

이중 인벌루션 구조를 지니는 가변길이 블록 암호 알고리즘

이인실[†] · 심경섭[†] · 김혜정[†] · 신 원[†] · 신상욱[†] · 이경현[†]

요 약

본 논문에서는 새로운 가변길이 블록 암호 알고리즘을 제안한다. 제안된 블록 암호 알고리즘은 128비트에서 256비트까지의 가변적인 키 길이를 가지며 가변적인 라운드 수를 사용한다. 각 라운드는 서로 다른 두 개의 F함수를 사용하여 2단계로 구성되는 이중 인벌루션 구조를 사용하며, 또한 두 개의 서로 다른 키 스케줄링 알고리즘을 사용하여 알려진 공격에 대해 안전하도록 설계하였다.

A Variable Length Block Algorithm with Double Involution-BADI

In sil Lee[†], Kyeong seop Sim[†], Hea jeong Kim[†], Weon Shin[†],
Song uk Shin[†], Kyung hyune Rhee[†]

ABSTRACT

In this paper, we propose a new variable length block cipher. It has a variable key length from 128-bit to 256-bit and uses a variable number of rounds. In each round, the proposed algorithm uses the double involution stucture which consists of two steps and two different F functions. In addition, the proposed algorithm has two different key schedulings for providing the strength against known attacks.

1. 서 론

본 논문에서는 많은 양의 메시지를 빠른 처리 속도로 암호화하는데 적합한 새로운 안전한 블록 암호 알고리즘을 제안한다. 기존의 블록 암호 알고리즘으로는 DES(Data Encryption Standard), IDEA (International Data Encryption Algorithm), SAFER (Secure And Fast Encryption Routine), MISTY 등이 있는데 DES는 1973년 NBS(National Bureau of Standard)가 암호 시스템을 등록시키기 위해 공고하여 1977년 표준으로 채택(FIPS(Federal Information Process Standard)-46)하였으며 1997년까지 미국 표준으로 사용하였다. DES는 64비트의 블록크기와 56비트의 키 길이를 가지며, 1998년 이후 DES의 대

체 알고리즘으로 AES(Advanced Encryption Standard)를 개발추진중이다. IDEA는 1990년 Lai와 Massey가 Europe의 표준 블록 알고리즘으로 PES (Proposed Encryption Standard)를 제안하였으며 1991년에 차분공격(Differential Cryptanalysis)[1]에 강하도록 PES를 개선시킨 IPES(Improved PES)를 개발하였고, 1992년에 IDEA라는 이름으로 바꾸어서 사용하게 되었다. IDEA는 64비트의 블록크기와 128비트의 키 길이를 가지며 차분공격에 강하도록 키에 의존하는 S-box를 사용하며, PGP (Pretty Good Privacy)의 기밀성 서비스를 제공하는데 사용하고 있다. SAFER는 1993년 Massey가 개발한 알고리즘으로 64비트의 블록크기와 64비트의 키 길이를 가지며 키 스케줄링을 제외한 모든 부분이 바이트 단위로 구성되어 있어 소프트웨어적으로 구현이 용

[†] 부경대학교 전자계산학과

이하므로 스마트 카드용으로 적합하다[2][3]. MISTY는 Matsui, Ichigawa, Sorimachi, Tokita, Yanagishi가 만든 알고리즘으로 64비트의 블록크기와 128비트의 키 길이를 가지며 Feistel 구조에서 라운드 함수 내부를 한번 더 Feistel 구조로 설계하였고 S-box의 크기를 다르게 사용하므로 대수적 공격에 강하다[4][5].

