情報保護學會論文誌
第14券 第4號, 2004. 8

수리계획법을 이용한 S-box의 부울함수 합성

송 정 환[†], 구 본 욱[‡]
한양대학교

Synthesizing a Boolean Function of an S-box with Integer Linear Programming

JungHwan Song[†], Bon Wook Koo[‡]
Hanyang University

요 약

공개되지 않은 함수에 대한 입력과 그에 따른 출력을 이용하여 그 함수와 같은 입출력을 가지는 부울함수표현을 찾아내는 것이 부울함수 합성문제이다. 전자공학 및 암호학 분야에서는 이 문제가 수리계획법의 한 부류인 0-1 integer programming 문제로 귀결되며, 본 논문에서는 부울함수 합성문제를 해결하는 하나의 예로 DES 의 비공개 논리인 입력 6비트, 출력 4비트의 S-box에 대한 부울함수표현을 찾는다. 이러한 결과는 임의의 함수에 대한 효율적인 하드웨어 구현과 블록암호 알고리즘의 대수적 구조를 이용한 암호분석기법에 이용될 수 있다.

ABSTRACT

Boolean function synthesize problem is to find a boolean expression with in/outputs of original function. This problem can be modeled into a 0-1 integer programming. In this paper, we find a boolean expressions of S-boxes of DES for an example, whose algebraic structure has been unknown for many years. The results of this paper can be used for efficient hardware implementation of a function and cryptanalysis using algebraic structure of a block cipher.

Keywords : *S-box, Boolean function.*

Ⅰ. 서 론

암호기술에 사용되는 함수들은 모두 부울함수의 형태로 표현가능하다. 또한 임의의 함수에 대한 부울함수 표현은 소프트웨어 구현뿐만 아니라 하드웨어 구현시 효율성 측면에서 매우 중요한 역할을 한다.

동일한 입출력을 가지는 함수라고 할지라도, 그 구현방법에 따른 효율성과 제작비용의 차이는 매우 크다. 따라서, 임의의 수학적 논리로 이루어진 함수를 가장 효율적이고, 제작비용이 적게 드는 부울함수 형태로 전환하는 작업은 반드시 필요하다. 또한, 암호학 분야에서는 암호알고리즘 세부논리에 대한 부울함수 표현을 대수공격 등 각종 분석법에 적용할 수 있다.

Ⅱ. 부울대수

2.1. 기본적인 정의 , 성질[1]

본 절에서는 부울대수에서 사용되는 연산과 부울함수를 정의하고 부울함수의 성질에 대해서 간략히 언급한다.

* 본 연구는 2001년 한양대학교 지원으로 수행되었습니다.
** 본 연구는 2001년 고려대학교 지원으로 수행되었습니다.
† 주저자 : camp123@hanyang.ac.kr
‡ 교신저자 : kidkoo@ihanyang.ac.kr

정의 1) (부울연산)

집합 {0,1} 에서의 부울 연산 +($\vee$), ·($\wedge$), ′(¬) 을 다음과 같이 정의한다.

$0+0 = 0 ; 0+1 = 1+0 = 1+1 = 1$

$0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0 ; 1 \cdot 1 = 1$

$0' = 1 ; 1' = 0$

정의 2) (부울함수)

임의의 양의정수 n, m 에 대하여, 함수 f가 $f: \{0,1\}^n \rightarrow \{0,1\}^m$인 경우 부울함수 라고 부른다.

정의 3)

부울함수 f가 n개의 변수 $x_1, x_2, \ldots, x_n$을 가질 때,

a) 각각의 항 x_i, x_i' $(1 \leq i \leq n)$ 를 리터럴 (literal) 이라고 부른다.

b) x_i 혹은 x_i' $(1 \leq i \leq n)$ 의 곱으로만 이루어진 항을 디스정트(disjunct) 라고 부른다.

c) Disjunct 들의 합으로 부울함수 f를 표현한 경우 f의 disjunctive normal form (d.n. f)이라고 부른다.

d) f의 정의역의 원소를 민텀(minterm) 이라고 부르며, f가 $f: \{0,1\}^n \rightarrow \{0,1\}$ 인 경우, $\{x \in \{0,1\}^n \mid f(x)=1\}$을 ON-set, $\{x \in \{0,1\}^n \mid f(x)=0\}$을 OFF-set 이라고 한다.

위와 같이 정의된 부울함수는 다음과 같은 성질을 만족한다.

정리 4)

f, g, h 를 출력이 1 비트인 부울함수 라고 하고, x, y, z 를 literal 이라고 할 때, 다음성질을 만족한다.

a) $(f')' = f$

b) $(f+g)' = f'g'$
$f'g' = (f+g)'$

c) $f+g = g+f$
$fg = gf$

d) $f+(g+h) = (f+g)+h$
$f(gh) = (fg)h$

e) $f+gh = (f+g)(f+h)$
$f(g+h) = fg+fh$

f) $f+f = f$
$ff=f$

e) $f+0 = f$
$f \cdot 1 = f$

f) $f+f' =$
$ff' = 0$

g) $f+1 = 1$
$f \cdot 0 = 0$

h) $f+fg = f$
$f(f+g) = f$

2.2. 부울함수 합성문제

부울함수 합성문제는 입출력 비트들을 이용하여 원래의 함수와 동일한 입출력을 가지는 가장 간단한 부울함수 표현을 유추해내는 문제이다.[4,5]

부울함수 합성문제는 다음과 같이 크게 두 가지로 분류된다.

- Deductive inference problem : 함수 정의역의 원소의 개수와 같은 개수의 입출력 데이터를 이용하여 합성하는 문제
- Inductive inference problem : 함수 정의역의 원소의 개수보다 적은 개수의 입출력 데이터를 이용하여 합성하는 문제

특히, Inductive inference problem을 해결하여 합성된 부울 함수를 불완전하게 정의된 부울 함수라고 하며, 이는 ON-set과 OFF-set 원소의 개수의 합이 정의역의 원소보다 적은 경우를 말한다. 즉, $|ON-set| + |OFF-set| < 2^n$이다.

불완전하게 정의된 부울 함수도 대수적으로 부울합과 부울곱 연산으로 표현가능하며, ON-set에 해당되는 모든 minterm들에 대한 함수 값이 1이 되고, OFF-set에 해당되는 모든 minterm들에 대한 함수 값이 0이 되며, 그리고 그 밖의 minterm들에 대한 함수 값이 1 혹은 0이 되도록 표현할 수 있다. 모든 부울함수는 부울곱들의 부울합 형태로 표현될 수 있다.

Ⅲ. 출력이 1비트인 부울함수 합성문제

출력이 1비트인 부울함수를 3-Layer System을 이용하여 합성하는 문제는 3-Layer System의 특성을 이용하여 도출한 방정식 또는 부등식에 대한 해집합을 구하는 문제로 변환 될 것이며, 변환된 문제의 해는 수리계획적 방법으로 찾을 수 있다.

3.1. 3-Layer System

입력이 n 비트이고, 출력이 1 비트인 임의의 부울함수 F를 고려한다.

$F: \{0, 1\}^n \rightarrow \{0, 1\}$

F는 다음과 같이 n개의 부울변수로 이루어진 K개의 disjunct들의 논리합으로 표현될 수 있다.

$$C_1 \vee C_2 \vee \cdots \vee C_K$$

또한, 위 부울함수 표현은 언제나 아래 그림과 같은 3-Layer system 으로 꾸며질 수 있다.

