논문 00-25-3A-10 한국통신학회논문지 '00-3 Vol.25 No.3A

IDEA 알고리즘의 특성분석

정회원 김 지 홍*, 장 영 달*, 윤 석 창*

The Properties Analysis of IDEA Algorithm

Ji-hong Kim*, Young-dal Jang*, Suck-chang Yun* ***Regular Members***

요 약

본 논문에서는 블록암호시스템의 대표적인 방법인 IDEA(International Date Encryption Algorithm)알고리즘을 다룬다. IDEA 알고리즘에서의 키생성 알고리즘을 분석함으로서, 라운드별 사용되는 키 비트열과 사용되지 않는 키 비트열을 분류한다. 이를 이용하여 MA(Multiplication/Addition) 구조를 생략한 형태의 IDEA 알고리즘에 대한 MSB (Most Significant Bit) 차분에 의한 차분 분석법(differential analysis)과 입력계열과 각 라운드별 사용 키계열의 LSB(Least Significant Bit) 비트만을 사용하는 선형 분석법(linear analysis)을 제안한다.

ABSTRACT

In this paper, we deal with block cipher algorithm IDEA(International Date Encryption Algorithm), previously known as typical block cipher system. First of all, analysing key scheduler we classify the key sequences with the used key bits and the unused key bits in each round. With this properties we propose the two method, which are differential analysis using differences of plaintext pairs and linear analysis using LSB bits of plaintexts and key sequences.

Ⅰ. 서 론

국가 기간망, 초고속정보통신망 구축과 함께 정보통신분야의 급속한 발전과 컴퓨터의 많은 보급으로 인하여 데이터 보안이 매우 중요하고도 필수적인 문제로 부각되고 있다.

특히 각종 정보시스템을 융합시킨 종합정보통신망(ISDN)내에서의 전송 또는 처리되는 데이터, 기업의 경영비밀 및 개인비밀 등이 도청되어 범죄에 악용하는 사례가 많아, 현재 사회적으로 큰 문제가 되고 있다.

따라서 컴퓨터 전산망 혹은 이동통신망에서 이러한 불법적인 정보사용을 막기 위한 중요한 기술중 하나인 암호와 인증에 관한 연구가 최근 활발히 진행되고 있으며, 국내에서도 정보통신부 산하의 정보보호센타 설립과 함께 정보보호에 대한 인식이 높아지고 있다. 일반적으로 이러한 정보보호를 위하여 사용되고 있는 기술로는 사용자 및 메시지에 대한 인증기능을 부가하기 위한 공개키 암호알고리즘과 사용자간의 메시지를 보호하기 위한 블록암호시스템이 전반적으로 사용되고 있다. 정보보호를 위한 블록암호시스템의 대표적인 예로서는 DES 알고리즘[4,6]과 IDEA알고리즘[3]이 있으나, DES 알고리즘은 안전성에 문제가 제기되고 있으며, 현재 새로운 알고리즘인 AES (Advanced Encryption Standard)를 제정하고 있다. 따라서 현재까지 블록 암호알고리즘으로 가장 안전성이 높은 것으로 평가되고 있는 알고리즘은 64 비트의 평문입력과 128 비트의 키 구조를 사용하여 64 비트의 출력계열을 생성하며, 자체적으로 3가지의 연산법칙과 MA구조를 이용하여 복잡도를 향상시킨 구조의 IDEA 암호알고리즘이다.

* 세명대학교 전자공학과 정보보호연구실(jhkim@infosec.semyung.ac.kr)
논문번호 : 99341-0823, 접수일자 : 1999년 8월 23일
※ 본 연구는 '98년도 세명대학교 교내연구비 지원 과제로 수행되었습니다.

IDEA 알고리즘은 현재 PGP(Pretty Good Protocol) 알고리즘[6] 등의 다양한 응용분야에 사용되고 있다.

본 논문에서는 IDEA 알고리즘에 관한 분석을 위하여, 각 라운드별로 사용된 키와 사용하지 않는 키 비트를 분석하고, 이러한 결과를 이용하여 IDEA 알고리즘에 대한 MSB 를 이용한 차분 분석과 LSB 를 이용한 선형 분석을 시행한다.

