*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1295

# SEED암호에서 S-함수에 대한 고찰

양 정 모†‡
중부대학교 공과대학 정보보호학과

# A Study on S-Function in SEED Cryptosystem

Jeong-Mo Yang†‡
Dep. of information security, College in Engineering, Joongbu University

## 요 약

국내 블록암호표준(안)으로는 SEED암호가 있다. 이 암호는 한국정보보호진흥원(KISA)이 1998년 10월에 초안을 설계하고 같은 해 12월에 공개검증과정을 거쳐 안전성과 성능이 개선된 최종 수정안을 발표하였다. DES와는 달리 128비트 블록암호로서 여러 과정을 거쳐 2005년에 국제표준으로 확정되었다. DES와 같은 페이스텔 구조를 가진 블록암호로서 다만 입력비트블록이 DES의 두 배인 128비트로 늘어났다는 것이다. 본 논문에서는 첫째, SEED 암호의 일반적인 알고리즘을 소개하고 F-함수에서 적용되는 열쇠 값의 생성원리를 수학적으로 분석해 보았다. 둘째, S-함수의 8비트 입력 값에 대응되는 원시원소 $\alpha$의 멱승 값을 계산하는 표를 도출해 보았으며 마지막으로 G-함수 내에 설계되어져 있는 S-함수의 계산 원리를 수학적인 방법으로 새로운 정리와 예제를 통해 분석해 보는 것으로 한정하였다. 이러한 과정을 통하여 현재 알려져 있는 SEED암호의 취약점을 보완할 수 있는 새로운 암호체계를 개발하는데 필요한 아이디어와 이론적인 근거를 제공하는 데 어느 정도 도움이 되고자 한다.

## ABSTRACT

There is SEED cryptosystem in domestic block cipher standard. This code was drafted by the Korea Information Security Agency (KISA) in October 1998 and underwent a public verification process in December of the same year, which resulted in the final amendment to improve safety and performance.

Unlike DES, it is a 128-bit block cipher that has been passed through various processes and established in 2005 as an international standard. It is a block cipher with a pastel structure like DES, but the input bit block has been increased to 128 bits, double DES.

In this paper, first, we introduce the general algorithm of SEED cryptosystem and analyzed mathematically generating principle of key-value which is used in F-function. Secondly, we developed a table that calculates the exponent of the primitive element $\alpha$ corresponding to the 8-bit input value of the S-function and finally analyzed calculating principle of S-function designed in G-function through the new theorem and example. Through this course, we hope that it is to be suggest the ideas and background theory needed in developing new cryptosystem to cover the weakness of SEED cryptosystem

**Keywords:** primitive polynomial, F-function, G-function, S-function, Key-value

Received(09. 29. 2017), Modified(10. 26. 2017), Accepted(10. 30. 2017)

† 주저자, jmyang@jbm.ac.kr
‡ 교신저자, jmyang@jbm.ac.kr(Corresponding author)

# I. 서 론

## 1.1 기본 정의 및 정리

SEED암호 알고리즘을 이해하기 위해서 몇 가지 수학적인 용어에 대한 이해와 정리가 필요하다. 이를 소개하면 다음과 같다.

**정의 1.1.** 임의의 소수 $p$ 와 양의 정수 $n$ 에 대하여 갈루아 체(Galois field) $F_{p^n}$ 에서

$$F_{p^n}^* = F_{p^n} - \{0\} = <\alpha> = \{1, \alpha, \alpha^2, \cdots, \alpha^{p^n-2}\},\ \alpha^{p^n-1} = 1$$

인 원소 $\alpha \in F_{p^n}$ 을 체 $F_{p^n}$ 의 원시원소(primitive element)라 하고 $\alpha \in F_{p^n}$ 의 체 $F_p$ 에서의 최소다항식

$$p(x) = \min.poly._{F_{p^n}} \alpha\ ,\ p(x) \in F_p[x]$$

을 $F_p$ 위에서의 원시다항식(primitive polynomial)이라 한다[1].

일반적으로 체 $K$가 체 $F$의 확대체일 때 원소 $\alpha \in K$가 체 $F$위에서 대수적(원소)이면 즉, $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \in F[x]$ 에 대하여

$$p(\alpha) = a_0 + a_1\alpha + a_2\alpha + \cdots + a_n\alpha^n = 0$$

이면 다음 세 조건을 만족하는 유일한 다항식 $p(x) \in F[x]$ 을 체 $F$위에서의 $\alpha$ 의 최소다항식(minimum polynomial) 또는 기약다항식(irreducible polynomial)이라 하고 이를 $p(x) = \min.poly._F \alpha$ 또는 $p(x) = irr(\alpha, F)$ 로 나타낸다. 즉,

i) $\deg(p(x)) \geq 1$ 이고 $p(x)$ 의 최고차항의 계수는 1이다.
ii) $p(\alpha) = 0$
iii) 다항식 $f(x) \in F[x]$ 에 대하여 $F[x]$ 에서 $f(\alpha) = 0 \Leftrightarrow p(x) \mid f(x)$

위의 다항식 $p(x)$ 의 차수를 $\alpha$ 의 $F$위에서의 차수라 하고 이것을 $[\alpha : F] = \deg(p(x))$ 로 나타낸다. 위의 설명에 의하면 최소다항식 또는 기약다항식 중 특별한 경우가 바로 원시다항식이다[1].

**정리 1.2.** 임의의 소수 $p$ 와 양의 정수 $n$ 에 대하여 갈루아 체 $F_{p^n}$ 에는 $\varphi(p^n-1)$ 개의 원시원소가 존재하고 또한 체 $F_p$ 위의 $n$ 차의 원시다항식은 $\frac{\varphi(p^n-1)}{n}$ 개 존재한다[1].

**정의 1.3.** 체 $F_2$ 위의 $f(0) \neq 0$ 인 $n$ 차 다항식 $f(x)$ 에 대하여

$$f(x) \mid (x^e - 1),\quad 1 \leq e \leq 2^n - 1$$

인 가장 작은 양의 정수 $e$ 을 $f(x)$ 의 위수(order)라고 하고 이것을 $\mathrm{ord}(f(x)) = e$ 로 나타낸다[2].