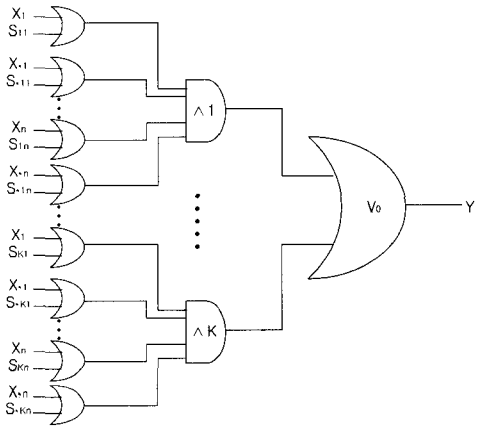


그림 1. 출력이 1비트인 3-Layer System

위 그림은 n개의 부울변수 $x_1, \cdots, x_n$을 입력으로 하고 출력이 y가 되는 부울함수를 K개의 논리곱 $\wedge_1, \cdots, \wedge_K$과 그 출력을 입력으로 하는 하나의 논리합 $\vee_0$으로 구성한 것이다.

각각의 논리곱 $\wedge_j$, $j=1,\cdots,K$의 입력은 $2n$개의 논리합 $\vee^l_{ji}$, $i=1,\cdots,n$, $l=0,1$의 결과이다. $\vee^0_{ji}$의 입력은 부울변수 x_i'와 결정변수 s_{ji}'이고 출력은 게이트 $\wedge_j$의 입력이 된다. 마찬가지로 $\vee^1_{ji}$의 입력은 부울변수 x_i와 결정변수 s_{ji}이고 출력은 게이트 $\wedge_j$의 입력이 된다. 여기서 결정변수 s_{ji}와 s_{ji}'는 각각 x_i와 x_i'가 j번째 disjunct에 참여한 경우를 나타내는 결정변수로서 다음과 같이 정의된다.

$$s_{ji} = \begin{cases} 0 & x_i \text{ 가 j 번째 disjunct 에 있을 경우} \\ 1 & x_i \text{ 가 j 번째 disjunct 에 없을 경우} \end{cases}$$

$$s'_{ji} = \begin{cases} 0 & x'_i \text{ 가 j 번째 disjunct 에 있을 경우} \\ 1 & x'_i \text{ 가 j 번째 disjunct 에 없을 경우} \end{cases}$$

합성하고자 하는 부울함수는 K개의 disjunct를 가진 함수이고 각각의 disjunct에는 x_i 또는 x_i' $(i=1,2,\ldots n)$가 참여하거나 또는 참여하지 않고 있다.

임의의 i에 대해 x_i와 x_i'가 동시에 같은 disjunct에 참여하는 경우는 0이 되므로 의미 없는 disjunct가 된다. 이런 경우는 없다고 가정한다.

위의 그림 1 은 합성하고자 하는 부울함수를 일반화 하여 나타내고 있다. 여기서 s_{ji} 와 s'_{ji} 는 x_i 또는 x_i' 가 j 번째 disjunct 에 참여하는 경우에는 i 번째 논리합 $\vee_i$ 의 결과가 x_i 또는 x_i'이 되도록 0을 반납하고, x_i 또는 x_i'가 참여하지 않는 경우에는 j번째 disjunct의 결과에 영향을 주지 않도록 1을 반납하도록 정의되었다.

x_i와 x_i'가 같은 disjunct 에 동시에 참여하는 경우가 없다는 가정에 의해 모든 입력에 대하여 다음 조건을 만족해야 한다.

$$s_{ji} \vee s'_{ji} = 1 \ , \ i=1,\cdots,n, \ j=1,\cdots,K \qquad (1)$$

다음은 ON-set 과 OFF-set 의 원소에 의해 도출되는 성질들을 살펴보기로 하자.

3.2. OFF-set 의 원소에 의한 조건

$R = |OFF-set|$이라 하고, 임의의 원소 $w \in OFF-set$에 대해 $U_r = \{i \mid x_i$가 w에 포함되고 $x_i = 1\}$ $U_r = \{i \mid x_i$가 w에 포함되고 $x_i = 0\}$ $r = 1,2,\ldots,R$이라고 정의하면, OFF-set 의 원소를 입력하였을 때의 결과는 언제나 $y=0$이어야 하므로 모든 $j=1,2,\ldots,K$에 대하여 게이트 $\wedge_j$의 출력은 반드시 0이 되어야 한다. 각각의 j에 대하여 $\wedge_j$의 출력이 0이 되어야 하기 때문에 적어도 하나의 $\vee^l_{ji}$, $i=1,2,\ldots,n$, $l=0,1$의 출력이 0이 되어야 한다.

OFF-set 의 원소에 대해 $x_i=1$인 경우에는 각각의 disjunct에 $x_i'(i \in U_r)$ 중 적어도 하나는 참여하거나 또는 $x_i=0$인 i에 대해서는 각각의 disjunct에 $x_i(i \in U_r')$ 중 하나는 참여하여야 한다.

위 결과를 수식으로 표현하면 다음과 같다.

$$(\bigwedge_{i\in U_r} s'_{ji}) = 0 \text{ or } (\bigwedge_{i\in U'_r} s_{ji}) = 0$$
$$j=1,2,\ldots,K, \quad r=1,2,\ldots,R$$

따라서,

$$(\bigvee_{i\in U_r} \neg s'_{ji}) \vee (\bigvee_{i\in U'_r} \neg s_{ji}) = 1 \qquad (2)$$
$$j=1,2,\ldots,K, \quad r=1,2,\ldots,R$$

이 만족되어야 한다.

3.3. ON-set minterm에 의한 조건

$A = |ON-set|$이라 하고, ON-set의 원소를 $a_p(p=1,2,\ldots,A)$라고 하자.

모든 ON-set 원소의 입력에 대하여 출력 y는 언제나 $y=1$이어야 하므로 적어도 하나의 j에 대하여 $\wedge_j$의 출력은 반드시 1 이 되어야 한다. 임의의 minterm a_p에 대한 게이트 $\wedge_j$의 출력을 $x_j^{a_p}$ 라고 하고 a_p에 대하여 게이트 $\vee_{ji}^{l}$ $i=1,2,\ldots,n$, $l=0,1$의 출력들을 생각하면, a_p의 x_i가 1인 경우 게이트 $\vee_{ji}^{1}$의 출력은 1이고 $\vee_{ji}^{0}$의 출력은 s'_{ji}가 되어야 한다. 마찬가지로 x_i 가 0인 경우 게이트 $\vee_{ji}^{0}$ 의 출력은 1이고 $\vee_{ji}^{1}$의 출력은 s_{ji}가 되어야 한다. $\sigma_{ji}^{a_p}$ 를 다음과 같이 정의하면

$$\sigma_{ji}^{a_p} = \begin{cases} s_{ji} & a_p\text{에서 } x_i=0\text{인 경우} \\ s'_{ji} & a_p\text{에서 } x_i=1\text{인 경우} \end{cases}$$

부울변수 $x_j^{a_p}$와 $\sigma_{ji}^{a_p}$간에는 다음과 같은 식이 성립한다.

$$x_j^{a_p} = \bigwedge_{i=1}^{n} \sigma_{ji}^{a_p}, \quad j=1,2,\ldots,K, \quad p=1,2,\ldots,A$$

또한 적어도 하나의 게이트 $\wedge_j$에 대하여 $x_j^{a_p}=1$이 되려면

$$\bigvee_{j=1}^{K} x_j^{a_p} = 1, \quad p=1,2,\ldots A \qquad (3)$$

가 만족되어야 한다.