Ⅱ. IDEA 해석

2.1 IDEA 암호부 분석

IDEA 알고리즘의 암호화 과정[4,5,6]은 64비트 평문입력을 128비트 키를 이용하여 64비트의 암호문을 생성하는 알고리즘이다. 각 입력은 16bit binary tuple로 구성되며, 알고리즘에서 사용된 연산식은 다음과 같다.

⊕ : Z_2^{16}상에서의 덧셈(이진합)

⊞ : Z_{2^n}상에서의 덧셈

덧셈 연산과정에서 발생되는 캐리 비트를 처리하는 덧셈연산(⊞)과 처리하지 않는 이진합 연산(⊕)이 있다.

또한 곱셈연산(⊙)은 다음과 같은 특성을 가진다[1].

$$\begin{aligned} 1\odot A \bmod(2^n+1) &= A \bmod(2^n+1) \\ &= A \bmod(2^n) \\ -1\odot A \bmod(2^n+1) &= -A \bmod(2^n+1) \\ &= /A+2 \bmod(2^n) \end{aligned} \quad (1)$$

여기서 "1" 이란 mod(2^n+1)에서 "1" 인 값을 의미하며, "-1" 이란 mod(2^n+1)에서는 "2^n"을 의미하며, mod(2^n)에서 "0"을 의미한다[1].

IDEA 1단 구조를 살펴보면 1단계 변환과 관련된 식은 다음과 같다

$$\begin{aligned} &P1 = X1\odot Z1,\ P2 = X2\boxplus Z2 \\ &P3 = X3\boxplus Z3,\ P4 = X4\odot Z4 \end{aligned} \quad (2)$$

MA 구조와 관련된 입력 A,B 및 출력 S,T에 관한 식은 다음과 같다

$$\begin{aligned} &A = P1\oplus P3 = X1\odot Z1 \oplus X3\boxplus Z3 \\ &B = P2\oplus P4 = X2\boxplus Z2 \oplus X4\odot Z4 \\ &S = A\odot Z5 \boxplus T \\ &T = B\boxplus(A\odot Z5) \odot Z6 \end{aligned} \quad (3)$$

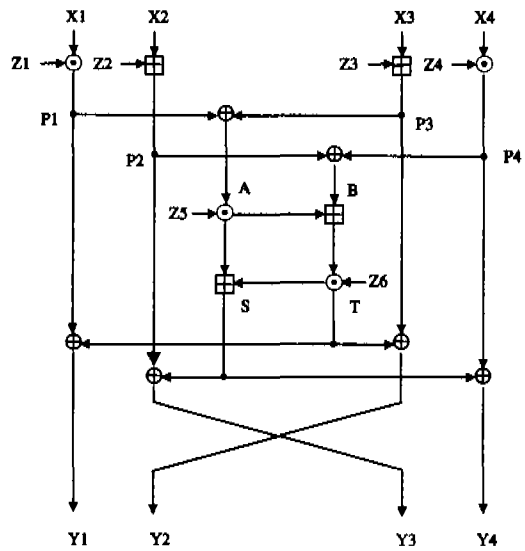


그림 1. IDEA알고리즘(1라운드)블럭도

IDEA 1단의 출력에 관한 식은 다음과 같다.

$$\begin{aligned} &Y1 = T \oplus P1 = T \oplus (X1\odot Z1) \\ &Y2 = T \oplus P3 = T \oplus (X3\boxplus Z3) \\ &Y3 = S \oplus P2 = S \oplus (X2\boxplus Z2) \\ &Y4 = S \oplus P4 = S \oplus (X4\odot Z4) \end{aligned} \quad (4)$$

위의 4개의 식 중에서 MA 구조를 생략하면 다음과 같은 식이 성립된다

$$\begin{aligned} &Y1\oplus Y2=(X1\odot Z1)\oplus(X3\boxplus Z3)=A \\ &Y3\oplus Y4=(X4\odot Z4)\oplus(X2\boxplus Z2)=B \end{aligned} \quad (5)$$

식(5)는 MA 구조를 생략한 상태에서의 IDEA 1 라운드의 출력값(Y1, Y2, Y3, Y4)과 평문입력과의 관계(X1, X2, X3, X4)를 보인다.

