 &= EF \cdot X_0 + 5B \cdot X_1 + EF \cdot X_3 \\ Y_3 &= EF \cdot X_0 + EFX_2 + 5B \cdot X_3 \end{aligned} \quad (4)$$

식 (4)에서 MDS는 0x5B와 0xEF에 해당하는 연산을 수행하면 되므로 MDS의 0x5B와 0xEF에 대하여 modulo-2를 이용하여 전개하면 식 (5)와 같이 표현할 수 있다.

$$\begin{aligned} y_7 &= x_7 \oplus x_1 & y_7 &= x_7 \oplus x_1 \\ y_6 &= x_6 \oplus x_0 & y_6 &= x_7 \oplus x_6 \\ y_5 &= x_7 \oplus x_5 \oplus x_1 & y_5 &= x_7 \oplus x_6 \oplus x_5 \oplus x_1 \\ y_4 &= x_6 \oplus x_4 \oplus x_1 \oplus x_0 & y_4 &= x_6 \oplus x_5 \oplus x_4 \oplus x_1 \\ y_3 &= x_5 \oplus x_3 \oplus x_0 & y_3 &= x_5 \oplus x_4 \oplus x_3 \oplus x_0 \\ y_2 &= x_4 \oplus x_2 \oplus x_1 & y_2 &= x_4 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_0 \\ y_1 &= x_3 \oplus x_1 \oplus x_0 & y_1 &= x_3 \oplus x_2 \oplus x_1 \oplus x_0 \\ y_0 &= x_2 \oplus x_0 & y_0 &= x_2 \oplus x_1 \oplus x_0 \end{aligned} \quad (5)$$

식 (5)와 같이 곱셈연산은 단순 modulo-2 연산을 이용하면 쉽게 해결된다. 이때 식 (5)를 간략화 하면 식 (6)과 같다. 이때 a는 0x5B를, b는 0xEF항을 의미한다.

$$\begin{aligned} y_{7a} &= x_7 \oplus x_1, & y_{7b} &= x_7 \oplus x_1 \\ y_{6a} &= x_6 \oplus x_0, & y_{6b} &= x_7 \oplus x_6 \\ y_{5a} &= y_{7a} \oplus x_5, & y_{5b} &= y_{6b} \oplus x_5 \oplus x_1 \\ y_{4a} &= y_{6a} \oplus x_4 \oplus x_1, & y_{4b} &= y_{4a} \oplus x_5 \\ y_{3a} &= x_5 \oplus x_3 \oplus x_0, & y_{3b} &= y_{3a} \oplus x_4 \\ y_{2a} &= x_4 \oplus x_2 \oplus x_1, & y_{2b} &= x_4 \oplus y_{1b} \\ y_{1a} &= x_3 \oplus x_1 \oplus x_0, & y_{1b} &= x_3 \oplus y_{0b} \\ y_{0a} &= x_2 \oplus x_0, & y_{0b} &= y_{0a} \oplus x_1 \end{aligned} \quad (6)$$

식 (6)과 같이 단순화 시킨 식에서 최소항에 해당하는 식으로 변환하면 식 (7)과 같다.

$$MDS(x_i, y_{ia}, y_{ib}) \quad (7)$$

식 (7)에서 $i = 0...7$의 범위를 갖는다. 이때 식 (7)은 MDS에 대한 곱셈 수행을 위한 최종 함수이며 이를 LUT에 고정할 경우, 기존 MDS보다 stage 연산이 감소되어 전체 처리시간의 감소를 가져오게 된다.

Ⅳ. 제안된 MDS를 사용한 Twofish 암호시스템 설계

식 (6)과 식 (7)을 이용하여 MDS를 설계하면 그림 2와 같다. 그림 2는 LUT 블록과 modulo-2 연산만으로 구성된 블록으로 곱셈연산만을 수행하게 된다. 식 (4)를 식 (6)과 같이 단순화 할 때 0x5B와 0xEF에 대한 항목은 식 (6)의 a, b항으로 구별된다. 입력 8 비트씩 4블록은 MDS의 상수항과 곱셈연산을 수행하게 되고 이에 대한 출력은 8 비트씩 4블록의 출력으로 산출된다.

그림 3은 Twofish 암호시스템에서 키 데이터를 생성하는 키 스케쥴러 블록이다. S-box들에 의한 S 함수와 g 함수 그리고 q 치환 기능을 수행하는 블록으로서 RS(Reed-Solomon) 곱셈에 의한 연산을 수행한다.

그림 4는 LUT와 modulo-2연산을 사용한 제안된 MDS를 Twofish 암호시스템에 적용하여 설계한 암호시스템의 전체 합성 회로이다. 입력데이터 128 비트, 입

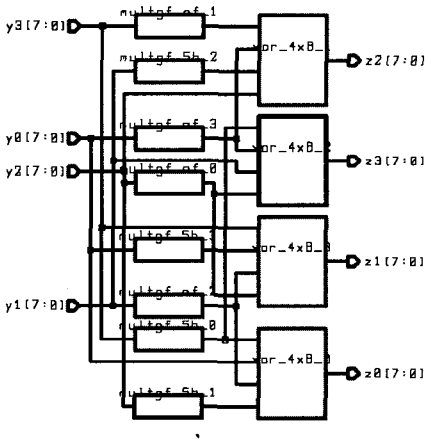

그림 2. 제안된 LUT와 modulo-2 연산을 적용한 MDS 블록

Fig. 2. Proposed MDS block using LUT and modulo-2.