또한, 어떠한 $i=1,\ldots n$에 대해서도 $x_j^{a_p}=1$, $\sigma_{ji}^{a_p}=0$이 동시에 만족될 수 없으므로

$$\sigma_{ji}^{a_p} \vee \neg x_j^{a_p} = 1, \quad i=1,2,\ldots,n, \qquad (4)$$
$$j=1,2,\ldots,K, \quad p=1,2,\ldots,A$$

가 만족되어야 한다.

3.4. 출력이 1비트인 경우의 수리계획적 모델링

위에서 구한 식 (1)~(4) 에 대한 해집합을 구하기 위한 방법인 0-1 Integer Programming에 대해서 알아보고, 0-1 Integer Programming에 알맞도록 모델링 하여 문제를 해결한다.

3.4.1. 0-1 Integer Programming

우리가 해결할 문제는 m개의 선형부등식을 만족하는 이진벡터 $x=(x_1,\cdots,x_n)^T$를 찾는 문제이다.

정의 5)(ILP)

일반적으로 Integer Linear Programming (ILP)은 다음과 같이 정의된다.

$$\text{(ILP)} \quad \begin{array}{ll} \max & f^T x \\ s.t & A^T x \le b \\ & x \in \{0,1\}^n \end{array} \qquad (5)$$

여기서,

$$A^T = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$
$$a_i \in \mathbf{R}^n, \quad b_i \in \mathbf{R}, \quad i=1,\cdots,m$$

이다.[2,3]

본 논문에서 다루는 ILP는 목적함수 $f^T x$와 무관하게, 해집합 $\{ x \in \{0,1\}^n \mid A^T x \le b \}$을 찾는 것이 목적이지만, 목적함수의 적절한 설정으로 인해

대수적 차수가 가장 낮은 함수, 게이트 수가 적은 함수, disjunct 수가 적은 함수 등 여러 가지 형태의 함수를 합성할 수 있다.

만일 임의의 $x \in \{0,1\}^n$가 (5)의 해(feasible solution)라면 부울함수 합성문제는 Satisfiable 하다.

ILP를 해결하는 알고리즘들에는 Branch and Bound, Lagrangian relaxation[2,3,4] 그리고 interior point method를 이용한 방법[5,6]들이 있다.

3.4.2. 부울함수 합성문제에의 적용

위에서 소개한 출력이 1 비트일 경우의 부울함수 합성문제는 다음과 같은 ILP 의 해집합을 구하는 문제로 변형될 수 있다

n개의 literal $x_1, \ldots, x_n$로 구성된 m개의 disjunct $C_1, C_2, \ldots, C_m$와 그것으로 구성된 3-Layer System을 고려하고 다음을 정의하자.

$I_C = \{ i \mid x_i$ 가 *disjunct* C 에 포함 $\}$

$J_C = \{ i \mid x_i'$ 가 *disjunct* C 에 포함 $\}$

각 disjunct C가 1이려면 I_C 혹은 J_C에 포함된 인수에 해당되는 적어도 하나의 literal이 1이 되어야 한다. 그러므로,

$$\sum_{j \in I_C} x_j - \sum_{j \in J_C} x_j' \geq 1 \ , \ C = C_1, \ldots, C_m$$

이다.

여기서, x_j'를 $1 - x_j$로 바꾸면 다음과 같이 변형된다.

$$\sum_{j \in I_C} x_j - \sum_{j \in J_C} x_j \geq 1 - |J_C|, \ C = C_1, \ldots, C_m$$

$x_1, \ldots x_n, s_{11}, s_{11}', \cdots, s_{Kn}, s_{Kn}', x_1^a, \ldots x_n^a$를 이용하여 3.1. 의 부울함수 합성문제를 Integer linear programming으로 변형하면 다음과 같다.

$$s_{ji} + s'_{ji} \geq 1 \ , \ i = 1, \ldots n \ , \ j = 1, \ldots K \quad (6)$$

$$\sum_{i \in U_r} s_{ji}' + \sum_{i \in U_r'} s_{ji} \leq n - 1, r = 1, \ldots R, j = 1, \ldots K \quad (7)$$

$$\sum_{j=1}^{K} x_j^{a_p} \geq 1 \ , \ p = 1, \ldots A \quad (8)$$

$$x_j^{a_p} \leq \sigma_{ji}^{a_p}, i = 1, \ldots n, j = 1, \ldots K, p = 1, \ldots, A \quad (9)$$

위 부등식들을 제약조건으로 하는 ILP 의 임의의 한 해를 구하고, 구해진 해의 $s_{11}, s_{11}', \cdots, s_{Kn}, s_{Kn}'$의 값들을 이용하면, 결정변수의 정의에 의해 함수를 합성할 수 있다.

3.5. 계산복잡도

제약조건의 개수와 그에 사용되는 변수의 개수를 이용하여 입력변수의 개수와 합성될 부울함수의 형태에 따른 계산복잡도를 추정한다.

ON-set의 원소의 개수와 OFF-set의 원소의 개수는 함수마다 다를 수 있지만, 암호알고리즘에서 사용되는 부울함수들은 일반적으로 regularity[7]를 만족하므로,

$$|ON-set| = |OFF-set| = 2^{(n-1)}$$

이라고 가정한다.

사용되는 변수는 s_{ji}, s_{ji}', $x_j^{a_p}$, $\sigma_{ji}^{a_p}$이다. 각 변수의 종류별 개수는 다음과 같다.

표 1. 문제의 규모에 따른 변수의 개수

변 수	s_{ji}, s_{ji}'	$x_j^{a_p}$	$\sigma_{ji}^{a_p}$
개 수	$2nK$	$K \cdot 2^{n-1}$	$nK \cdot 2^{n-1}$

식 (6),(7),(8),(9) 에 의한 제약조건의 개수는

$$nK + 2^{n-1}K + 2^{n-1} + 2^{n-1}nK$$
$$= nK + 2^{n-1}((n+1)K + 1)$$

이다.

여기서 n은 입력변수의 개수이고, K는 합성하고자 하는 함수의 disjunct 의 개수이다.

따라서 예를 들어 입력의 크기가 6 비트이고, 출력이 1 비트인 함수를 10개의 disjunct를 가지는 부울함수표현으로 나타내려면, 2,360개의 변수로 이루어진 2,332 개의 제약조건을 만족하는 해를 구해

야 한다.

Ⅳ. 출력이 2비트 이상인 부울함수 합성문제

블록암호 알고리즘에서 실제로 사용되고 있는 부울함수들은 그 출력의 크기가 2비트 이상인 경우가 대부분이다. 따라서 실제 사용되는 함수에 대한 부울함수표현을 알아내기 위해서는 출력이 1 비트인 경우의 부울함수 합성문제를 출력의 비트 수만큼 되풀이 하여 풀어야 한다.

본 논문에서는 출력이 1 비트인 경우의 부울함수 합성문제의 되풀이 하여 출력이 여러비트인 부울함수를 합성하는 방법을 소개한다.