2.2 키 스케줄러 분석

IDEA 알고리즘에서의 키 스케줄러는 128비트의 키 입력계열을 이용하여, 52개의 키 계열을 생성하는 키 스케줄러로 구성된다. 본 논문에서는 MA구조부분을 생략한 구조를 검토한다. 따라서 MA구조와 관련되는 16개의 키 계열은 고려 대상에서 제외하고, 각 라운드별로 사용되는 4개의 키계열은 표 1과 같다.

각 라운드에서 사용되는 키를 보면 Z1에서는 16-21, 59-74, 112-127번 비트까지 38 비트는 사용되지 않으며, Z2에서는 0-15, 32-37, 75-90번 비트

까지 38 비트는 사용되지 않으며, Z3에서는 9-24, 48-53, 84-99번 비트까지 38 비트는 사용되지 않으며, Z4에서는 25-40, 64-69, 100-115번 비트까지 38 비트는 사용되지 않는다.

표 1. 각 라운드별 사용키

라운드 번호	라운드별 사용키			
	Z1	Z2	Z3	Z4
1	Z1 (0-15)	Z2 (16-31)	Z3 (32-47)	Z4 (48-63)
2	Z7 (96-111)	Z8 (112-127)	Z9 (25-40)	Z10 (41-56)
3	Z13 (89-104)	Z14 (105-120)	Z15 (121-8)	Z16 (9-24)
4	Z19 (82-97)	Z20 (98-113)	Z21 (114-1)	Z22 (2-17)
5	Z25 (75-90)	Z26 (91-106)	Z27 (107-122)	Z28 (123-10)
6	Z31 (43-58)	Z32 (59-74)	Z33 (100-115)	Z34 (116-3)
7	Z37 (36-51)	Z38 (52-67)	Z39 (68-83)	Z40 (84-99)
8	Z43 (29-44)	Z44 (45-60)	Z45 (61-76)	Z46 (77-92)
8.5	Z49 (22-37)	Z50 (38-53)	Z51 (54-69)	Z52 (70-85)
사용되지 않는 키 bit	16-21, 59-74 112-127	0-15, 32-37, 75-90	9-24, 48-53 84-99	25-40, 64-69, 100-115

$$Y1 \oplus Y2 = (X1 \odot Z1) \oplus (X3 \boxplus Z3) \qquad (6\text{-}1)$$

$$Y3 \oplus Y4 = (X4 \odot Z4) \oplus (X2 \boxplus Z2) \qquad (6\text{-}2)$$

따라서 식(6-1)과 관련된 Z1과 Z3에서 공통적으로 사용되지 않는 비트는 16-21 비트로 5개의 비트가 있다. 또한 식(6-2)과 관련된 Z2와 Z4에서 공통적으로 사용되지 않는 비트는 32-37 비트로 5개의 비트가 이에 해당된다.

Ⅲ. IDEA 시스템에 대한 차분분석 방법

3.1 차분분석 방법

입력 64 비트의 평문입력은 X1, X2, X3, X4로 4개의 16 비트 서브블럭으로 나뉜다. 입력쌍 X와 X^*에 대한 라운드 입력시 각각의 출력을 Y와 Y^*이라 하고, 이들에 대한 XOR결과 값을 $X'=X \oplus X^*$, $Y'=Y \oplus Y^*$ 이라 정의한다. 따라서 X'=0는 두 개의 평문 입력쌍(X,X^*)이 동일하다는 것을 의미하며, X'=v는 두 개의 평문 입력쌍(X,X^*)의 이진합의 결과가 0x8000으로 나타나는 것을 의미한다. 즉, 16 비트 중 최상위 비트만이 서로 다른 입력쌍을 의미한다. v 값에 해당되는 0x8000에 의한 평문 입력의 차이는 ⊕, ⊞연산에서 항상 동일한 결과를 가진다. 즉, 최상위 비트에서 발생될 수 있는 캐리 비트는 결과에 영향을 주지 않기 때문이다.