본 논문에서는 지금까지 다양하게 제안된 블록 암호 알고리즘의 안전성을 근간으로 각종 멀티미디어 데이터 응용에 적합한 가변성 있는 알고리즘을 새로이 제안한다. 본 논문에서 제안된 블록 암호 알고리즘은 128비트의 블록크기, 가변적인 키 길이와 두 개의 다른 키 스케줄링을 가지며 가변적인 라운드 수를 사용할 수 있도록 하여 알고리즘의 용도에 따라서 다양한 가변성을 부가하였다. 이러한 알고리즘의 가변성은 멀티미디어 데이터 암호화의 QoS(Quality of Service)에 따르는 각 서비스별 암호화에 개별적으로 적용하는데 유용하게 이용할 수 있다. 또한 라운드마다 두 단계로 나누어서 서로 다른 두 개의 F함수를 적용한 이중 인벌루션(double involution)구조를 가짐으로써 기존의 알려진 공격에 안전하도록 설계하였다.

2. 전체 구조 및 구조도

2.1 기본구조

본 논문에서 제안하는 블록 암호 알고리즘은 Feistel 구조를 기본 구조로 한다. Feistel 구조는 암호화와 복호화가 같은 알고리즘으로 복호화시에는 서브키만 역순으로 적용하며 라운드에 상관없이 역변환이 가능하고 알고리즘 수행 속도가 빠르다. 또한, 하드웨어, 소프트웨어적으로 구현이 용이하다는 장점이 있다. 본 논문에서 제안한 블록 암호 알고리즘은 이중 인벌루션 구조로 되어 있으며 confusion과 diffusion 효과가 최소 라운드 내에서 이루어지도록 구성하였고 두 개의 키 스케줄링과 F함수를 사용한다.

기본적인 라운드의 수는 8라운드로 설계하며 복잡도를 높이기 위해 4의 배수로 증가시킬 수 있도록 라운드를 가변적으로 설계하고 차분공격의 대상이 되는 S-box대신 차분공격과 선형공격(Linear Crypt-

analysis)[6]에 강한 부울함수를 사용한다[7]. 부울함수는 비 선형적이고 입력의 차이가 한 비트이면 출력은 최소 두 비트가 차이가 나는 SAC(Strict Avalanche Criteria)를 만족하며 출력 비트에 독립적이면서 bijection을 만족하도록 설계한다. 키 스케줄링은 각각의 입력의 균등성을 보장하고 있고 모든 연산은 32비트 단위로 수행되어 다정도 연산이 필요없으므로 수행속도가 빠르다.

2.2 전체적인 구조

제안된 블록 암호 알고리즘은 128비트의 평문 블록을 입력으로 받아 128비트의 암호문 블록을 출력하며 128비트, 160비트, 256비트의 가변적인 키를 사용한다. 초기 입력 평문에 키를 적용시켜 반복할 때 각각의 반복 단계를 라운드라고 하며 라운드의 수는 안전성과 속도를 고려해 볼 때 서로 반비례한다. 따라서 본 논문에서 제안하는 블록 암호 알고리즘은 기본적으로는 8라운드를 제안하며 안전성에 따라 4라운드씩 $8 + 4j$ ($j = 0, 1, 2, \dots$)로 라운드 수를 가변적으로 구성하였다. 사용되는 두 개의 F함수는 서로 다른 대수적 군에서 연산을 하는데 $\text{mod } 2^{16} + 1$ 곱셈과 $\text{mod } 2^{16}$ 덧셈을 결합하고 암호학적으로 강한 부울 함수를 사용한다. 사용된 연산은 $\text{mod } 2^{16}$ 덧셈, $\text{mod } 2^{16} + 1$ 곱셈, 순환 쉬프트, 비트 단위 논리 연산, 즉 AND, OR, NOT, XOR을 사용한다.

2.3 암호화

암호화는 각 라운드가 2단계로 구성되며 입력 128비트를 64비트씩 왼쪽과 오른쪽으로 나누고 나눈 64비트를 각각 32비트씩 왼쪽과 오른쪽으로 나누는 이중 인벌루션 Feistel 구조를 가지며 두 단계로 나누어 F함수의 순서를 바꾸어서 적용한다. 전체적인 구조도는 그림 1과 같다.

“田”기호는 $\text{mod } 2^{32}$ 덧셈이고, “⊕”는 비트별 XOR이며 “<<”는 왼쪽으로 순환 쉬프트이고 “&”는 비트별 AND 연산이며 마지막 라운드 후에는 변수를 서로 맞바꾸지 않고 바로 128비트 암호문으로 출력한다.