4.1. 출력이 2 비트인 3-Layer System

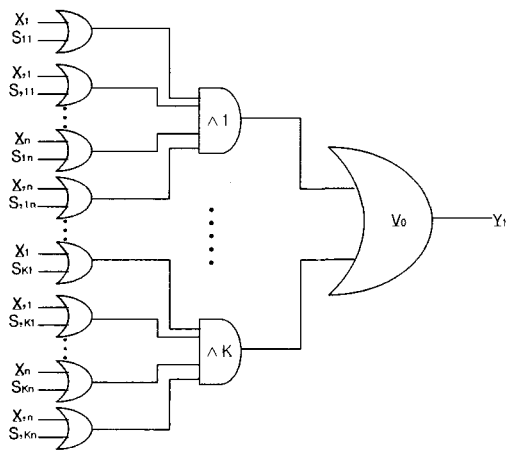
부울함수 F를 $F:\{0,1\}^n \rightarrow \{0,1\}^2$ 라고 하자.

다음과 같이 출력이 1 비트인 두개의 부울함수로 나누어 생각할 수 있다.

$$F = \begin{cases} F_1 : \{0,1\}^n \rightarrow \{0,1\} \\ F_2 : \{0,1\}^n \rightarrow \{0,1\} \end{cases}$$

F_1, F_2를 각각 출력이 1 비트인 부울함수로 생각하면, 다음 그림과 같이 3-Layer System 두개를 동시에 고려하면 된다.

F_1



F_2

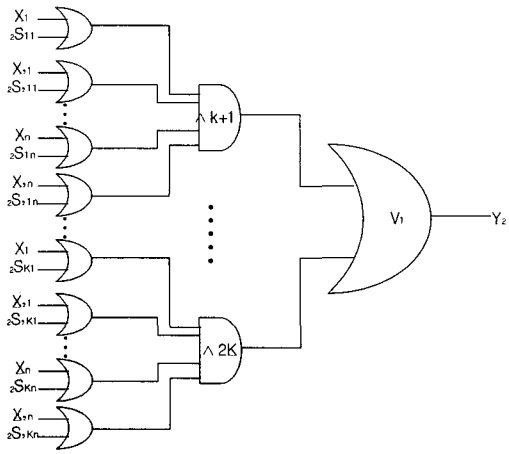


그림 2. 출력이 2 bit인 3-Layer System

출력이 2 비트 이상인 부울함수 합성문제의 계산복잡도는 출력이 1 비트인 부울함수 합성문제의 계산복잡도의 정수배 이다.

Ⅴ. DES S-box의 부울함수 합성문제

잘 알려진 블록암호 알고리즘 중 하나인 DES 에 사용된 비선형 함수인 S-box 에 대한 부울함수 표현을 찾아봄으로써 위에서 제시한 수리계획법을 이용한 부울함수 합성문제의 해결가능성 및 암호 분석기술로의 적용가능 여부를 확인해 보았다.

DES의 S-box[8] 는 입력이 6 비트 이고, 출력이 4 비트인 함수이다.

DES의 첫 번째 S-box 인 S_1-box의 치환표는 다음과 같다.

표 2. DES S_1-box의 치환표

row	column number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

따라서, DES의 S_1-box 를

$$F : \{0,1\}^6 \rightarrow \{0,1\}^4$$
$$(x_1, x_2, x_3, x_4, x_5, x_6) \mapsto (y_1, y_2, y_3, y_4)$$

라고 하고, F를

$$F_1 : \{0,1\}^6 \rightarrow \{0,1\}$$
$$F_2 : \{0,1\}^6 \rightarrow \{0,1\}$$
$$F_3 : \{0,1\}^6 \rightarrow \{0,1\}$$
$$F_4 : \{0,1\}^6 \rightarrow \{0,1\}$$

로 나누어 고려하였다. 여기서, x_i $(i=1,..,6)$은 입력비트이고, y_i $(i=1,..,4)$는 출력비트이다.

ILP의 제약조건에 포함되는 계수들은 입력비트와 그에 대한 출력비트로 결정되므로, 모든 입력에 대한 출력을 구해보면 표 3 과 같다.

표 3. DES S_1-box의 입 출력표

출 력	입 력
0000	000001, 111011, 011100, 111110
0001	101101, 000110, 100010, 001111
0010	100111, 001000, 001011, 101100
0011	110101, 010000, 111000, 011101
0100	100000, 000010, 000111, 101001
0101	011011, 011000, 111100, 110001
0110	111101, 010011, 010100, 101010
0111	110110, 011110, 101111, 000101
1000	001110, 011111, 100110, 100101
1001	101011, 011001, 110100, 011010
1010	111001, 111010, 010010, 010001
1011	101110, 110011, 010111, 001100
1100	100011, 010110, 010101, 110010
1101	111111, 000100, 101000, 001101
1110	100100, 001001, 110111, 000000
1111	001010, 110000, 000011, 100001

출력의 첫 비트 y_1에 대한 F_1을 구해보자.

$n=6$이고, $|ON-set|=|OFF-set|=2^5$이므로 임의의 K를 정하여 다음 제약조건을 가지는 0-1 Integer programming의 해를 구하면 된다.

$$s_{ji} + s'_{ji} \geq 1 \ , \ i=1,..6 \ , \ j=1,..K \quad (10)$$

$$\sum_{i \in U_r} s_{ji}' + \sum_{i \in U_{r'}} s_{ji} \leq 5, r=1,..32, j=1,..K \quad (11)$$

$$\sum_{j=1}^{K} x_j^{a_p} \geq 1 \ , \ p=1,...32 \quad (12)$$

$$x_j^{a_p} \leq \sigma_{ji}^{a_p}, i=1,..6, j=1,..K, p=1,..,32 \quad (13)$$

K가 매우 크면, 해는 반드시 존재하지만, 계산복잡도가 커지고, K가 매우 작으면 해가 존재하지 않을 수 있으므로, 적절한 K를 선택하는 것이 중요하다. 본 논문에서는 $K=20$을 기본으로, 해가 존재하지 않는 경우 30 으로 확장하였다.

이상으로 해를 구하기 위한 모델링은 모두 끝났다. 실제로 해를 구하는 연산은 ILOG 에서 개발한 M.P. (Mathematical Programming) Solver 인 AMPL[9] 을 이용하였으며, 이는 목적함수와 제약조건이 입력되는 M.P. 형태의 문제를 가장 적합한 방법으로 해결해 주는 Solver 이다.

AMPL 에 의해 구해진 S_1-box의 F_1에 대한 결정변수 s_{ji} , s_{ji}'의 해와 그에 따라 찾아진 부울함수표현은 표 4와 같다.

표 4. S_1-box의 F_1 에 대한 해

	s_{ji}	s_{ji}'
j \ i	1 2 3 4 5 6	1 2 3 4 5 6
1	1 1 0 0 1 1	0 0 1 1 0 1
2	0 1 1 1 0 0	1 1 0 0 1 1
3	1 1 0 1 1 0	0 1 1 0 0 1
4	1 0 1 0 0 0	1 1 1 1 1 1
5	1 1 1 1 0 0	1 0 0 0 1 1
6	1 1 1 1 0 0	1 0 0 0 1 1
7	1 1 1 1 0 0	1 0 0 0 1 1
8	0 1 1 0 1 1	1 0 0 1 0 1
9	1 1 1 1 1 1	0 0 0 1 0 0
10	0 1 1 1 1 0	1 0 0 1 0 1
11	1 1 1 1 0 0	1 0 0 0 1 1
12	1 0 1 1 0 1	1 1 1 0 1 0
13	1 1 0 1 0 1	0 0 1 1 1 0
14	0 0 1 1 1 1	1 1 0 1 0 0
15	1 0 1 0 0 1	0 1 0 1 1 1
16	0 1 1 1 0 0	1 0 1 0 1 1
17	1 0 1 1 1 0	0 1 0 1 0 1
18	1 0 0 1 1 0	1 1 1 0 0 1
19	0 1 1 0 0 1	1 0 1 1 1 0
20	0 1 0 1 1 1	1 0 1 0 0 0

$$y_1 = x_3x_4x_1'x_2'x_5' \vee x_1x_5x_6x_3'x_4' \vee x_3x_6x_1'x_4'x_5' \vee x_2x_4x_5x_6 \vee x_5x_6x_2'x_3'x_4' \vee x_1x_4x_2'x_3'x_5' \vee x_1'x_2'x_3'x_5'x_6' \vee x_1x_6x_2'x_3'x_5' \vee x_2x_5x_4'x_6' \vee x_3x_5x_1'x_2'x_6' \vee x_1x_2x_3'x_5'x_6' \vee x_2x_4x_5x_1'x_3' \vee x_1x_5x_6x_2'x_4' \vee x_2x_6x_1'x_3'x_5' \vee x_2x_3x_6x_4'x_5' \vee x_1x_4x_5x_2'x_6' \vee x_1x_3x_2'x_4'x_5'x_6'$$