표 2. XOR 전달특성[1]

XOR 전달특성	Z1	Z4	Z5	Z6
(0,0,0,v) -> (v,v,v,0)		(-)1		(-)1
(0,0,v,0) -> (v,0,0,0)			(-)1	(-)1
(0,0,v,v) -> (0,v,v,0)		(-)1	(-)1	
(0,v,0,0) -> (v,v,0,v)				(-)1
(0,v,0,v) -> (0,0,v,v)		(-)1		
(0,v,v,0) -> (0,v,0,v)			(-)1	
(0,v,v,v) -> (v,0,v,v)		(-)1	(-)1	(-)1
(v,0,0,0) -> (0,v,0,0)	(-)1		(-)1	(-)1
(v,0,0,v) -> (v,0,v,0)	(-)1	(-)1	(-)1	
(v,0,v,0) -> (v,v,0,0)	(-)1			
(v,0,v,v) -> (0,0,v,0)	(-)1	(-)1		(-)1
(v,v,0,0) -> (v,0,0,v)	(-)1		(-)1	
(v,v,0,v) -> (0,v,v,v)	(-)1	(-)1	(-)1	(-)1
(v,v,v,0) -> (0,0,0,v)	(-)1			(-)1
(v,v,v,v) -> (v,v,v,v)	(-)1	(-)1		

그러나 ⊙ 연산에서는 영향을 주기 때문에 출력결과가 동일한 차분을 유지하기 위해서는 mod (2^n+1) 곱셈연산과 관련되는 키 계열 Z1, Z4, Z5, Z6에 대한 (-)1 외 가정이 필요하다.

표 2 에 의하면, v외 입력에 의한 키 계열과의 영향을 정리하면 다음과 같다.

주기1(Z1,Z4):

(v,v,v,v)->(v,v,v,v)

주기2(Z1,Z4,Z6):

(0,0,0,v)->(v,v,v,0)->(0,0,0,v)

주기3(Z4,Z5):

(0,v,0,v)->(0,0,v,v)->(0,v,v,0)->(0,v,0,v)

주기3(Z1,Z4,Z5):

(v,0,v,0)->(v,v,0,0)->(v,0,0,v)->(v,0,v,0)

주기6(Z1,Z4,Z5,Z6):

(0,0,v,0)->(v,0,0,0)->(0,v,0,0)->(v,v,0,v)
->(0,v,v,v)->(v,0,v,v)->(0,0,v,0)

예로서, X1'= X2'= X3'= X4'= v 이며, Z1 = Z4 = (-)1일 때, 라운드내의 MA구조의 입력쌍과 출력쌍에 대한 차분값은 모두 "0"으로 나타난다. 따라서 라운드의 출력결과는, Y1'= Y2'= Y3'= Y4'= v이며, 이때 IDEA 암호부는 그림 2와 같이 간략화 된다.

본 논문에서는 MA 구조를 제외한 키 계열에 대

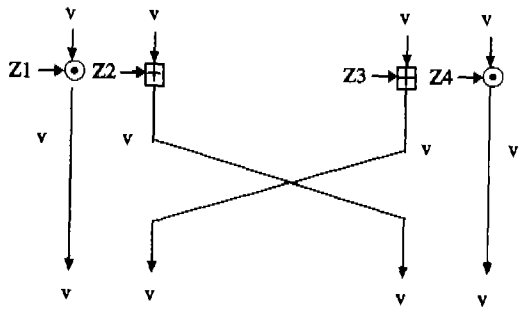


그림 2. IDEA 알고리즘(1라운드)의 차분등가 블록도

한 특성을 분석하기 위하여, 주기 1인 (v,v,v,v)인 경우를 다룬다. 입력 쌍의 이진합이 v인 경우, 라운드의 XOR 전달특성은 입력과 출력이 동일하게 나타난다.

최종 8.5 라운드까지 모두 거친 출력은 다음과 같은 식으로 표기될 수 있다.

$$Y1'=X1'=v,\ Y2'=X2'=v,$$
$$Y3'=X3'=v,\ Y4'=X4'=v$$

표 3 에서와 같이 1-2 라운드에서 키 제한조건으로 사용된 비트는 1-15, 42-63, 97-111 비트이며, 총 52개의 비트가 사용되었으며, 3-4 라운드까지 키 제한조건으로 사용된 비트는 1-24, 42-63, 83-111 비트로서, 총 75개의 비트가 사용되었다. 이와 같이 최종 8.5 라운드까지 키 제한조건으로 사용된 비트는 0-63, 71-111, 117-127 비트이며, 총 116 개의 비트가 사용되었다.