2.4 복호화

복호화는 암호화와 같은 알고리즘을 사용하고 단

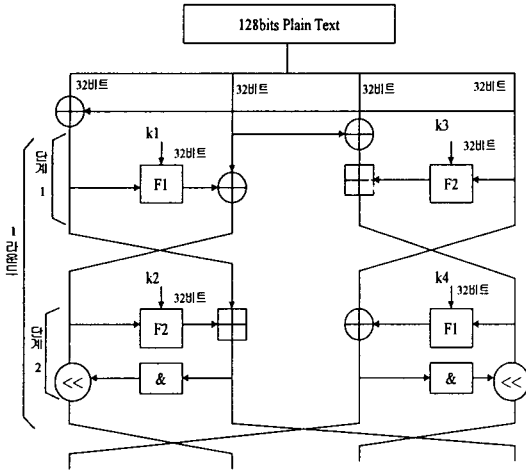


그림 1. 전체 구조도

지 암호화의 역순으로 수행하며 키, 역시 역순으로 적용한다. 연산에 있어서도 $\text{mod } 2^{16}$ 덧셈은 $\text{mod } 2^{16}$ 뺄셈(2의 보수를 이용한 덧셈)을 하고 왼쪽 순환 쉬프트는 오른쪽 순환 쉬프트를 한다.

2.5 F1함수

서로 다른 대수적 군에 의해 연산을 함으로써 confusion 효과를 높일 수 있으며 입력 32비트를 상위비트와 하위비트로 16비트씩 분할해서 수행하고 입력 키 32비트 역시 상위비트와 하위비트 16비트씩 서브키 K1과 K2로 분할한다. MUL은 $\text{mod } 2^{16} + 1$ 곱셈이고 ADD는 $\text{mod } 2^{16}$ 덧셈이다. 이 때 입력 키가 64비트로 확장되더라도 32비트씩 서브키로 분할 처리가 가능하므로 확장성이 용이하다. 간단하게 구조도로 나타내면 그림 2와 같다.

2.6 F2함수

F2함수는 암호학적으로 강한 부울 함수로 구성하는데 먼저, 32비트의 키 k3은 4개의 8비트 (c_1, c_2, c_3, c_4)로 분할되어 각각 부울 함수 BF-1, BF-2, BF-3, BF-4에 입력으로 들어가고 32비트 입력 P는 4개의 부울 함수에 동일하게 사용된다. 각각의 부울 함수에서는 32비트 입력 P를 8비트 (a, b, c, d)로 나누어서 적용한다. 사용되는 부울 함수는 다음과 같고 그림 3에 간단한 구조도를 나타내었다.