위에 제시한 해는 식 (10), (11), (12), (13)을 제약조건으로 하는 ILP (5) 에 대한 feasible solution 중 하나이다. 목적함수의 적절한 조정을 통하여 disjunct 의 수가 가장 적은 함수, gate 수가 가장 적은 함수, 대수적 차수가 가장 낮은 함수 등의 최적화 된 형태의 부울함수표현을 찾을 수 있다. 하지만, 본 논문에서는 feasibility 만을 고려하였으므로, 목적함수에 대한 설정은 생략하였다.

다른 함수들에 대한 부울함수 표현을 찾기 위해서 해당함수의 입출력에 대한 *ON−set*과 *OFF−set* 으로 교체하여 하여 다시 시행하였다.

S_1-box의 F_2, F_3, F_4에 대한 부울함수표현은 다음과 같다.

$$y_2 = x_2'x_3'x_5'x_6' \vee x_2x_3x_1'x_4'x_5'x_6' \vee x_1x_2x_4x_5x_6 \vee x_5x_1'x_2'x_4'x_6' \vee x_4x_1'x_3'x_5' \vee x_1x_3'x_4'x_5' \vee x_4x_1'x_3'x_5' \vee x_1x_2'x_4'x_5' \vee x_1x_2x_3x_4x_5' \vee x_2x_4x_5x_1'x_6' \vee x_1x_3x_4x_5x_6 \vee x_5x_6x_1'x_2'x_3' \vee x_1x_2x_5x_3'x_6' \vee x_1x_6x_2'x_3'x_4' \vee x_2x_5x_6x_1'x_4' \vee x_3x_5x_2'x_4'x_6' \vee x_3x_6x_1'x_2'x_5'$$

$$y_3 = x_3x_1'x_2'x_4' \vee x_1x_4x_2'x_5'x_6' \vee x_1x_2x_3x_6x_5' \vee x_1x_3x_5x_2'x_6' \vee x_3x_1'x_2'x_5'x_6' \vee x_2x_1'x_3'x_4' \vee x_2x_1'x_3'x_5'x_6' \vee x_1x_2x_3x_4'x_6' \vee x_2x_3'x_4'x_5'x_6' \vee x_2x_5x_6x_3' \vee x_1x_2x_4x_5x_3' \vee x_5x_6x_1'x_2'x_4' \vee x_2x_3x_4x_6x_5' \vee x_1x_2x_4x_6x_3' \vee x_1x_4x_5x_6x_2' \vee x_1'x_3'x_4'x_5'x_6' \vee x_2x_3x_4x_5x_1'x_6' \vee x_4x_6x_1'x_2'x_3'x_5' \vee x_1x_6x_2'x_3'x_4'x_5'$$

$$y_4 = x_1x_3x_4x_5x_6 \vee x_1x_2x_5'x_6' \vee x_1x_3x_4x_5x_2' \vee x_2x_3x_1'x_4' \vee x_1x_2x_3'x_5' \vee x_2x_4'x_5'x_6' \vee x_3x_4x_6x_2' \vee x_4x_1'x_2'x_5' \vee x_1x_2x_6x_3'x_4' \vee x_2x_3x_6x_1'x_5' \vee x_4x_1'x_2'x_3'x_6' \vee x_3x_5x_1'x_4'x_6' \vee x_1x_6x_3'x_4'x_5' \vee x_2x_3x_5x_1'x_6' \vee x_1x_2x_4x_3'x_6' \vee x_1x_3x_5x_6x_2' \vee x_1x_3x_4'x_5'x_6' \vee x_2x_4x_5x_6x_1'x_3' \vee x_1x_5x_2'x_3'x_4'x_6' \vee x_5x_6x_1'x_2'x_3'x_4'$$

이와 마찬가지 방법으로 다른 28개의 함수에 대한 해를 구하고, 그 해를 이용해 DES S-box 의 부울함수표현을 모두 찾을 수 있었다. (별첨#1)

Ⅵ. 결 론

본 논문에서는 부울함수 표현이 알려지지 않은 임의의 함수에 대해 그 입력과 출력에 관한 정보만을 이용하여 동일한 입출력을 가지는 부울함수표현을 찾아내는 문제를 수리계획법의 한 부류인 0-1 Integer programming을 이용하여 해결하는 방법을 제시하였다. 또한, 그 방법을 적용하여 입력이 6비트인 DES 의 S-box 의 부울함수 표현을 찾아내었다. 이런 결과는 임의의 함수에 대한 최적화된 부울함수표현을 이용하여 효율적인 하드웨어 구현을 가능하도록 할 뿐 아니라, 치환표 형태로 공개된 비선형 암호논리에 대한 부울함수표현을 밝혀내어 대수적 구조를 이용한 암호분석기술에 유용하게 사용될 수 있다.

입출력의 크기가 작은 함수의 경우는 모든 가능한 입출력의 정보를 이용하여 동일한 함수를 합성할 수 있으나, 입출력의 크기가 큰 함수의 경우에는 제한된 입출력의 정보만을 이용하여 함수를 합성한 후 출력이 알려지지 않은 입력에 대한 출력을 추측해 내는 방법에 대한 연구가 향후 연구과제로 남아있다. 입력의 크기가 큰 함수의 부울함수표현을 찾아내는 계산복잡도를 현저하게 줄이는 방법이 연구되어 진다면, 이를 이용하여 수집된 평문과 그에 대한 암호문을 이용하여 임의의 암호문에 대한 평문을 추측해 낼 수 있는 방법에 대한 연구가 가능하게 된다.

참 고 문 헌

[1] Ralph P. Grimaldi, Discrete and combinatorial mathematics an applied introduction, Addison wiley , 1989.

[2] John E. Mitchell, Panos M. Pardalos and Mauricio G.C. Resende, Interior Point methods for combinatorial optimization, Handbook of combinatorial Optimization, Kluwer Academic publishers , 1998.

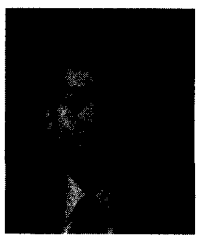
[3] G.L.Nemhauser and L.A. Woolsey, Integer and Combinatorial Optimization, John wiley, New York, 1988.

[4] John E. Mitchell and Mike Todd, "Solving combinatorial optimization problems using Karmarkar's algorithm," Mathematical Programming 56, pp. 245-284, 1992.