표 3. (v,v,v,v)->(v,v,v,v) 의 XOR 전달특성

라운드 번호	키제한조건 ("0"인 비트)		사용된 비트	누적 사용된 비트	사용된 비트수
	Z1	Z4			
1-2	Z1 (1-15)	Z4 (49-63)	1-15, 42-63 97-111	1-15, 42-63 97-111	52
	Z7 (97-111)	Z10 (42-56)			
3-4	Z13 (90-104)	Z16 (10-24)	3-24, 83-104	1-24, 42-63 83-111	75
	Z19 (83-97)	Z22 (3-17)			
5-6	Z25 (76-90)	Z28 (124-10)	44-58, 76-90 0-10, 117-127	0-24, 42-63, 76-111, 117-127	94
	Z31 (44-58)	Z34 (117-3)			
7-8	Z37 (37-51)	Z40 (85-99)	30-51, 78-99	0-24, 30-63, 76-111, 117-127	106
	Z43 (30-44)	Z46 (78-92)			
8.5	Z49 (23-37)	Z52 (71-85)	23-37, 71-85	0-63, 71-111, 117-127	116

결과적으로 평문 차분쌍 (v,v,v,v)에 의한 차분 분석법에 의하면, 128 비트의 키계열중에서 총 116개의 비트값에 대한 "0" 이란 가정이 필요하며, 나머지 14비트와는 전혀 무관함을 알 수 있다.

이는 7-8 라운드에서 12 비트(30-41 비트)가 추가되고, 마지막 8.5 라운드에서 10 비트(25-29, 71-75 비트)가 추가됨을 알 수 있으며, 만일 4라운드의 IDEA 알고리즘이 사용되었다면 128비트의 키계열 중에서 사용된 비트 수는 75 비트이며, 사용되지 않은 키 비트 수는 53비트가 이에 해당된다.

3.2 LSB 연산에 의한 분석 방법

IDEA 알고리즘에서 사용되는 세 가지 연산방법은 각 연산으로 입력되는 워드형 데이터의 LSB 비트만을 계산하면 다음과 같은 특성을 가진다.

$$A \oplus B|_0 = A|_0 \oplus B|_0$$
$$A \boxplus B|_0 = A|_0 \oplus B|_0$$
$$A \odot B|_0 = A|_0 \oplus B|_0 \oplus 1 \qquad (7)$$

식(7)에 의하여, IDEA알고리즘의 분석을 LSB비트만을 해석한다면, 세가지 연산이 모두 이진합 연산으로 해석될 수 있다.

$P1 = X1 \odot Z1 = X1 \oplus Z1 \oplus 1,$
$P2 = X2 \boxplus Z2 = X2 \oplus Z2,$
$P3 = X3 \boxplus Z3 = X3 \oplus Z3,$
$P4 = X4 \odot Z4 = X4 \oplus Z4 \oplus 1$
$A = P1 \oplus P3 = X1 \oplus Z1 \oplus 1 \oplus X3 \oplus Z3,$
$B = P2 \oplus P4 = X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus 1$
$T = B \boxplus (A \odot Z5) \odot Z6$
$= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus 1 \oplus X1 \oplus Z1 \oplus 1 \oplus X3 \oplus Z3 \oplus Z5 \oplus 1 \oplus Z6 \oplus 1$
$= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus X1 \oplus Z1 \oplus X3 \oplus Z3 \oplus Z5 \oplus Z6$
$S = A \odot Z5 \boxplus T$
$= X1 \oplus Z1 \oplus 1 \oplus X3 \oplus Z3 \oplus Z5 \oplus 1 \oplus X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus X1 \oplus Z1 \oplus X3 \oplus Z3 \oplus Z5 \oplus Z6$
$= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus Z6$

따라서 IDEA 1단의 출력에 관한 식은 다음과 같다.

$Y1 = T \oplus P1 = T \oplus (X1 \odot Z1)$
$= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus X1 \oplus Z1 \oplus X3 \oplus Z3 \oplus Z5 \oplus Z6 \oplus X1 \oplus Z1 \oplus 1$
$= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus X3 \oplus Z3 \oplus Z5 \oplus Z6 \oplus 1$
$= X2 \oplus X3 \oplus X4 \oplus Z2 \oplus Z3 \oplus Z4 \oplus Z5 \oplus Z6 \oplus 1$