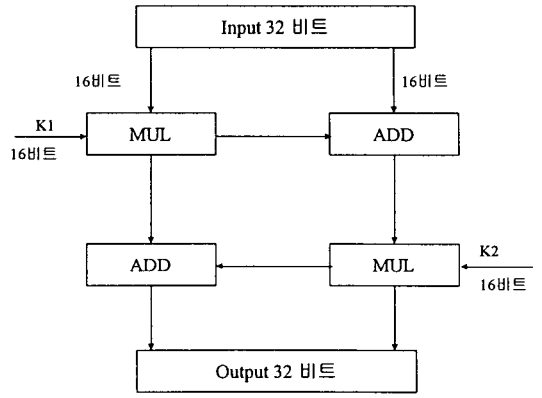


그림 2. F1함수의 구조도

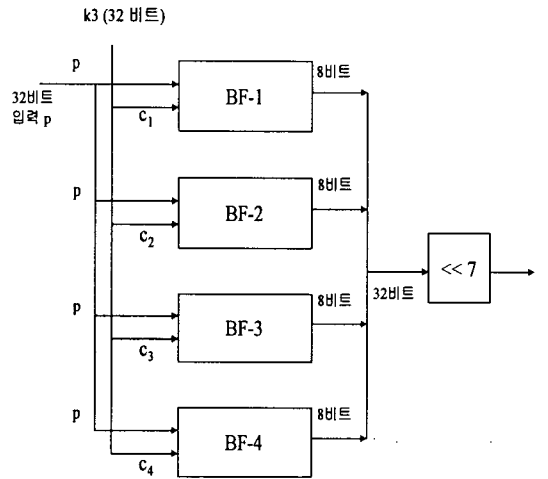


그림 3. F2함수의 구조도

$$\begin{aligned}
 BF-1(a, b, c, d, c_1) &= (a \& b) \wedge (c \& d) \wedge ((b \& c \& d) \wedge c_1) \\
 BF-2(a, b, c, d, c_2) &= (b \wedge ((d \& c_2) | (a \& c))) \\
 BF-3(a, b, c, d, c_3) &= (a \wedge (b \& (a \wedge d))) \wedge (((a \& d) \wedge c) \& c_3) \\
 BF-4(a, b, c, d, c_4) &= ((a \& c_4) \wedge b) \wedge (c \& d)
 \end{aligned}$$

이러한 5변수 부울함수는 다음과 같은 암호학적으로 강한 성질들을 만족하는데 첫 번째 성질은 출력 수열의 0과 1이 균형(balanced)을 이루고 두 번째 성질은 SAC을 만족하며 세 번째 성질은 높은 비선형성(non-linearity)을 가지는 것이다.

3. 키 스케줄링

적은 양의 키 비트를 가지고 많은 양의 라운드 키를 생성하는 것이 키 스케줄링의 목적으로 설계 원칙은 먼저 라운드 키 생성에 있어서 모든 키 비트가 사용되어야 하고 모든 키 비트가 각 라운드마다 독립적으로 새로운 키를 생성해야 한다. 또한, 적어도 일방향성을 만족해야 하고 취약키가 없어야 하며 라운드 변환에서의 대칭성과 라운드간의 대칭성을 제거해야 한다. 본 알고리즘에서는 128비트, 160비트, 256비트의 가변길이의 키를 사용하며 키 스케줄링도 이원적인 키 스케줄링(K_1 과 K_2)을 사용한다.

3.1 K_1 스케줄링

K_1 스케줄링의 단계를 살펴보면

- 1) K_1 을 4개의 32비트로 분할하고 각각을 fk_1, fk_2, fk_3, fk_4 라고 한다.
- 2) 4개의 32비트 레지스터 R1, R2, R3, R4를 초기화한다.
 $R1 = DF7BD629, R2 = E9DB362F,$
 $R3 = 5D00F20F, R4 = C3D11FD2$
- 3) i 번째 라운드의 단계 1에서 사용하는 서브키중 K_1 스케줄링에서는 $k1$ 과 $k3$ 을 생성한다.

$$fk_1 = ((fk_1 \wedge R1) \vee (\neg fk_1 \wedge R2)) \ll^{14 \& 5} \oplus R3$$

$$fk_2 = (fk_2 \oplus R2 \oplus R3) + R4 \ll^{R1}$$

$$fk_3 = ((fk_3 \wedge R3) \oplus (fk_3 \wedge R4) \oplus (R3 \wedge R4)) + fk_1$$

$$fk_4 = ((fk_4 \wedge R1) \vee (R4 \wedge \neg R1)) + fk_2$$

$$fk_1 = fk_1 \oplus fk_3 + R2$$

$$fk_2 = fk_2 + fk_4 \oplus R3$$

$$R1 = R3, R2 = R4, R3 = fk_3 \ll^{11}, R4 = fk_4$$

생성된 서브키 중에서 fk_1 과 fk_2 는 라운드 i 의 $k1$ 과 $k3$ 으로 사용한다.

- 4) i 를 1증가시켜서 단계 3)을 반복하여 모든 8라운드에 대한 서브키를 생성한다.

3.2 K_2 스케줄링

K_2 스케줄링의 단계를 살펴보면

- 1) K_2 를 4개의 32비트로 분할하고 각각을

sk_1, sk_2, sk_3, sk_4 라고 한다.