**예 1.1.** 체 $F_2$ 위에 주어진 다음 다항식의 위수를 구하여라.

(1) $f(x) = x^2 + x + 1$
(2) $g(x) = x^4 + x^2 + 1$

풀이.

(1) $x^3 + 1 = (x+1)(x^2 + x + 1) = (x+1)f(x)$
이므로 $f(x) \mid (x^3 + 1)$ 이다. 따라서
$\mathrm{ord}(f(x)) = 3$
이다.

(2) $x^6 + 1 = (x+1)(x^4 + x^2 + 1) = (x+1)g(x)$
이므로 $g(x) \mid (x^6 + 1)$ 이다. 따라서
$\mathrm{ord}(g(x)) = 6$
이다.

다항식 $f(x)$ 가 기약일 때 다항식의 위수 $\mathrm{ord}(f(x))$ 와 대수적 차수 $\deg(f(x))$ 는 다음 정리와 같은 관계가 있다.

**정리 1.4.** 체 $F_2$ 위의 다항식 $f(x) \in F_2[x]$ 에 대하여

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} + x^n$$

가 기약다항식이고 $f(0) \neq 0$ 일 때 $\alpha$ 을 체 $F_{2^n}$ 에서의 $f(x)$ 의 근 이라고 하면 다음이 성립한다.

(1) $\mathrm{ord}(f(x))$ 는 곱셈군 $F_{2^n}^*$ 에서의 $\alpha$ 의 위수와 같다.

(2) $\mathrm{ord}(f(x)) \mid (2^n-1)$

(3) 영이 아닌 수열 $\{s_t\} \in \Omega(f(x))$ 는 주기가 모두 $\mathrm{ord}(f(x))$ 인 순환이진 수열이다[1].

특히 위의 정리에 의하면 다항식 $f(x)$ 가 $n$ 차 원시다항식이면 정의 1.1에서 $p=2$ 인 경우이므로 정리 1.4의 (1)에 의하여 $\mathrm{ord}(f(x)) = 2^n-1$ 이다.

**정의 1.5.** 체 $F_2$ 위의 $n$ 차 원시다항식 $f(x)$ 에 의하여 생성되는 영이 아닌 수열 $\{s_t\} \in \Omega(f(x))$ 을 주기가 $2^n-1$ 인 최대주기수열(maximal length sequence) 또는 PN수열(pseudo-noise sequence)이라 한다[2].

체 $F_2$ 위의 다항식 $f(x) \in F_2[x]$ 에 대하여

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} + x^n$$

가 기약다항식이고 $\mathrm{ord}(f(x)) = e$ 일 때 $e = 2^n-1$ 인 원시다항식은 $n=6$ 일 때 모두 16개가 있다. 즉, $f(x)$ 가 6차인 원시다항식은

$x^6+x+1,\ x^6+x^4+x^3+x+1,\ x^6+x^5+1$

$x^6+x^5+x^2+x+1,\quad x^6+x^5+x^3+x^2+1,$

$x^6+x^5+x^4+x+1$

이고 주기는 모두가 63이다. 원시다항식은 아니지만 6차 기약다항식은

$$x^6+x^5+x^4+x^2+1$$

과 같은 주기가 21인 다항식이 있다. 실제로 $n$ 차 원시다항식의 개수 $\lambda(n)$ 는

$$\lambda(n) = \frac{\varphi(2^n-1)}{n}$$

이며 $n$ 차 기약다항식은 이보다 더 많다. 여기서 $\varphi$ 는 오일러함수이다. $n$ 의 값에 따라 그 개수를 나타내면 다음 Table 1과 같다[3].

예를 들면 $n=8$ 인 경우 $\lambda(8) = \frac{\varphi(2^8-1)}{8}$ 이 되고 여기서

$$\varphi(2^8-1) = \varphi(255) = \varphi(3\times 5\times 17)$$
$$= \varphi(3)\cdot\varphi(5)\cdot\varphi(17) = 2\times 4\times 16 = 128$$

이 되어 $\lambda(8) = 16$ 이 된다. 8차 기약다항식은 원시다항식 16개를 포함하여 이보다 더 많다.

Table 1. primitive polynomial number $\lambda(n)$ of degree $n$

| $n$ | $\lambda(n)$ | $n$ | $\lambda(n)$ |
|---|---|---|---|
| 1 | 1 | 11 | 176 |
| 2 | 1 | 12 | 144 |
| 3 | 2 | 13 | 630 |
| 4 | 2 | 14 | 756 |
| 5 | 6 | 15 | 1800 |
| 6 | 6 | 16 | 2048 |
| 7 | 18 | 17 | 7710 |
| 8 | 16 | 18 | 8064 |
| 9 | 48 | 19 | 27594 |
| 10 | 60 | 20 | 24000 |

## 1.2 SEED암호 알고리즘

알려진 SEED알고리즘의 전체 구조는 변형된 페이스텔 구조로 이루어져 있으며 128비트의 평문입력 블록을 128비트 열쇠로부터 생성된 16개의 64비트의 회전열쇠를 사용하여 128비트의 암호문을 출력한다. 이 알고리즘의 전체 구조는 블록의 크기만 다를 뿐 DES의 구조와 같으며 자세한 구조는 Fig. 1과

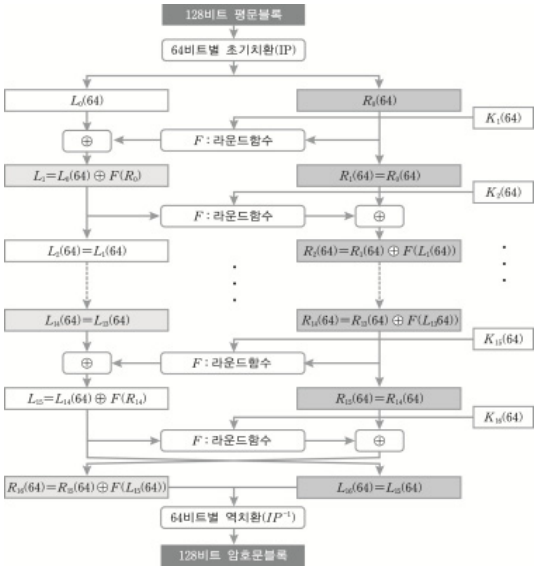


Fig. 1. The structure of SEED cryptosystem

같다.