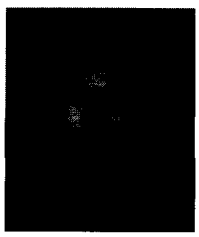
[5] M.Raghavachari, "On connections between zero-one integer programming

under linear constraints," Operations
[6] B.H.Faaland and F.S.Hilier ,"Interior path methods for heuristic integer programming procedures," Operations Research 27, pp.1069-1087, 1979.
[7] Koji Kusuda, Tsutomu Matumoto Strength Evaluation of the Data Encryption Standard, Yokohama National University, October 1993.
[8] NIST "Data Encryption Standard, Federal Information Processing Standards Publication (FIPS PUB)," 46, National Bureau of Standards, Washington, DC, 1977.
[9] Robert Fourer, David M. Gay, Brian W. Kernighan, AMPL A Mmodeling Language For Mathematical Programming, Boyd and Fraser, Massachusetts, 1993.

〈著者紹介〉



송 정 환 (JungHwan Song) 정회원
1984년 2월 : 한양대학교 수학과 졸업
1989년 5월 : Syracuse University Mathematics 석사
1993년 5월 : Rensselaer Polytechnic Institute Mathematics 박사
1999년 3월~현재 : 한양대학교 수학과 조교수
〈관심분야〉 암호학, 최적론, 수리계획법, 정보보호



구 본 욱 (Bon Wook Koo) 학생회원
2001년 2월 : 한양대학교 수학과 졸업
2003년 2월 : 한양대학교 수학과 석사
2003년 2월~2004년 2월 : 한양대학교 자연과학 연구소 연구원
2004년 3월~현재 : 한양대학교 수학과 박사과정
〈관심분야〉 암호학, 정보보호

별첨#1

$S_2 - box$

$$F_1 = x_3x_4x_6x_1'x_2'\vee x_1x_5x_2'x_3'x_6'\vee x_2x_4x_5x_1'x_6'\vee x_1x_2x_4x_5x_6\vee x_3x_5x_1'x_2'x_4'x_6'\vee x_1x_3x_2'x_5'x_6'\vee x_1x_5x_6x_2'x_4'\vee x_4x_1'x_2'x_3'x_6'\vee x_1x_6x_2'x_3'x_5'\vee x_1x_5x_3'x_4'x_6'\vee x_5x_6x_2'x_3'x_4'\vee x_2x_6x_3'x_4'x_5'\vee x_2x_3x_4x_6x_5'\vee x_1'x_3'x_4'x_5'x_6'\vee x_2x_3x_4'x_5'x_6'\vee x_2x_4x_5x_6x_3'\vee x_2x_3x_4x_5x_6'\vee x_3x_6x_1'x_2'x_5'\vee x_2x_3x_5x_6x_1'x_4'\vee x_1x_2x_4x_3'x_5'x_6'$$

$$F_2 = x_1x_2x_4x_3'\vee x_1x_2x_4x_5x_6'\vee x_3x_4x_5x_6x_1'\vee x_1x_6x_2'x_3'x_4'x_5'\vee x_5x_6x_1'x_2'x_3'\vee x_4x_5x_1'x_2'\vee x_2x_5x_1'x_3'x_6'\vee x_5x_6x_1'x_2'x_3'\vee x_3x_1'x_4'x_5'\vee x_1x_2x_4x_3'\vee x_1x_3x_4x_6x_5'\vee x_1x_5x_2'x_4'x_6'\vee x_1x_3x_5x_6x_4'\vee x_1'x_2'x_4'x_5'x_6'\vee x_1x_2x_3'x_5'x_6'\vee x_1x_4x_2'x_5'x_6'\vee x_2x_3x_1'x_5'x_6'\vee x_4x_6x_1'x_2'x_3'\vee x_2x_6x_1'x_4'x_5'\vee x_1x_2x_5x_6x_4'$$

$$F_3 = x_2x_3x_4x_5x_6'\vee x_4x_5x_2'x_3'x_6'\vee x_1x_2x_6x_3'x_5'\vee x_1x_4x_2'x_3'x_5'\vee x_1'x_2'x_4'x_5'\vee x_1x_5x_2'x_3'x_6'\vee x_4x_5x_2'x_3'x_6'\vee x_3x_5x_6x_2'\vee x_3x_1'x_2'x_4'\vee x_1'x_2'x_4'x_5'\vee x_1x_2x_3x_5x_6'\vee x_1x_2x_3x_4x_5'\vee x_1x_4x_5x_3'x_6'\vee x_2x_3x_6x_1'x_5'\vee x_4x_5x_6x_1'x_3'\vee x_1x_2x_6x_3'x_4'\vee x_3x_1'x_2'x_5'x_6'\vee x_2x_5x_1'x_3'x_4'x_6'\vee x_2x_4x_1'x_3'x_5'x_6'$$

$$F_4 = x_2x_4x_6x_1'x_5'\vee x_1x_2x_6x_3'x_5'\vee x_1x_4x_2'x_6'\vee x_1'x_3'x_4'x_6'\vee x_1'x_3'x_4'x_6'\vee x_2x_3x_5x_6\vee x_1x_2x_3x_5\vee x_5x_1'x_2'x_4'x_6'\vee x_4x_5x_6x_2'x_3'\vee x_1x_2x_4'x_5'x_6'\vee x_5x_6x_1'x_2'x_3'\vee x_1x_3x_6x_2'x_4'\vee x_6x_2'x_4'x_5'\vee x_3x_4x_1'x_5'x_6'\vee x_2x_5x_1'x_3'x_6'$$

$S_3 - box$

$$F_1 = x_1'x_2'x_3'x_5'x_6'\vee x_4x_5x_6x_1'x_3'\vee x_1x_4x_5x_3'x_6'\vee x_1x_4x_6x_3'x_5'\vee x_2x_5x_6x_1'x_3'\vee x_1x_3x_4'x_5'x_6'\vee x_4x_1'x_2'x_3'x_6'\vee x_1x_5x_6x_2'x_4'\vee x_1x_4x_6x_2'x_5'\vee x_2x_5x_1'x_3'x_4'\vee x_4x_5x_6x_1'x_2'\vee x_1'x_2'x_3'x_5'x_6'\vee x_1x_3x_5x_4'x_6'\vee x_1x_3x_2'x_4'x_6'\vee x_2x_5x_6x_3'x_4'\vee x_2x_3x_6x_1'x_5'\vee x_1x_4x_6x_2'x_5'\vee x_2x_3x_1'x_4'x_5'\vee x_2x_3x_6x_1'x_4'\vee x_2x_3x_6x_4'x_5'\vee x_4x_1'x_3'x_5'x_6'\vee x_1x_2x_3x_4x_5x_6\vee x_2x_3x_4x_5x_1'x_6'\vee x_1x_2x_3x_4x_5'x_6'$$

$$F_2 = x_1x_2x_6x_3'x_5'\vee x_1x_2x_3x_4x_6'\vee x_1x_2x_4x_5x_6'\vee x_3x_4x_1'x_2'x_5'\vee x_1x_5x_2'x_4'x_6'\vee x_1'x_5'\vee x_6x_1'x_2'x_3'x_4'\vee x_1x_3x_4x_5x_6\vee x_2x_4x_1'x_3'\vee x_1x_5x_2'x_4'x_6'\vee x_1x_4x_2'x_3'x_5'\vee x_1x_2'x_3'x_5'x_6'\vee x_4x_5x_1'x_2'x_6'\vee x_3x_4'x_1'x_2'x_5'x_6'\vee x_2x_3x_6x_1'x_4'\vee x_4x_5x_6x_2'x_3'\vee x_1x_2x_3x_4x_5x_6'\vee x_2x_3x_4x_5x_1'x_6'\vee x_1x_2x_3x_4x_5'x_6'$$