$$
\begin{aligned}
Y2 &= T \oplus P3 = T \oplus (X3 \boxplus Z3) \\
&= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus X1 \oplus Z1 \oplus X3 \oplus \\
&\quad Z3 \oplus Z5 \oplus Z6 \oplus X3 \oplus Z3 \\
&= X1 \oplus X2 \oplus X4 \oplus Z1 \oplus Z2 \oplus Z4 \oplus Z5 \oplus Z6 \\
Y3 &= S \oplus P2 = S \oplus (X2 \boxplus Z2) \\
&= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus Z6 \oplus X2 \oplus Z2 \\
&= X4 \oplus Z4 \oplus Z6 \\
Y4 &= S \oplus P4 = S \oplus (X4 \odot Z4) \\
&= X2 \oplus Z2 \oplus X4 \oplus Z4 \oplus Z6 \oplus X4 \oplus Z4 \oplus 1 \\
&= X2 \oplus Z2 \oplus Z6 \oplus 1
\end{aligned} \tag{8}
$$

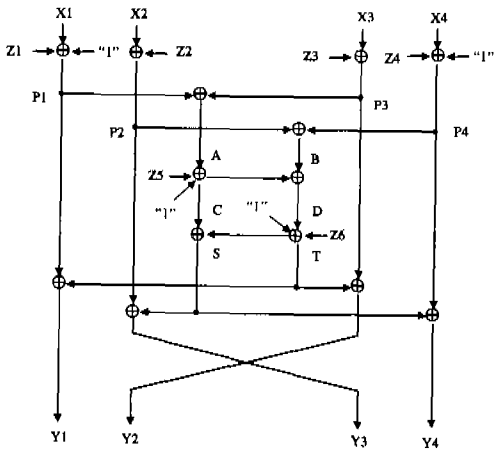


그림 3. IDEA 알고리즘(1라운드)의 선형동가 블록도

위의 식의 결과를 보면, 1 라운드의 결과 암호문은 각각 3개의 입력문과 2개 이상의 키 계열과 관련됨을 알 수 있다. 여기서부터는 편의상 1단의 출력을 y11, y12, y13, y14라 표시하고, 2단의 출력을 y21, y22, y23, y24라 표시하고 최종 8.5단의 출력을 Y1, Y2, Y3, Y4라 표시하기로 한다.

$$
\begin{aligned}
y11 &= X2 \oplus X3 \oplus X4 \oplus Z2 \oplus Z3 \oplus Z4 \oplus Z5 \oplus Z6 \oplus 1 \\
y12 &= X1 \oplus X2 \oplus X4 \oplus Z1 \oplus Z2 \oplus Z4 \oplus Z5 \oplus Z6 \\
y13 &= X4 \oplus Z4 \oplus Z6 \\
y14 &= X2 \oplus Z2 \oplus Z6 \oplus 1
\end{aligned}
$$

이를 2단에 적용하면 다음과 같은 결과를 얻을 수 있다.

$$
\begin{aligned}
y21 &= X1 \oplus Z1 \oplus Z5 \oplus Z6 \oplus Z8 \oplus Z9 \oplus Z10 \\
&\quad \oplus Z11 \oplus Z12 \\
y22 &= X1 \oplus X2 \oplus X3 \oplus Z1 \oplus Z2 \oplus Z3 \oplus Z6 \\
&\quad \oplus Z7 \oplus Z8 \oplus Z10 \oplus Z11 \oplus Z12 \\
y23 &= X2 \oplus Z2 \oplus Z6 \oplus Z10 \oplus Z12 \oplus 1 \\
y24 &= X1 \oplus X2 \oplus X4 \oplus Z1 \oplus Z2 \oplus Z4 \oplus Z5 \\
&\quad \oplus Z6 \oplus Z8 \oplus Z12 \oplus 1
\end{aligned}
$$

이러한 결과를 전체 8.5단 IDEA 알고리즘에 적용하면, 최종 출력 Y1, Y2, Y3, Y4는 다음과 같다.