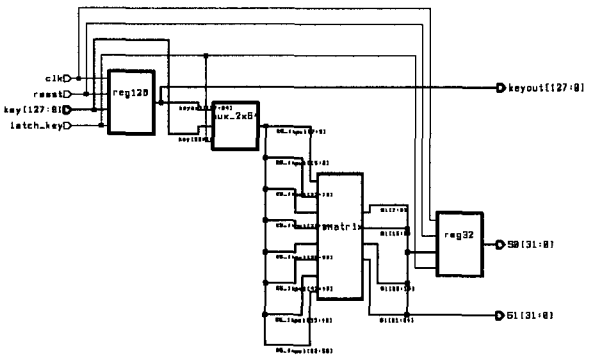

그림 3. 키 스케쥴러 블록

Fig. 3. Key scheduler block.

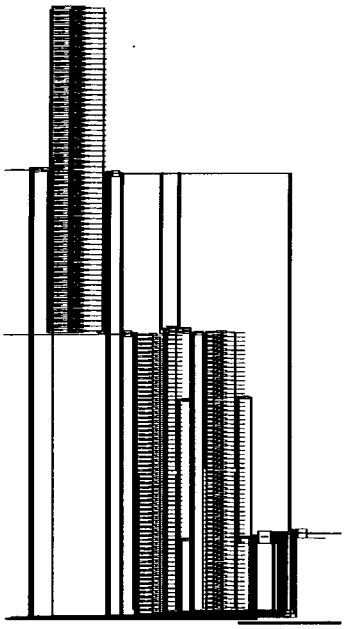

그림 4. Twofish 암호시스템 회로
Fig. 4. Twofish cryptosystem.

표 1. LUT-m2 MDS와 기존 MDS에 대한 Twofish 암호시스템 비교
Table 1. Comparison of existed and proposed MDS.

	LUT-m2 MDS	기존 MDS
Throughput	68Mbps	61Mbps
Cell counter (MDS)	112	101
Total counter	3,523	3,215

력 키는 128 비트이며 출력되는 암호데이터는 128 비트의 1:1 관계를 갖는 연산을 수행한다.

표 1은 LUT와 modulo-2연산을 사용한 제안된 MDS를 이용한 Twofish와 기존 Twofish 암호시스템들과의 비교표이다.

표 1에서 보는바와 같이 제안된 LUT와 modulo-2연산을 사용한 제안된 MDS 구조를 갖는 Twofish 암호시스템이 기존 암호시스템에 비하여 10% 정도 속도가 향상됨을 확인하였다.[5][7]

V. 결 론

본 논문에서는 Rijndael 암호알고리즘에 비하여 다소 속도가 떨어지지만 암/복호화기 동시에 가능하며 구현상의 용이성이 높은 Twofish 암호알고리즘의 속도를 향상시키기 위하여 LUT & modulo-2를 적용한 MDS를 설계하였다. 또한 이를 Twofish 암호알고리즘에 적용하여 Twofish 암호시스템을 Synopsys 1999.10과 MaxplusII 10.1을 이용하여 설계하였다. 설계된 암호시스템의 모의실험 결과 기존방식의 Twofish 암호시스템보다 처리속도 면에서 10% 처리속도의 향상을 보임을 확인하였다.

그러므로 본 논문에서 제안한 LUT와 modulo-2연산을 사용한 Twofish 암호시스템은 인터넷 보안기술을 요구하는 네트워크 환경에 매우 적합한 암호시스템이라 사료된다.

참 고 문 헌

[1] National Institute Standards & Technology, http://csrc. nist.gov/encryption/aes/
[2] Pawel Chodowice, Kris Gaj, "Implementation of the Twofish Cipher Using FPGA Devices", Technical Report, George Mason University, 1999.
[3] J. Daeman, V. Rijnmen, "AES Proposal: Rijndael, http://csrc.nist.gov/encryption/aes
[4] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson. Twofish : A 128-Bit Block Cipher, 1998.
[5] Third AES candidate conference, "AES3 Proceedings, http://csrc.nist.gov/encryption/aes /round2/conf3/ papers/", pp. 44-54, April, 2000.
[6] NIST, "Draft FIPS for the AES", http:// csrc.nist.gov/publications/drafts.html, Feb. 2001.
[7] Helion, "Rijndael core", http://www. heliontech. com/core2.htm

저 자 소 개

이 선 근(정회원)

제 41권 SD편 4호 참조

김 환 용(정회원)

제 42권 SD편 6호 참조