$$F_3 = x_3x_4x_5x_6x_2'\vee x_1x_2x_5x_6x_3'\vee x_1x_2x_4x_5'\vee x_1'x_2'x_4'x_5'x_6'\vee x_2x_4x_1'x_5'\vee x_3x_4x_1'x_5'\vee x_3x_1'x_5'x_6'\vee x_3x_1'x_2'x_5'\vee x_1x_2x_3x_5x_6'\vee x_4x_5x_1'x_3'x_6'\vee x_1x_5x_2'x_4'x_6'\vee x_1x_2x_3'x_5'x_6'\vee x_1x_3x_6x_4'x_5'\vee x_3x_1'x_2'x_4'x_6'\vee x_5x_6x_2'x_3'x_4'\vee x_1'x_2'x_4'x_5'x_6'\vee x_2x_6x_1'x_3'x_4'x_5'\vee x_2x_3x_5x_6x_1'x_4'$$

$$F_4 = x_1x_2x_5x_6x_3'\vee x_6x_1'x_2'x_4'x_5'\vee x_2x_4'x_5'x_6'\vee x_6x_2'x_3'x_4'x_5'\vee x_1x_6x_2'x_3'x_5'\vee x_1x_2x_3x_6x_4'\vee x_1x_3'x_4'x_5'x_6'\vee x_2x_5x_6x_1'x_3'\vee x_2x_3'x_4'x_6'\vee x_1x_3x_5x_2'x_4'\vee x_1x_2x_3x_4x_5x_6'\vee x_3x_5x_1'x_2'x_6'\vee x_1x_3x_5x_6x_2'\vee x_2x_5x_1'x_3'x_6'\vee x_2x_4x_6x_1'x_5'\vee x_5x_6x_1'x_2'x_3'\vee x_1x_4x_5x_2'x_3'x_6'$$

$S_4 - box$

$$F_1 = x_1x_3x_4x_5x_6\vee x_2x_3x_4x_6x_1'\vee x_1x_2x_6x_4'x_5'\vee x_1x_3x_2'x_4'x_6'\vee x_1x_2'x_3'x_5'x_6'\vee x_2x_4x_5x_6\vee x_1x_3x_5x_2'x_6'\vee x_1x_2x_3'x_4'x_5'\vee x_1x_3x_6x_2'x_5'\vee x_3x_4x_1'x_2'x_6'\vee x_4x_1'x_3'x_5'x_6'\vee x_5x_1'x_2'x_3'x_4'\vee x_1x_2x_4x_5x_3'\vee x_5x_6x_2'x_3'x_4'\vee x_2x_3x_1'x_4'x_6'\vee x_5x_6x_1'x_2'x_4'\vee x_6x_1'x_2'x_3'x_5'\vee x_1x_2x_3x_4x_5'x_6'$$

$$F_2 = x_1x_3x_4'x_5'x_6'\vee x_1x_2x_4x_6x_3'x_5'\vee x_2x_5x_6x_1'x_3'\vee x_1'x_2'x_3'x_5'x_6'\vee x_3x_5x_1'x_4'x_6'\vee x_1x_5x_6x_2'x_3'\vee x_1x_5x_6x_3'x_4'\vee x_6x_1'x_2'x_4'x_5'\vee x_2x_3x_4x_1'x_5'\vee x_1x_3x_4x_2'x_6'\vee x_4x_5x_6x_2'x_3'\vee x_1x_2x_3x_5x_6\vee x_1x_2x_3x_4'x_5'x_6'\vee x_2x_5x_1'x_4'x_6'\vee x_3x_6x_1'x_2'x_4'\vee x_1x_3x_4x_2'x_5'\vee x_5x_2'x_3'x_4'x_6'$$

$$F_3 = x_1x_2x_4x_5x_3'\vee x_2x_5x_6x_1'x_4'\vee x_2x_4x_6x_1'x_5'\vee x_1x_2'x_3'x_4'\vee x_1x_5x_6x_2'x_3'\vee x_3x_5x_1'x_2'\vee x_1x_2x_3x_5x_6\vee x_1x_2x_3x_4x_6\vee x_2x_5x_1'x_3'x_4'\vee x_1x_3x_5x_4'x_6'\vee x_4x_5x_1'x_2'x_6'\vee x_3x_6x_2'x_4'x_5'\vee x_1x_2x_3'x_5'x_6'\vee x_3x_4x_5x_1'x_6'\vee x_4x_6x_1'x_3'x_5'\vee x_1'x_2'x_3'x_5'x_6'\vee x_2x_3x_1'x_4'x_5'x_6'\vee x_1x_3x_4x_2'x_5'x_6'$$

$$F_4 = x_1x_2x_4x_6x_3'\vee x_2x_3x_1'x_4'x_5'\vee x_3x_4x_5x_6x_1'\vee x_1x_2x_3'x_5'\vee x_4x_5x_6x_1'x_2'\vee x_2x_4'x_5'x_6'\vee x_4x_5x_1'x_2'x_3'\vee x_5x_1'x_2'x_4'x_6'\vee x_1x_3x_4x_2'x_5'\vee x_1'x_2'x_3'x_4'x_6'\vee x_1x_3x_5x_2'x_6'\vee x_2x_4x_5x_1'x_6'\vee x_1x_4x_3'x_5'x_6'\vee x_3x_4x_2'x_5'x_6'\vee x_1x_6x_2'x_3'x_4'\vee x_1x_3x_5x_6x_4'\vee x_1x_2x_3'x_4'x_6'\vee x_3x_5x_6x_2'x_4'\vee x_6x_1'x_2'x_3'x_5'\vee x_2x_5x_6x_1'x_3'x_4'$$

$S_5 - box$

$$F_1 = x_1x_4x_5x_2'x_6'\vee x_1x_3x_4x_5x_6'\vee x_1x_3x_5x_2'\vee x_1x_6x_2'x_3'x_5'\vee x_2x_3x_1'x_5'x_6'\vee x_1x_2x_5x_6x_3'\vee x_1x_2x_3x_6x_4'x_5'\vee x_3x_4x_1'x_5'\vee x_4x_5x_6x_1'x_3'\vee x_1x_2x_3'x_5'x_6'\vee x_2x_4x_6x_1'x_5'\vee x_1x_3'x_4'x_5'x_6'\vee x_2x_4x_5x_1'x_6'\vee x_1x_2x_5x_3'x_4'\vee x_2x_1'x_4'x_5'x_6'\vee x_5x_1'x_2'x_4'x_6'\vee x_1x_2x_3x_6x_4'x_5'\vee x_2x_3x_5x_6x_1'x_4'$$

$$F_2 = x_2x_6x_3'x_4'x_5'\vee x_1x_3x_4x_2'x_5'x_6'\vee x_1x_2x_5x_6x_4'\vee x_1x_2x_3x_4x_6x_5'\vee x_6x_1'x_2'x_4'x_5'\vee x_2x_3x_4x_5x_6x_1'\vee x_1x_3x_5x_2'x_4'\vee x_1x_2x_6x_3'x_4'\vee x_5x_1'x_3'x_4'x_6'\vee x_2x_3x_4'x_5'x_6'\vee x_1x_4x_5x_6x_2'\vee x_4x_5x_6x_2'x_3'\vee x_5x_1'x_3'x_4'x_6'\vee x_3x_6x_1'x_2'x_4'\vee x_6x_1'x_3'x_4'x_5'\vee x_2x_4'x_5'x_6'\vee x_3x_1'x_4'x_5'x_6'\vee x_2x_6x_1'x_3'x_5'\vee x_2x_3'x_5'x_6'\vee x_1x_4x_6x_2'x_3'\vee x_1x_2x_3'x_5'x_6'\vee x_1x_2x_4x_5x_6'\vee x_1x_2x_4'x_5'x_6'\vee x_1x_3'x_4'x_5'x_6'\vee x_3x_4x_5x_1'x_2'x_6'\vee x_4x_1'x_2'x_3'x_5'x_6'$$