- 2) 두 개의 32비트 레지스터 R1과 R2를 초기화한다.

$$R1 = 589B4312, R2 = 91EB718E$$

- 3) i 번째 라운드의 단계 2에서 사용하는 서브키 $k2$ 와 $k4$ 를 생성한다.

$$sk_1 = sk_1 \oplus R1 \ll^7 \oplus R2$$

$$sk_2 = sk_2 \boxplus sk_1 \oplus R2 \ll^5$$

$$sk_3 = (sk_3 \oplus sk_2) \ll^7$$

$$sk_4 = sk_4 \boxplus sk_3$$

$$sk_1 = sk_1 \oplus sk_2$$

$$sk_2 = sk_3 \boxplus sk_4$$

$$R1 = sk_4, R2 = sk_3 \ll^{11}$$

생성된 서브키 중에서 sk_1 과 sk_2 는 라운드 i 의 $k2$ 와 $k4$ 로 사용한다.

- 4) i 를 1증가시켜서 단계 3)을 반복하여 모든 8라운드에 대한 서브키를 생성한다.

4. 알고리즘의 안전성

기존의 블록 암호 알고리즘에서는 하나의 인벌루션 구조를 사용하는데 반해 제안된 알고리즘의 각 라운드는 서로 다른 두 개의 F함수를 사용하는 이중 인벌루션 구조를 사용하므로 알려진 공격에 대해 기존의 알고리즘보다 높은 안전성을 제공한다. 이러한 복수개의 F함수를 사용하면, 기존의 단일 F함수에 의한 차분공격이나 선형공격에 강인한 성질을 제공하며 사용되는 F함수의 개수는 암호알고리즘의 성능과 밀접하게 연관되어 있다.

F1라운드 함수는 서로 다른 군에 속하는 연산을 사용하여 라운드의 각 출력 비트가 평문의 모든 비트와 라운드에 사용된 모든 비트에 의존하게 한다. 즉 F1은 $\text{mod } 2^{16} + 1$ 곱셈과 $\text{mod } 2^{16}$ 덧셈 연산을 사용하여 두 개의 16비트 입력 서브블록이 두 개의 16비트 키 서브블록에 의해 제어되며 두 개의 16비트 출력 서브블록으로 변환되어 완전한 diffusion을 달성한다. F2함수는 암호학적으로 강한 부울 함수를 사용하여 안전성을 향상시키는데 암호학적으로 강한 부울함수의 세 가지 성질중에서 첫 번째 성질은

Webster와 Tavares[8]가 소개한 SAC(Strict Avalanche Criteria)를 만족해야 한다는 것이다. 이러한 성질은 입력 비트가 1비트가 차이가 날 때 출력 비트는 최소한 2비트이상 차이가 나는 성질로 암호화적인 함수가 SAC를 만족하면 각 출력비트는 입력비트의 한 비트가 변하면 1/2의 확률로 변한다. 두 번째 성질은 암호 알고리즘이 비선형이어야 한다는 것으로 Meier와 Staffelbach[9]는 부울함수에 대한 비선형성이 암호화적인 설계에 대해 적합하다는 것을 증명하였다. 세 번째 성질은 Adam과 Tavares[10]가 제안한 부울함수가 기본적으로 0과 1이 균형이 될 필요가 있다는 것이다.

위에서의 세 가지 성질들을 만족하는 암호화적으로 강한 부울 함수로 구성된 라운드 함수와 가변적인 순환 쉬프트를 사용함으로써 차분공격과 선형공격에 대해 안전성을 제공한다[11]. 설계한 알고리즘에 대한 안전성을 평가하기 위해 통계적 랜덤 검정과 Avalanche Effect 검정의 결과로써 안전성을 확인한다[12].

4.1 통계적 랜덤 검정

블록 암호를 CBC(Cipher Block Chaining) 모드로 동작시켰을 때 출력 수열에 대해 다음의 통계적 검정으로 4가지 검정 데이터를 사용하여 각 검정을 수행하였다. 각 데이터 화일은 1Mbit의 길이를 가진다.