**〔암호화과정〕**

1. 128비트의 평문블록을 반으로 나누어 64비트 블록을 $L_0$ 와 $R_0$ 라 하자.

2. DES와 같은 과정을 거쳐 16회전을 하여 최종 출력 128비트 암호문 $(L_{16}, R_{16})$ 을 얻는다.

즉, $0 \le k \le 7$ 에 대하여

$L_{2k+1}(64) = L_{2k}(64) \oplus F(R_{2k})$, $R_{2k+1}(64) = R_{2k}(64)$,

$L_{2k+2}(64) = L_{2k+1}(64)$,

$R_{2k+2}(64) = R_{2k+1}(64) \oplus F(L_{2k+1}(64))$

이다. 단, 2의 공식에 의하여 마지막 블록인 $R_{16}$ 과 $L_{16}$ 을 구하되 마지막 라운드에서만 $R_{16}$ 은 왼쪽블록에 위치하고 $L_{16}$ 은 오른쪽 블록에 위치한다.

### 1.2.1 $F$-함수 구조

Fig. 1에서 사용된 $F$-함수는 64비트의 열쇠를 이용하여 입력된 64비트를 64비트로 출력하는 함수로서 $i\,(1 \le i \le 16)$ 회전에서의 구조는 Fig. 2와 같다. 즉, 64비트를 32비트의 2개의 블록 $(C, D)$ 로 나누어 입력받아서 32비트의 2개의 블록 $(C', D')$ 을 출력하는 함수이다. 여기서, $K_i = (K_{i,0}, K_{i,1})$ 는 64비트 $i$ 회전 열쇠이고 연산 $\boxplus$는 $a \boxplus b \equiv a + b \pmod{32}$ 로 정의한다.

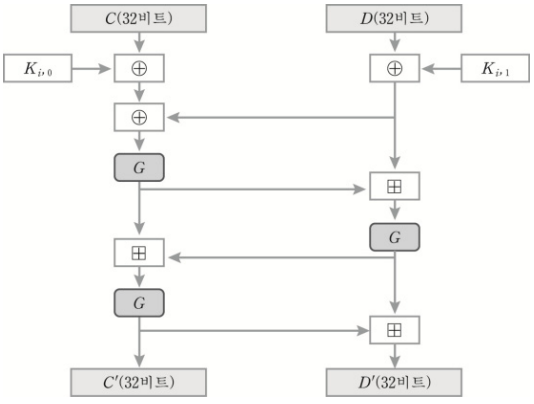


Fig. 2. The structure of F-function

### 1.2.2 $G-$ 함수 적용순서

Fig.2의 $F$-함수구조에서 $G$는 아래 소개될 두 개의 $S-$ 함수를 이용하여 32비트를 32비트로 보내는 함수이다. $G-$ 함수를 적용하는 순서는 다음과 같다(Fig. 3).

(1) 32비트 입력블록 $X$을 4개의 8비트 블록 $X=(X_3, X_2, X_1, X_0)$ 으로 비트 순서대로 분할하여 2개의 $S-$ 함수 $S_1, S_2$ 을

$$S=(S_2, S_1, S_2, S_1)$$

순서로 적용시켜

$$Y=S(X)=\{S_2(X_3), S_1(X_2), S_2(X_1), S_1(X_0)\}$$
$$=(Y_3, Y_2, Y_1, Y_0) \text{ 을 얻는다.}$$

(2) 4 개의 확장된 32비트 $S-$ 함수

$$SS_0 = Y_0 \otimes m_3 \,\|\, Y_0 \otimes m_2 \,\|\, Y_0 \otimes m_1 \,\|\, Y_0 \otimes m_0$$
$$SS_1 = Y_1 \otimes m_0 \,\|\, Y_1 \otimes m_3 \,\|\, Y_1 \otimes m_2 \,\|\, Y_1 \otimes m_1$$
$$SS_2 = Y_2 \otimes m_1 \,\|\, Y_2 \otimes m_0 \,\|\, Y_2 \otimes m_3 \,\|\, Y_2 \otimes m_2$$
$$SS_3 = Y_3 \otimes m_2 \,\|\, Y_3 \otimes m_1 \,\|\, Y_3 \otimes m_0 \,\|\, Y_3 \otimes m_3$$

들을 XOR하여 $Z=(Z_3, Z_2, Z_1, Z_0)$ 을 구한다. 즉,

$Z= SS_0(X_0) \oplus SS_1(X_1) \oplus SS_2(X_2) \oplus SS_3(X_3)$

이고 비트 순서대로 XOR하기 때문에

$Z_3 = (Y_0 \otimes m_3) \oplus (Y_1 \otimes m_0) \oplus (Y_2 \otimes m_1) \oplus (Y_3 \otimes m_2)$

$Z_2 = (Y_0 \otimes m_2) \oplus (Y_1 \otimes m_3) \oplus (Y_2 \otimes m_0) \oplus (Y_3 \otimes m_1)$

$Z_1 = (Y_0 \otimes m_1) \oplus (Y_1 \otimes m_2) \oplus (Y_2 \otimes m_3) \oplus (Y_3 \otimes m_0)$

$Z_0 = (Y_0 \otimes m_0) \oplus (Y_1 \otimes m_1) \oplus (Y_2 \otimes m_2) \oplus (Y_3 \otimes m_3)$

가 된다. 단, 여기서

$$m_0 = 0xfc = 11111100 \ ,$$
$$m_1 = 0xf3 = 11110011 \ ,$$
$$m_2 = 0xcf = 11001111 \ ,$$
$$m_3 = 0x3f = 00111111$$

이고 $0x$ 는 그 뒤에 나오는 수가 16진수임을 나타낸다. 또, $\otimes$ 은 비트단위 곱 연산이고 $\|$ 는 연접이다.