$$
\begin{aligned}
Y1 &= X1 \oplus Z1 \oplus Z5 \oplus Z6 \oplus Z8 \oplus Z9 \oplus Z10 \oplus Z11 \oplus \\
&\quad Z12 \oplus Z13 \oplus Z17 \oplus Z18 \oplus Z20 \oplus Z21 \\
&\quad \oplus Z22 \oplus Z23 \oplus Z24 \oplus Z25 \oplus Z29 \oplus Z30 \\
&\quad \oplus Z32 \oplus Z33 \oplus Z34 \oplus Z35 \oplus Z36 \oplus Z37 \\
&\quad \oplus Z41 \oplus Z42 \oplus Z44 \oplus Z45 \oplus Z46 \oplus Z47 \\
&\quad \oplus Z48 \oplus Z49 \oplus 1 \\
Y2 &= X2 \oplus Z2 \oplus Z6 \oplus Z10 \oplus Z12 \oplus Z15 \oplus Z17 \\
&\quad \oplus Z18 \oplus Z19 \oplus Z21 \oplus Z22 \oplus Z24 \oplus Z25 \\
&\quad \oplus Z26 \oplus Z27 \oplus Z30 \oplus Z31 \oplus Z32 \oplus Z34 \\
&\quad \oplus Z35 \oplus Z36 \oplus Z38 \oplus Z42 \oplus Z46 \oplus Z48 \\
&\quad \oplus Z50 \oplus 1 \\
Y3 &= X1 \oplus X2 \oplus X3 \oplus Z1 \oplus Z2 \oplus Z3 \oplus Z6 \oplus Z7 \\
&\quad \oplus Z8 \oplus Z10 \oplus Z11 \oplus Z12 \oplus Z14 \oplus Z18 \\
&\quad \oplus Z22 \oplus Z24 \oplus Z27 \oplus Z29 \oplus Z30 \oplus Z31 \\
&\quad \oplus Z33 \oplus Z34 \oplus Z36 \oplus Z37 \oplus Z38 \oplus Z39 \\
&\quad \oplus Z42 \oplus Z43 \oplus Z44 \oplus Z46 \oplus Z47 \oplus Z48 \\
&\quad \oplus Z51 \\
Y4 &= X1 \oplus X2 \oplus X4 \oplus Z1 \oplus Z2 \oplus Z4 \oplus Z5 \oplus Z6 \\
&\quad \oplus Z8 \oplus Z12 \oplus Z16 \oplus Z18 \oplus Z21 \oplus Z23 \\
&\quad \oplus Z24 \oplus Z25 \oplus Z27 \oplus Z28 \oplus Z30 \oplus Z31 \\
&\quad \oplus Z32 \oplus Z33 \oplus Z36 \oplus Z37 \oplus Z38 \oplus Z40 \\
&\quad \oplus Z41 \oplus Z42 \oplus Z44 \oplus Z48 \oplus Z52
\end{aligned}
$$

Y1, Y2, Y3, Y4를 다시 128 비트의 키 계열에 의해 생성된 라운드 키에 해당되는 LSB로 표 1 에 의하여 정리하면 다음과 같다.

$$
\begin{aligned}
Y1 &= X1 \oplus b15 \oplus b79 \oplus b95 \oplus b127 \oplus b40 \\
&\quad \oplus b56 \oplus b72 \oplus b88 \oplus b104 \oplus b65 \oplus b81 \\
&\quad \oplus b113 \oplus b1 \oplus b17 \oplus b33 \oplus b49 \oplus b90 \\
&\quad \oplus b26 \oplus b42 \oplus b74 \oplus b115 \oplus b3 \oplus b19 \\
&\quad \oplus b35 \oplus b51 \oplus b12 \oplus b28 \oplus b60 \oplus b76 \\
&\quad \oplus b92 \oplus b108 \oplus b124 \oplus b37 \oplus 1 \\
Y2 &= X2 \oplus b31 \oplus b95 \oplus b56 \oplus b88 \oplus b8 \\
&\quad \oplus b65 \oplus b81 \oplus b97 \oplus b1 \oplus b17 \oplus b49 \\
&\quad \oplus b90 \oplus b106 \oplus b122 \oplus b42 \oplus b58 \oplus b74 \\
&\quad \oplus b3 \oplus b19 \oplus b35 \oplus b67 \oplus b28 \oplus b92 \\
&\quad \oplus b124 \oplus 53 \oplus 1
\end{aligned}
$$

Y3 = X1⊕X2⊕X3⊕b15⊕b31⊕b47⊕b95
⊕b111⊕b127⊕b56⊕b72⊕b88⊕b120
⊕b81⊕b17⊕b49⊕b122⊕b26⊕b42
⊕b58⊕b115⊕b3⊕b35⊕b51⊕b67
⊕b83⊕b28⊕b44⊕b60⊕b92⊕b108
⊕b124⊕b69