표 1. 검정 데이터

데이터 화일	사용된 128 비트 키	사용된 128 비트 평문 스트링
데이터1	0000000000000000 0000000000000000	0his is a plaint
데이터2	0000000000000000 0000000000000001	1his is a plaint
데이터3	ffffffffffffffff ffffffffffffffff	0his is a plaint
데이터4	0000000000000000 0000000000000011	0his is a plaint

4.1.1 Frequency Test(빈도 검정)

대상 수열에 대해 0과 1의 수가 균일하게 분포하고 있는지를 결정하는데 사용되는 검정이다.

- 통계량의 분포 : $T \sim$ 자유도 1인 χ^2 분포
유의수준 α 의 기각역 : $T > \chi^2(1, \alpha)$
- 검정 결과 ($\alpha = 5\%$, $\chi^2(1, \alpha) = 3.841$)

표 2. 빈도 검정 결과

데이터 화일	χ^2 값	검정 결과
데이터1	0.1603 ($n_0 = 524083$, $n_1 = 524493$)	통과
데이터2	1.1878 ($n_0 = 523730$, $n_1 = 524846$)	통과
데이터3	0.4835 ($n_0 = 523932$, $n_1 = 524644$)	통과
데이터4	2.3627 ($n_0 = 523501$, $n_1 = 525075$)	통과

4.1.2 Serial Test(계열 검정)

대상 수열에서 00, 01, 10, 11의 비트쌍이 고려되어지는 검정으로 검정하려는 대상 이진 수열에서 한 비트가 그 다음 비트로 가는 전이확률을 검증하는 것이다.

- 통계량의 분포 : $T \sim$ 자유도 2인 χ^2 분포
유의수준 α 의 기각역 : $T > \chi^2(2, \alpha)$
- 검정 결과 ($\alpha = 5\%$, $\chi^2(2, \alpha) = 5.991$)

표 3. 계열 검정 결과

데이터 화일	χ^2 값	검정 결과
데이터1	2.1516($n_{00}=262300$, $n_{01}=261783$, $n_{10}=261782$, $n_{11}=262710$)	통과
데이터2	1.4730($n_{00}=261724$, $n_{01}=262006$, $n_{10}=262006$, $n_{11}=262839$)	통과
데이터3	1.3038($n_{00}=261556$, $n_{01}=262376$, $n_{10}=262376$, $n_{11}=262267$)	통과
데이터4	3.2398($n_{00}=261117$, $n_{01}=262383$, $n_{10}=262384$, $n_{11}=262691$)	통과

4.1.3 Poker Test(포커 검정)

대상 수열에서 임의의 m 비트의 패턴을 고려하는 검정으로서 2^m 의 서로 다른 패턴이 존재한다.

- 통계량의 분포 : $T \sim$ 자유도 $2^m - 1$ 인 χ^2 분포
유의수준 α 의 기각역 : $T > \chi^2(2^m - 1, \alpha)$
- 검정 결과($m=5$, $\alpha = 5\%$, $\chi^2(31, \alpha) = 44.9853$)

표 4. 포커 검정 결과

데이터 화일	χ^2 값	검정 결과
데이터1	25.9161	통과
데이터2	39.1901	통과
데이터3	34.1428	통과
데이터4	23.2150	통과

4.1.4 Run Test(런 검정)

수열에서 '0'이나 '1'이 연속하여 나타나는 것을 run이라 한다. 이 검정은 수열에서 다양한 길이의 run들의 수가 난수에 대해 예측되어지는 것과 같은지를 결정한다.

- 통계량의 분포: $T \sim$ 자유도 $2(L-1)$ 인 χ^2 분포
- 유의수준 α 의 기각역: $T > \chi^2(2(L-1), \alpha)$
- 검정 결과 ($L=15, \alpha = 5\%, \chi^2(28, \alpha) = 41.34$)

표 5. 런 검정 결과

데이터 화일	χ^2 값	검정 결과
데이터1	23.6430	통과
데이터2	29.4638	통과
데이터3	31.4948	통과
데이터4	25.4448	통과

4.1.5 Autocorrelation Test(자기상관 검정)

이진 수열 S_n 이 주어졌을 때 (S_n)에서 d 비트만큼 전이시켜 생성한 수열 (S_{n+d})과의 상관 관계를 조사하는 검정이다.