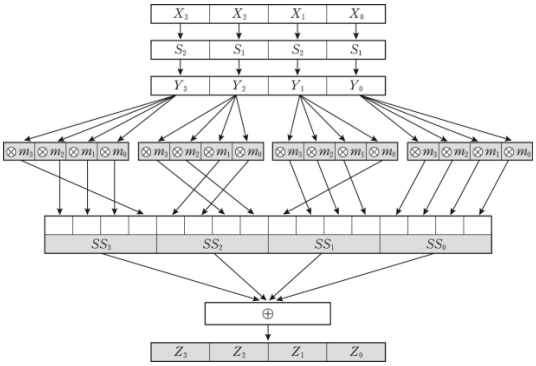

Fig. 3. The structure of G-function

### 1.2.3 $S-$ 함수

$G-$ 함수에서 사용되는 두 개의 $S-$ 함수는 모두 부울함수(Boolean function)를 사용하고 있다. 즉, $S-$ 함수는 전단사함수 $x^n$ 의 선형변환

$$S(x) = A \cdot x^n \oplus b \quad (S(0) \neq 0,\ S(1) \neq 1)$$

으로 사용한다. 단,

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

이고 $x = (x_7,\ x_6,\ x_5,\ x_4,\ x_3,\ x_2,\ x_1,\ x_0)^t$

$$= x_7 2^7 + x_6 2^6 + \cdots + x_1 2^1 + x_0 2^0$$

값으로 약속한다.

이를 근거로 $S-$ 함수 $S_1$ 과 $S_2$ 을 다음과 같이 정의한다.

**정의 1.6[ $S-$ 함수 $S_1$ 과 $S_2$ ]**

위에서 지수 $n$ 은 전단사함수 $x^n$ $(0 \leq n \leq 255)$ 에서 $DC$ 및 $LC$ 의 특성에 가장 우수한 2 개의 $n$ 값으로 247과 251을 선택하고 체 $GF(2^8)$ 위에서의 지수 승을 구하기 위해 이 위의 모든 원소를 원시원소 $\alpha$ (원시다항식 $x^8 + x^6 + x^5 + x + 1$ 의 근)의 멱승으로 표현한다. 이 때 두 개의 $S-$ 함수 $S_1$ 과 $S_2$ 을 다음과 같이 정의한다. 즉,

$$S_1(x) = A^{(1)} \cdot x^{247} \oplus b_1$$
$$= (P_7,\ P_6,\ P_5,\ P_4,\ P_3,\ P_2,\ P_1,\ P_0)^t$$
$$= P_7 2^7 + P_6 2^6 + P_5 2^5 + \cdots + P_2 2^2 + P_1 2^1 + P_0$$
$$S_2(x) = A^{(2)} \cdot x^{251} \oplus b_2$$
$$= (Q_7,\ Q_6,\ Q_5,\ Q_4,\ Q_3,\ Q_2,\ Q_1,\ Q_0)^t$$
$$= Q_7 2^7 + Q_6 2^6 + Q_5 2^5 + \cdots + Q_2 2^2 + Q_1 2^1 + Q_0$$

이다. 단, 행렬 $A^{(1)}$ 은 행렬 $A$ 의 2행과 3행, 4행과 6행을 교환한 행렬이고 행렬 $A^{(2)}$ 은 행렬 $A$ 의 1행과 5행, 6행과 7행을 교환한 행렬이다. 또한

$$b_1 = (1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1)^t = 169,$$
$$b_2 = (0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0)^t = 56$$

으로 약속한다.

위의 정의에 의하면 행렬 $A^{(1)}$ 과 $A^{(2)}$ 는 다음과 같다. 즉,

$$A^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

이다.

## II. 열쇠생성 알고리즘

SEED암호에 사용되는 회전열쇠 생성과정은 다음과 같다.

**[회전열쇠 생성과정]**

1단계 : 128비트의 열쇠를 4개의 워드(32비트) $(A,\ B,\ C,\ D)$ 로 나눈다.

2단계:

$K_{1,0} = G(A + C - KC_0),\ K_{1,1} = G(B - D + KC_0)$

을 계산하여 1라운드 열쇠를 생성한다.

여기서 $KC_0$ 은 회전상수이다.

3단계: $B \parallel A = (B \parallel A) \gg 8$ 이동하여 새로운 $(A,\ B,\ C,\ D)$ 을 구성한다. 여기서

$B \parallel A = (B \parallel A) \gg 8$ 은 $B \parallel A$ 을 오른쪽으로 8비트 회전시킨 64비트이다.

4단계 :

$K_{2,0} = G(A + C - KC_1),\ K_{2,1} = G(B - D + KC_1)$

을 계산하여 2라운드 열쇠를 생성한다. 여기서 $KC_1$ 은 회전상수이다.

5단계 : $D \parallel C = (D \parallel C) \ll 8$ 이동하여 새로운 $(A,\ B,\ C,\ D)$ 을 구성한다. 여기서

$D \parallel C = (D \parallel C) \ll 8$ 은 $D \parallel C$ 을 왼쪽으로 8비트 회전시킨 64비트이다.

6단계 : 위의 2단계에서 5단계의 과정을 16회전 열쇠를 생성할 때까지 반복한다.

단, $KC_i$ 는 황금비의 소수부분으로부터 생성된 회전상수이며 다음과 같이 정의한다. 즉,

$$KC_0 = int(\frac{\sqrt{5}-1}{2} \times 2^{32}) = 0x9e3779b9$$

$$KC_i = KC_{i-1} \ll 1 \quad (1 \le i \le 16)$$

**[수학적 분석]**

SEED암호에서 설계한 열쇠는 순환하지 않는 황금비와 관련된 무리수를 선택하여 설계하였다. 즉, 위에서 설계된 $int\left(\frac{\sqrt{5}-1}{2} \times 2^{32}\right)$ 은 무리수 $\frac{\sqrt{5}-1}{2}$ 을 2진수로 나타낸 것 중 32비트 정수부분만 나타낸다는 의미이다. 즉,

$$\frac{\sqrt{5}-1}{2} = 0.6180339887498948\cdots$$

$$= 0.10011110001101110111111001101111001\cdots_{(2)}$$

이므로

$$\frac{\sqrt{5}-1}{2} \times 2^{32}$$

$$= 0.10011110001101110111111001101111001\cdots_{(2)} \times 2^{32}$$

$$= 10011110001101110111110011011110.01\cdots_{(2)}$$

이 되어 정수부분만 나타내면

$$int\left(\frac{\sqrt{5}-1}{2} \times 2^{32}\right)$$

$$= 10011110001101110111110011011110_{(2)}$$

이다. 이를 16진수로 나타내면

$KC_0 = int(\frac{\sqrt{5}-1}{2} \times 2^{32}) = 0x9e3779b9$ 이다.