Y4 = X1⊕X2⊕X4⊕b15⊕b31⊕b65⊕b79
⊕b95⊕b127⊕b88⊕b24⊕b81⊕b1
⊕b33⊕b49⊕b90⊕b106⊕b10⊕b42
⊕b58⊕b74⊕b115⊕b35⊕b51⊕b67
⊕b99⊕b12⊕b28⊕b60⊕b124⊕b85

결과적으로 출력계열(Y1,Y2,Y3,Y4)의 각 LSB 비트는 입력계열(X1,X2,X3,X4)의LSB비트와 128 비트의 키 계열중 각각 위의 비트들과 연관성을 가짐을 알 수 있다.

Ⅳ. 결 론

전 장에서 설명된 바와 같이 MA 구조를 배제한 상태에서 MSB에 의한 차분분석 방법은 키계열 Z1, Z4에 대한 (-)1 가정을 두고, IDEA 함수를 분석하였다. 표 3 에서와 같이 4 라운드의 IDEA 알고리즘이 사용되었다면 128 비트의 키 계열 중에서 사용된 비트 수는 75 비트이며, 사용되지 않은 키 비트 수는 53 비트이다. 마지막으로 7-8 라운드에서 12 비트가 추가되고, 마지막 8.5라운드에서 10 비트가 추가됨을 알 수 있다 8.5 라운드까지 전체 알고리즘을 고려할 때, 128 비트의 키 계열 중에서 사용되지 않은 키 비트수는 12비트이다.

또한 LSB를 이용한분석방법에서는입력계열(X1, X2,X3,X4)의 각 LSB 비트와 출력계열(Y1,Y2,Y3, Y4)의 각 LSB 비트간에는 키 비트들의 이진합으로 체크될 수 있음을 보였다.

이러한 결과는 현재 사용되고 있는 IDEA 알고리즘의 안전성에 관한 평가분야에 영향을 줄 것으로 기대되며, 더욱 더 안전한 블록 암호시스템을 제안하기 위한 기초자료로서 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] J.Daemen and R.Govaerts and J.Vandewalle, "Weak Keys for IDEA", *Crypto'93*, pp224-231, 1993.

[2] W.Meier, "On the Security of the IDEA Block Cipher", *Eurocrypto'93*, p.371-385, 1993.

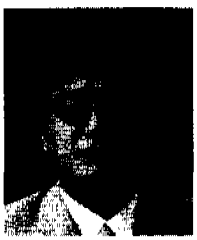
[3] X.Lai and J.LMassey," A Proposal for a New Block Encryption Standard, *Advances in Cryptology-Eurocrypt '90*, Springer-Verlag, Berlin 1991, pp 389-404.

[4] M.Y.Rhee, *Cryptography and Secure Communications*, McGraw-Hill, 1993.

[5] B.Schneier, *Applied Cryptography : Protocols, Algorithms, and Source Code in C*, Wiley, 1994.

[6] W.Stallings, *Network and Internetwork Security*, Prentice Hall, Sydney, 1989.

김 지 홍(Ji-hong Kim) 정회원

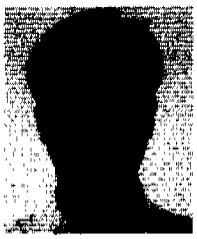


1982년 2월 : 한양대학교 전자공학과 졸업(공학사)
1984년 2월 : 한양대학교 대학원 전자통신공학과 (공학석사)
1996년 2월 : 한양대학교 대학원 전자통신공학과 (공학박사)
1991년 3월~현재: 세명대학교 전자공학과 부교수
<주관심 분야> 부호이론, 공개키기반구조

윤 석 찬(Suck-Chang Yun) 정회원

1975년 2월 : 한양대학교 전자공학과 졸업(공학사)
1977년 8월 : 성균관대학교 대학원 전기공학과 (공학석사)
1988년 2월 : 성균관대학교 대학원 전자공학과 (공학박사)
1991년 3월~현재:세명대학교 전자공학과 부교수
<주관심 분야> 암호이론, 정보보호

장 영 달(Young-dal Jang)



1998년 2월 : 세명대학교
전자공학과 졸업(공학사)
1998년 2월~현재 : 세명대학교
대학원 전기전자공학부
석사과정
<주관심 분야> 공개키기반구조,
상호인증