- 통계량의 분포: $T \sim$ 자유도 2인 χ^2 분포
- 유의수준 α 의 기각역: $T > \chi^2(2, \alpha)$
- 검정 결과 ($\alpha = 5\%, \chi^2(2, \alpha) = 5.991$)

표 6. 자기 상관 검정 결과

데이터 화일	d=3, χ^2 값	d=7, χ^2 값	검정 결과
데이터1	2.1336	2.7097	통과
데이터2	5.6538	2.2305	통과
데이터3	1.6593	2.0641	통과
데이터4	3.3680	4.5009	통과

4.2 Avalanche Effect Test(쇄도 효과 검정)

키와 평문이 1 비트씩 바뀌었을 때 4 라운드까지의 avalanche effect를 보이며 테스트는 128비트의 각 비트 위치를 전부 한번씩 변경시켜 수행하였다.

표 7. 쇄도 효과 검정 결과

	1라운드 후	2라운드 후	3라운드 후	4라운드 후
평문 고정, 키 1비트 변경	16.6%	55.2%	53.6%	47.8%
키 고정, 평문 1 비트 변경	27.4%	48.8%	50.8%	48.9%

실험 결과 2 라운드 후에는 완전한 avalanche effect를 보이는 것을 알 수 있다.

5. 알고리즘 동작 속도에 대한 성능 평가

성능 평가 환경은 CPU는 Pentium Pro 180MHz, 메모리는 64MBytes, 컴파일러는 DJGPP 2.01를 사용해서 평가하였을 경우, 128 비트 입력 키에 대한 각 라운드의 수행 속도는 표 8과 같다.

표 8. 라운드에 따른 성능 평가

라운드 수	성능(Mbps)
1	27.30
4	10.92
8	6.06
12	3.90
16	2.87

또한, 다른 블록 알고리즘과 성능을 비교해 보았을 때 그 결과는 표 9와 같이 처리 속도면에서 훨씬 떨어지는 것처럼 보이지만 사실상 제안방식은 라운드별로 2단계를 수행하므로 IDEA나 RC5의 16라운드 처리와 같기 때문에 처리 속도가 비슷하다는 것을 알 수 있다. 따라서 제안 암호 알고리즘이 비록 기존의 블록 암호 알고리즘과 유사한 수행속도를 나타내었지만 알고리즘을 최적화시키고 어셈블러 수준으로 작성할 경우와 특히, 이를 하드웨어로 구현하여

custom chip화 할 경우 상당히 고속의 암호화가 적용 가능하리라 기대된다.

표 9. 다른 블록 암호알고리즘과의 성능 비교

알고리즘	블록 크기	키 크기 (bits)	라운드 수	처리속도 (Mbps)
DES	64	64	16	15.9
IDEA	64	128	8	15.9
RC5	64(가변)	128(가변)	12(가변)	22.75
제안방식	128	128(가변)	8(가변)	6.06

6. 결 론

본 논문에서는 효율적이고 안전한 새로운 블록 암호 알고리즘을 제안하였다. 제안된 블록 암호는 비교적 간단한 연산만을 사용하므로 소프트웨어와 하드웨어 구현이 효율적이다. 또한 128비트에서 256비트까지의 가변적인 키 길이와 가변적인 라운드 수를 가지므로 안전성의 요구에 따라 키 길이와 라운드 수를 조절할 수 있다. 각 라운드는 2단계로 나누어서 서로 다른 2개의 F함수를 사용하는 이중 인벌루션 구조를 가지며, 다른 2개의 키 스케줄링을 사용함으로써 기존의 알려진 공격에 대해 안전성을 제공한다. 제안된 알고리즘의 가장 큰 특징은 가변성을 제공함으로써 다양한 정보원으로부터 추출된 멀티미디어 정보의 암호화에 정보원 서비스별로 알고리즘을 적용함으로써 서비스 수준에 맞는 알고리즘의 안전성을 제공할 수 있으리라 기대된다.