6단계에서 정의한 회전상수 $KC_i$ 을 구체적으로 나열하면 다음 Table 2와 같다.

Table 2. Round constant $KC_i$

| | |
|---|---|
| $KC_0 = 0x9e3779b9$ | $KC_8 = 0x3779b99e$ |
| $KC_1 = 0x3c6ef373$ | $KC_9 = 0x6ef3733c$ |
| $KC_2 = 0x78dde6e6$ | $KC_{10} = 0xdde6e678$ |
| $KC_3 = 0xf1bbcdcc$ | $KC_{11} = 0xbbcdccf1$ |
| $KC_4 = 0xe3779b99$ | $KC_{12} = 0x779b99e3$ |
| $KC_5 = 0xc6ef3733$ | $KC_{13} = 0xef3733c6$ |
| $KC_6 = 0x8dde6e67$ | $KC_{14} = 0xde6e678d$ |
| $KC_7 = 0x1bbcdccf$ | $KC_{15} = 0xbcdccf1b$ |

## III. 주요 정리와 예제

**〔수학적 분석〕**

SEED암호에서 $S-$ 함수를 계산하는 과정에서 8비트 입력 벡터 $x$ 에 대하여 이를 10진수로 변환한 값을 $i$ 라 했을 때 이에 대응하는 $\alpha^n$ 의 $n$ 값을 알아야 S-함수의 결과 값을 도출할 수 있다. 먼저, $S-$ 함수를 계산하는 전 단계인 8비트 입력벡터를 10진수 $i$ 로 변환하는 방법에 대하여 살펴본다.

8비트 입력벡터의 경우의 수는 모두 $2^8 = 256$ 가지이므로 이 입력벡터는 다음과 같이 구한다. 즉, 8비트 입력 벡터 $x \in GF(2^8)$ 에 대하여 함수 $T$ 을 다음과 같이 정의하여 $i$ 값을 계산한다.

$$T:\ GF(2^8) \rightarrow i\ ,$$

$$T(x) = (a_7, a_6, \cdots, a_1, a_0)^t$$
$$= a_7 2^7 + a_6 2^6 + \cdots a_1 2^1 + a_0 = i,$$
$$x = a_7\alpha^7 + \cdots + a_1\alpha^1 + a_0 \in GF(2^8)$$

단, 체 $GF(2^8) = F_2(\alpha)$

$$= \{a_7\alpha^7 + \cdots + a_1\alpha^1 + a_0 \mid a_7, a_6, \cdots, a_1, a_0 \in F_2\}$$
$$= \{0, 1, \alpha, \alpha^2, \cdots, \alpha^{254}\},$$
$$GF^*(2^8) = GF(2^8) - \{0\} = <\alpha>$$
$$= \{1, \alpha, \alpha^2, \cdots, \alpha^{254}\}, \alpha^{255} = 1,$$
$$\alpha^8 = \alpha^6 + \alpha^5 + \alpha + 1$$

이다. 결론적으로 체 $F_2$ 위에서 정의된 갈로아체 $GF(2^8)$ 의 모든 원소 $x = \alpha^n$ ($0 \le n \le 254$) 에 대하여

$$\alpha^n = a_7\alpha^7 + a_6\alpha^6 + \cdots + a_1\alpha^1 + a_0$$

로 나타내었을 때 이에 대응하는 계수들을 순서대로 나열한 8비트 $(a_7 a_6 \cdots a_1 a_0)$ 을 10진수로 나타낸 것이 $i$ 값이 된다(표 6). 이 표는 SEED암호에서 $S-$ 함수를 계산하는 데 유용하다.

**예 3.1** 체 $F_2$ 위에서의 8차 원시다항식

$$f(x) = x^8 + x^6 + x^5 + x + 1$$

에 대하여 이 다항식의 원시원소를 $\alpha \in F_{2^8}$ 라 하면

$$f(\alpha) = \alpha^8 + \alpha^6 + \alpha^5 + \alpha + 1 = 0,$$
$$\alpha^8 = \alpha^6 + \alpha^5 + \alpha + 1$$

이고

$$F_{2^8} - \{0\} = <\alpha> = \{1, \alpha, \alpha^2, \alpha^3, \cdots, \alpha^{254}\},$$
$$\alpha^{255} = 1$$

이다. 원시원소 $\alpha$ 의 멱승 값 $n$ 에 대응되는 이진벡터를 구하기 위해 $\alpha$ 의 멱승 값을 표현해 보면

$$\alpha^1 = 0\cdot\alpha^7 + 0\cdot\alpha^6 + 0\cdot\alpha^5 + 0\cdot\alpha^4$$
$$+ 0\cdot\alpha^3 + 0\cdot\alpha^2 + 1\cdot\alpha^1 + 0$$
$$\alpha^2 = 0\cdot\alpha^7 + 0\cdot\alpha^6 + 0\cdot\alpha^5 + 0\cdot\alpha^4$$
$$+ 0\cdot\alpha^3 + 1\cdot\alpha^2 + 0\cdot\alpha^1 + 0$$

등으로 나타낼 수 있으므로 $\alpha^1$ 의 1에 대응되는 이진벡터는 $(0,0,0,0,0,0,1,0)$ 이고 $\alpha^2$ 의 2에 대응되는 이진벡터는 $(0,0,0,0,0,1,0,0)$ 가 된다. 이를 10진법으로 표현하면 $\alpha^1$ 의 1에 대응하는 $i = 2$ 이고 $\alpha^2$ 의 2에 대응하는 $i = 4$ 가 된다. 이렇게 하면 $\alpha^7$ 의 7에 대응되는 이진벡터는 $(1,0,0,0,0,0,0,0)$ 이고 $\alpha^8$ 의 8에 대응되는 이진벡터는 $\alpha^8 = \alpha^6 + \alpha^5 + \alpha + 1$ 이므로

$$(0,1,0,0,0,0,0,0) + (0,0,1,0,0,0,0,0)$$
$$+ (0,0,0,0,0,0,1,0) + (0,0,0,0,0,0,0,1)$$
$$= (0,1,1,0,0,0,1,1)$$

와 같이 계산한다. 같은 방법으로

$$\alpha^9 = \alpha(\alpha^8) = \alpha(\alpha^6 + \alpha^5 + \alpha + 1) = \alpha^7 + \alpha^6 + \alpha^2 + \alpha$$

가 되어 이진벡터는 $\alpha^8$ 의 8에 대응되는 이진벡터의 성분을 왼쪽으로 1비트씩 이동하면 된다. 또

$$\alpha^{10} = \alpha(\alpha^9) = \alpha(\alpha^7 + \alpha^6 + \alpha^2 + \alpha) = \alpha^8 + \alpha^7 + \alpha^3 + \alpha^2$$
$$= (\alpha^6 + \alpha^5 + \alpha + 1) + \alpha^7 + \alpha^3 + \alpha^2$$
$$= \alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1$$

가 되어 $\alpha^{10}$ 의 10에 대응하는 이진벡터는 $\alpha^9$ 의 9에 대응되는 이진벡터를 좌측으로 1비트씩 이동하고 8번째 비트는 버리고 대신 7번째, 6번째, 두 번째, 첫 번째 성분에 모두 1을 더하면 된다.

다음 정리와 예제는 이 논문의 주요 정리와 예로서 $S-$ 함수를 계산하는 방법을 수학적으로 제시한다.

**정리 3.1** 8비트 입력블록 $X = (x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0)$ 일 때 정의 1.6에서 주어진 조건에 대하여

$$S_1(x) = A^{(1)} \cdot x^{247} \oplus b_1$$
$$= (P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0)^t$$
$$= P_7 2^7 + P_6 2^6 + P_5 2^5 + \cdots + P_2 2^2 + P_1 2^1 + P_0$$
$$S_2(x) = A^{(2)} \cdot x^{251} \oplus b_2$$
$$= (Q_7, Q_6, Q_5, Q_4, Q_3, Q_2, Q_1, Q_0)^t$$
$$= Q_7 2^7 + Q_6 2^6 + Q_5 2^5 + \cdots + Q_2 2^2 + Q_1 2^1 + Q_0$$

에 의해 계산되어진다.

**증명.** 주어진 입력블록을 $X=(x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0)$ 라 하자. 이 때 주어진 8비트 이진벡터를 다음과 같이 10진수로 계산한다. 즉, 정의 1.6에 의하여

$$x = (x_7, x_6, \cdots, x_2, x_1, x_0)^t$$
$$= x_7 \times 2^7 + x_6 \times 2^6 + \cdots + x_2 \times 2^2 + x_1 \times 2^1 + x_0$$

이다. 이 10진수로 나타낸 값을 $i$ 라 하면 이에 대응하는 원시원소 $\alpha$ 의 멱승 값을 $n$ 이라 하자. 그러면

$$S_1(i) = A^{(1)} \cdot i^{247} \oplus b_1 = A^{(1)} \cdot (\alpha^n)^{247} \oplus b_1$$
$$= A^{(1)} \cdot \alpha^{247n} \oplus b_1$$
$$= A^{(1)} \cdot \alpha^{255m+k} \oplus b_1 ,$$
$$(247n = 255m + k,\ 0 \le k \le 254 \quad )$$
$$= A^{(1)} \cdot \alpha^k \oplus b_1 , \ (\because \alpha^{255} = 1)$$

이다. 여기에서 $\alpha^k$ 의 $k$ 에 대응되는 $i = c$ 이라 하고 이를 Table 6에서 찾아 이진벡터로 나타낸 것을 $c = (c_7, c_6, \cdots, c_2, c_1, c_0)$ 라 하면

$$S_1(i) = \begin{pmatrix} 1&0&0&0&1&0&1&0 \\ 1&1&1&1&1&1&1&0 \\ 1&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&1&0 \\ 0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \end{pmatrix} \begin{pmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$
$$= (P_7, P_6, \cdots, P_2, P_1, P_0)^t$$
$$= P_7 \times 2^7 + P_6 \times 2^6 + \cdots + P_2 \times 2^2 + P_1 \times 2^1 + P_0$$
$$= y$$

단, $P_7 = c_7 + c_3 + c_1 + 1$,
$P_6 = c_7 + c_6 + \cdots + c_2 + c_1$,
$P_5 = c_7 + c_2 + c_0 + 1$, $P_4 = c_6 + c_1$,
$P_3 = c_6 + c_2 + c_0 + 1$, $P_2 = c_5 + c_0$, $P_1 = c_7 + c_3$ ,
$P_0 = c_4 + c_2 + 1$ , $P_i \pmod 2 \ (0 \le i \le 7)$

이다. 같은 방법으로 $S_2(x)$ 도 구할 수 있다. □

**예 3.2** 8비트 입력블록 $X = (10010011)$ 일 때 $S_1(X)$ 을 구하여라.

**풀이.** $x = (1, 0, 0, 1, 0, 0, 1, 1)^t$
$= 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^1 + 1 = 147$

이므로 입력 값 $x = 147$에 대응되는 원시원소 $\alpha$의 멱승으로 나타내면 $\alpha^{128} = 147 = i$(Table 6 참조)이므로 147에 대응하는 원시근 $\alpha$의 멱승 값 $n = 128$ 이다. 따라서

$$S_1(147) = S_1(\alpha^{128}) = A^{(1)}(\alpha^{128})^{247} \oplus b_1 = A^{(1)}\alpha^{31616} \oplus b_1$$
$$= A^{(1)} \cdot (\alpha^{255})^{123} \cdot \alpha^{251} \oplus b_1 = A^{(1)} \cdot \alpha^{251} \oplus b_1 .$$
$$(\because \alpha^{255} = 1)$$

한편 $\alpha^{251}$의 251에 대응되는 $i = 211$이고 이를 이진벡터로 나타내면 (11010011)이다. 따라서

$$S_1(147) = \begin{pmatrix} 1&0&0&0&1&0&1&0 \\ 1&1&1&1&1&1&1&0 \\ 1&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&1&0 \\ 0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$
$$= (1, 0, 1, 0, 1, 1, 1, 0)^t$$
$$= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 = 174$$

이다. 따라서

$$S_1(X) = S_1(147) = 174 = (10101110) = Y$$

이다.