참 고 문 헌

[1] J.L. Massey, "SAFER K-64: A byte-oriented block-ciphering algorithm", R. Anderson, editor, Fast Software Encryption, Cambridge Security Workshop(LNCS 809), Springer-Ver-

lag, pp.1-17, 1994.
 [2] J.L. Massey, "SAFER K-64: One year later", B. Preneel, editor, Fast Software Encryption, Second International Workshop(LNCS 1008), Springer-Verlag, pp.212-241, 1995.
 [3] M. Matsui, "New Block Encryption Algorithm MISTY", R. Anderson, editor, Fast Software Encryption, Cambridge Security Workshop, Springer-Verlag, pp.53-67, 1997.
 [4] B. Schneier, "Applied Cryptography", John Wiley & Sons, 1996.
 [5] E. Biham, A. Shamir, "Differential Cryptanalysis of DES-like cryptosystems", Journal of Cryptology, vol. 4, no.1, pp.3-72, 1991.
 [6] M. Matsui, "Linear Cryptanalysis method for DES cipher", EUROCRYPT'93, LNCS. vol.765, Springer-Verlag, pp.386-397, 1993.
 [7] R. Rivest, "The RC5 Encryption Algorithm", Fast Software Encryption'94, LNCS, vol. 1008, Springer-Verlag, pp.86-96, 1994.
 [8] A. F. Webster and S. E. Tavares, "On the Design of S-boxes", Proc. of CRYPTO'85, Springer-Verlag, 1985.
 [9] W. Meier and O. Staffelbach, "Nonlinearity Criteria for Cryptographic Functions", Proc. of EUROCRYPTO'89, 1989.
 [10] C. Adam and S. E. Tavares, "The Structured Design of Cryptographically Good S-boxes", J. of Cryptology, 1990.
 [11] X. Lai, "On the design and security of block ciphers", ETH Series in Information Processing, J.L. Massey (editor), vol. 1, Hartung-Gorre Konstanz, Technische Hochschule (Zurich), 1992.
 [12] Knuth, Donald Ervin, "The Art of Computer Programming", 2nd ed., Addison-Wesley, 1981.



이 경 현

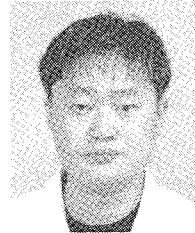
1982년 경북대학교 사범대학 수학교육과 졸업(이학사)
 1985년 한국과학기술원 응용수학과 졸업(이학석사)
 1992년 한국과학기술원 수학과 졸업(이학박사)
 1985년 2월~1993년 2월 한국 전자

통신 연구소 연구원, 선임연구원
 1991년 3월~1993년 2월 한국 전자 통신 연구소 선임 연구원
 1993년 3월~현재 부경대학교(구 부산수산대학교 전임강사, 조교수)
 1995년 7월~1996년 7월 Univ. of Adelaide, 응용수학과, Australia 방문교수



김 혜 정

1996년 부산수산대학교(현 부경대학교) 응용수학과 졸업(이학사)
 1996년~현재 부경대학교 전자계산학과 석사과정



심 경 섭

1998년 동서대학교 컴퓨터공학과 졸업(공학사)
 1998년~현재 부경대학교 전자계산학과 석사과정



신 상 옥

1995년 부산수산대학교(현 부경대학교) 전자계산학과 졸업(이학사)
 1997년 부경대학교 전자계산학과 대학원 졸업(이학석사)
 1997년~현재 부경대학교 전자계산학과 박사과정



이 인 실

1996년 동서대학교 컴퓨터공학과 졸업(공학사)
 1997년~현재 부경대학교 전자계산학과 석사과정



신 원

1996년 부산수산대학교(현 부경대학교) 전자계산학과 졸업(이학사)
 1998년 부경대학교 전자계산학과 대학원 졸업(이학석사)
 1998년~현재 부경대학교 전자계산학과 박사과정