실제로 참고문헌 [2]의 표 3.27(p. 147)에 의하

면 $S_1(147) = 174$이다.

## IV. 맺음말

본 논문에서는 SEED암호에서 사용되는 열쇠 값의 생성원리와 S-함수의 계산 원리를 수학적으로 분석해 보았다. 또한 S-함수 계산을 원활하게 할 수 있도록 도움을 주는 표를 고안해 보았다(참고문헌 [2]의 표 3.27의 결과). 이 논문의 목적은 지금까지 알려져 있는 SEED암호의 핵심인 열쇠 값 생성과정과 S-함수의 결과 값을 도출해 내는 방법을 수학적인 관점에서 분석하는 것으로 한정하였다. 나머지 효율성과 안전도 그리고 경제적인 측면 등은 보안 구현 전문가의 몫으로 돌리겠다. 향 후 이를 통해 이 암호체계의 취약점을 보완할 수 있는 새로운 암호체계를 개발하거나 새로운 암호시스템을 설계하는 데 필요한 이론적 배경과 아이디어를 제공하는데 도움이 되었으면 하는 것이다.

## References


[1] Seung-an Park, "Algebra and Cryptography," Kyeongmoon Press, pp. 210-218, July, 2000

[2] Jeong-mo Yang, "Cryptography," *Kyeongmoon Press*, pp. 140-149, Mar. 2013

[3] Min-surp Lee, "Morden Cryptography," Kyowoo *Press*, pp. 143-150, 212-213, Mar. 2007

[4] Rainer A. Rueppel, "Analysis and Design of Stream Ciphers", Springer-Verlag, pp. 26-30, 1986

[5] Jennifer Seberry, Josef Pieprzyk, "Cryptography(An Introduction to Computer Security), Prentice Hall, pp. 7-44, 1989

Table 3. value $i$ corresponding power value $n$ of $\alpha$

| $n$ of $\alpha^n$ | binary vector | $i$ | $n$ of $\alpha^n$ | binary vector | $i$ |
|---|---|---|---|---|---|
| 0 | 00000001 | 1 | 64 | 11100001 | 225 |
| 1 | 00000010 | 2 | 65 | 10100001 | 161 |
| 2 | 00000100 | 4 | 66 | 00100001 | 33 |
| 3 | 00001000 | 8 | 67 | 01000010 | 66 |
| 4 | 00010000 | 16 | 68 | 10000100 | 132 |
| 5 | 00100000 | 32 | 69 | 01101011 | 107 |
| 6 | 01000000 | 64 | 70 | 11010110 | 214 |
| 7 | 10000000 | 128 | 71 | 11001111 | 207 |
| 8 | 01100011 | 99 | 72 | 11111101 | 253 |
| 9 | 11000110 | 198 | 73 | 10011001 | 153 |
| 10 | 11101111 | 239 | 74 | 01010001 | 81 |
| 11 | 10111101 | 189 | 75 | 10100010 | 162 |
| 12 | 00011001 | 25 | 76 | 00100111 | 39 |
| 13 | 00110010 | 50 | 77 | 01001110 | 78 |
| 14 | 01100100 | 100 | 78 | 10011100 | 156 |
| 15 | 11001000 | 200 | 79 | 01011011 | 91 |
| 16 | 11110011 | 243 | 80 | 10110110 | 182 |
| 17 | 10000101 | 133 | 81 | 00001111 | 15 |
| 18 | 01101001 | 105 | 82 | 00011110 | 30 |
| 19 | 11010010 | 210 | 83 | 00111100 | 60 |
| 20 | 11000111 | 199 | 84 | 01111000 | 120 |
| 21 | 11101101 | 237 | 85 | 11110000 | 240 |
| 22 | 10111001 | 185 | 86 | 10000011 | 131 |
| 23 | 00010001 | 17 | 87 | 01100101 | 101 |
| 24 | 00100010 | 34 | 88 | 11001010 | 202 |
| 25 | 01000100 | 68 | 89 | 11110111 | 247 |
| 26 | 10001000 | 136 | 90 | 10001101 | 141 |
| 27 | 01110011 | 115 | 91 | 01111001 | 121 |
| 28 | 11100110 | 230 | 92 | 11110010 | 242 |
| 29 | 10101111 | 175 | 93 | 10000111 | 135 |
| 30 | 00111101 | 61 | 94 | 01101101 | 109 |
| 31 | 01111010 | 122 | 95 | 11011010 | 218 |
| 32 | 11110100 | 244 | 96 | 11010111 | 215 |
| 33 | 10001011 | 139 | 97 | 11001101 | 205 |
| 34 | 01110101 | 117 | 98 | 11111001 | 249 |
| 35 | 11101010 | 234 | 99 | 10010001 | 145 |
| 36 | 10110111 | 183 | 100 | 01000001 | 65 |
| 37 | 00001101 | 13 | 101 | 10000010 | 130 |
| 38 | 00011010 | 26 | 102 | 01100111 | 103 |
| 39 | 00110100 | 52 | 103 | 11001110 | 206 |
| 40 | 01101000 | 104 | 104 | 11111111 | 255 |
| 41 | 11010000 | 208 | 105 | 10011101 | 157 |
| 42 | 11000011 | 195 | 106 | 01011001 | 89 |
| 43 | 11100101 | 229 | 107 | 10110010 | 178 |
| 44 | 10101001 | 169 | 108 | 00000111 | 7 |
| 45 | 00110001 | 49 | 109 | 00001110 | 14 |
| 46 | 01100010 | 98 | 110 | 00011100 | 28 |
| 47 | 11000100 | 196 | 111 | 00111000 | 56 |
| 48 | 11101011 | 235 | 112 | 01110000 | 112 |
| 49 | 10110101 | 181 | 113 | 11100000 | 224 |
| 50 | 00001001 | 9 | 114 | 10100011 | 163 |
| 51 | 00010010 | 18 | 115 | 00100101 | 37 |
| 52 | 00100100 | 36 | 116 | 01001010 | 74 |
| 53 | 01001000 | 72 | 117 | 10010100 | 148 |
| 54 | 10010000 | 144 | 118 | 01001011 | 75 |
| 55 | 01000011 | 67 | 119 | 10010110 | 150 |
| 56 | 10000110 | 134 | 120 | 01001111 | 79 |
| 57 | 01101111 | 111 | 121 | 10011110 | 158 |
| 58 | 11011110 | 222 | 122 | 01011111 | 95 |
| 59 | 11011111 | 223 | 123 | 10111110 | 190 |
| 60 | 11011101 | 221 | 124 | 00011111 | 31 |
| 61 | 11011001 | 217 | 125 | 00111110 | 62 |
| 62 | 11010001 | 209 | 126 | 01111100 | 124 |
| 63 | 11000001 | 193 | 127 | 11111000 | 248 |

| $n$ of $\alpha^n$ | binary vector | $i$ | $n$ of $\alpha^n$ | binary vector | $i$ |
|---|---|---|---|---|---|
| 128 | 10010011 | 147 | 192 | 00001011 | 11 |
| 129 | 01000101 | 69 | 193 | 00010110 | 22 |
| 130 | 10001010 | 138 | 194 | 00101100 | 44 |
| 131 | 01110111 | 119 | 195 | 01011000 | 88 |
| 132 | 11101110 | 238 | 196 | 10110000 | 176 |
| 133 | 10111111 | 191 | 197 | 00000011 | 3 |
| 134 | 00011101 | 29 | 198 | 00000110 | 6 |
| 135 | 00111010 | 58 | 199 | 00001100 | 12 |
| 136 | 01110100 | 116 | 200 | 00011000 | 24 |
| 137 | 11101000 | 232 | 201 | 00110000 | 48 |
| 138 | 10110011 | 179 | 202 | 01100000 | 96 |
| 139 | 00000101 | 5 | 203 | 11000000 | 192 |
| 140 | 00001010 | 10 | 204 | 11100011 | 227 |
| 141 | 00010100 | 20 | 205 | 10100101 | 165 |
| 142 | 00101000 | 40 | 206 | 00101001 | 41 |
| 143 | 01010000 | 80 | 207 | 01010010 | 82 |
| 144 | 10100000 | 160 | 208 | 10100100 | 164 |
| 145 | 00100011 | 35 | 209 | 00101011 | 43 |
| 146 | 01000110 | 70 | 210 | 01010110 | 86 |
| 147 | 10001100 | 140 | 211 | 10101100 | 172 |
| 148 | 01111011 | 123 | 212 | 00111011 | 59 |
| 149 | 11110110 | 246 | 213 | 01110110 | 118 |
| 150 | 10001111 | 143 | 214 | 11101100 | 236 |
| 151 | 01111101 | 125 | 215 | 10111011 | 187 |
| 152 | 11111010 | 250 | 216 | 00010101 | 21 |
| 153 | 10010111 | 151 | 217 | 00101010 | 42 |
| 154 | 01001101 | 77 | 218 | 01010100 | 84 |
| 155 | 10011010 | 154 | 219 | 10101000 | 168 |
| 156 | 01010111 | 87 | 220 | 00110011 | 51 |
| 157 | 10101110 | 174 | 221 | 01100110 | 102 |
| 158 | 00111111 | 63 | 222 | 11001100 | 204 |
| 159 | 01111110 | 126 | 223 | 11111011 | 251 |
| 160 | 11111100 | 252 | 224 | 10010101 | 149 |
| 161 | 10011011 | 155 | 225 | 01001001 | 73 |
| 162 | 01010101 | 85 | 226 | 10010010 | 146 |
| 163 | 10101010 | 170 | 227 | 01000111 | 71 |
| 164 | 00110111 | 55 | 228 | 10001110 | 142 |
| 165 | 01101110 | 110 | 229 | 01111111 | 127 |
| 166 | 11011100 | 220 | 230 | 11111110 | 254 |
| 167 | 11011011 | 219 | 231 | 10011111 | 159 |
| 168 | 11010101 | 213 | 232 | 01011101 | 93 |
| 169 | 11001001 | 201 | 233 | 10111010 | 186 |
| 170 | 11110001 | 241 | 234 | 00010111 | 23 |
| 171 | 10000001 | 129 | 235 | 00101110 | 46 |
| 172 | 01100001 | 97 | 236 | 01011100 | 92 |
| 173 | 11000010 | 194 | 237 | 10111000 | 184 |
| 174 | 11100111 | 231 | 238 | 00010011 | 19 |
| 175 | 10101101 | 173 | 239 | 00100110 | 38 |
| 176 | 00111001 | 57 | 240 | 01001100 | 76 |
| 177 | 01110010 | 114 | 241 | 10011000 | 152 |
| 178 | 11100100 | 228 | 242 | 01010011 | 83 |
| 179 | 10101011 | 171 | 243 | 10100110 | 166 |
| 180 | 00110101 | 53 | 244 | 00101111 | 47 |
| 181 | 01101010 | 106 | 245 | 01011110 | 94 |
| 182 | 11010100 | 212 | 246 | 10111100 | 188 |
| 183 | 11001011 | 203 | 247 | 00011011 | 27 |
| 184 | 11110101 | 245 | 248 | 00110110 | 54 |
| 185 | 10001001 | 137 | 249 | 01101100 | 108 |
| 186 | 01110001 | 113 | 250 | 11011000 | 216 |
| 187 | 11100010 | 226 | 251 | 11010011 | 211 |
| 188 | 10100111 | 167 | 252 | 11000101 | 197 |
| 189 | 00101101 | 45 | 253 | 11101001 | 233 |
| 190 | 01011010 | 90 | 254 | 10110001 | 177 |
| 191 | 10110100 | 180 | 255 | 00000001 | 1 |

## 〈저자소개〉

양 정 모(Jeong-Mo Yang) 정회원
1984년 2월: 동국대학교 사범대 수학과 졸업 이학사
1989년 2월: 동국대학교 대학원 수학과 졸업 이학석사
1997년 2월: 단국대학교 대학원 수학과 졸업 이학박사
1995년~현재: 중부대학교 공과대학 정보보호학과 정교수, 한국정보보호학회 부회장/이사
〈관심분야〉 암호수학, 정보보호, 암호학, 암호알고리즘 설계