$F_3 =$ $x_2x_3x_6x_4'x_5'\vee x_6x_2'x_3'x_4'x_5'\vee x_1x_3x_4x_2'x_5'\vee$
$x_1x_5x_2'x_3'x_6'\vee x_3x_1'x_2'x_6'\vee x_2x_4x_1'x_3'\vee$
$x_2x_4x_1'x_5'x_6'\vee x_3x_2'x_5'x_6'\vee x_2x_4x_1'x_3'\vee$
$x_1x_2x_4'x_5'\vee x_2x_3x_4x_5x_6\vee x_1x_4x_5x_2'x_3'\vee$
$x_4x_6x_1'x_3'x_5'\vee x_1x_2x_6x_3'x_4'\vee x_5x_6x_1'x_2'x_4'\vee$
$x_1'x_2'x_4'x_5'x_6'\vee x_3x_5x_6x_2'x_4'\vee x_1x_2x_3x_5x_6'$

$F_4 =$ $x_1x_2x_3x_4x_6\vee x_1x_2x_3'x_4'x_6'\vee x_1x_2x_5x_3'\vee$
$x_5x_6x_1'x_2'x_4'\vee x_4x_5x_3'x_6'\vee x_2x_6x_1'x_4'x_5'\vee$
$x_2x_4x_5x_1'x_6'\vee x_3x_4x_1'x_2'x_5'\vee x_2x_3x_6x_1'x_4'\vee$
$x_5x_6x_1'x_2'x_4'\vee x_1x_4x_2'x_5'x_6'\vee x_1x_3x_5x_4'x_6'\vee$
$x_3x_4x_6x_1'x_2'\vee x_1x_6x_2'x_4'x_5'\vee x_2x_4x_1'x_3'x_5'\vee$
$x_3x_1'x_4'x_5'x_6'$

$S_6 - box$

$F_1 =$ $x_2'x_3'x_5'x_6'\vee x_3x_1'x_4'x_5'x_6'\vee x_1x_3x_4x_6\vee$
$x_6x_1'x_2'x_3'x_4'\vee x_2x_5x_1'x_3'x_4'x_6'\vee x_2x_3x_4x_5x_1'\vee$
$x_1x_3x_4x_5'\vee x_1x_3x_5x_4'x_6'\vee x_1x_4x_5x_6x_2'\vee$
$x_3x_4x_6x_2'x_5'\vee x_3x_5x_6x_1'x_4'\vee x_1x_5x_2'x_4'x_6'\vee$
$x_1x_2x_6x_3'x_4'\vee x_1x_3x_6x_2'x_5'\vee x_4x_5x_1'x_2'x_6'\vee$
$x_2x_4x_6x_1'x_3'\vee x_2x_5x_1'x_3'x_4'x_6'\vee x_1x_2x_4x_5x_3'x_6'$

$F_2 =$ $x_1x_2x_3'x_5'x_6'\vee x_1x_5x_2'x_3'x_6'\vee x_4x_{5x_1}'x_3'x_6'\vee$
$x_1x_4x_2'x_5'x_6'\vee x_4x_5x_2'x_3'x_6'\vee x_3x_5x_6x_2'x_4'\vee$
$x_3x_5x_6x_1'x_2'\vee x_1x_4x_5x_6x_3'\vee x_1x_3x_4x_2'x_5'\vee$
$x_2x_4x_6x_1'x_3'\vee x_3x_6x_1'x_2'x_4'\vee x_2x_5x_1'x_4'x_6'\vee$
$x_1x_2x_3'x_4x_5\vee x_3x_5x_6x_2'x_4'\vee x_1x_2x_5x_6x_3'\vee$
$x_2x_3x_1'x_5'x_6'\vee x_2x_3x_5x_4'x_6'\vee x_1x_2x_5x_6'x_3'\vee$
$x_1x_6x_2'x_3'x_4'x_5'\vee x_1'x_2'x_3'x_4'x_5'x_6'$

$F_3 =$ $x_2x_3x_4x_6x_1'x_5'\vee x_4x_1'x_3'x_5'x_6'\vee x_{1x_4}x_6x_2'x_5'\vee$
$x_1x_5x_6x_3'x_4'\vee x_4x_5x_6x_1'x_3'\vee x_2x_3x_1'x_4'x_6'\vee$
$x_2x_3x_5x_1'x_6'\vee x_1x_5x_2'x_3'x_4'\vee x_2x_3x_5x_{1'x_4}'\vee$
$x_1x_2x_6x_4'x_5'\vee x_1x_2x_4x_5x_6'\vee x_1x_2x_3'x_4'x_5'\vee$
$x_6x_1'x_3'x_4'x_5'\vee x_1x_2x_5x_6x_3'\vee x_5x_6x_1'x_2'x_3'\vee$
$x_1x_2x_3x_4x_6'\vee x_4x_5x_1'x_2'x_3'\vee x_1x_4x_2'x_3'x_5'\vee$
$x_1x_3x_4x_5x_2'\vee x_1x_3x_2'x_4'x_5'x_6'$

$F_4 =$ $x_1x_5x_6x_2'x_4'\vee x_2x_3x_5x_4'x_6'\vee x_5x_1'x_3'x_4'\vee$
$x_1x_2x_4'x_5'x_6'\vee x_3x_4x_6x_1'x_2'\vee x_3x_4x_6x_1'x_2'\vee$
$x_2x_5x_1'x_4'\vee x_3x_1'x_2'x_4'x_5'\vee x_3x_6x_2'x_5'\vee$
$x_5x_1'x_2'x_3'x_6'\vee x_1x_2x_4x_5x_6\vee x_1x_2x_6x_3'x_5'\vee$
$x_1x_4x_5x_2'x_6'\vee x_2x_3x_5x_1'x_6'\vee x_1x_2x_3x_5'x_6'\vee$
$x_1x_2'x_3'x_5'x_6'$

$S_7 - box$

$F_1 =$ $x_1x_4x_5x_3'\vee x_1x_2x_3x_4x_5'x_6'\vee x_1x_4x_{2'x_3}'\vee$
$x_2x_4x_5x_6x_3'\vee x_1x_2x_3'x_4'x_6'\vee x_6x_1'x_2'x_3'x_5'\vee$
$x_3x_4x_1'x_2'x_6'\vee x_1x_2x_6x_4'x_5'\vee x_4x_5x_2'x_6'\vee$
$x_6x_1'x_3'x_4'x_5'\vee x_1x_2x_4x_5x_6\vee x_1x_4x_6x_2'x_5'\vee$
$x_1x_5x_6x_2'x_3'\vee x_2x_3x_5x_1'x_4'\vee x_3x_5x_6x_1'x_2'\vee$
$x_3x_2'x_4'x_5'x_6'\vee x_2x_3x_4x_6x_1'x_5'\vee x_2x_4x_1'x_3'x_5'x_6'$

$F_2 =$ $x_1x_2x_5x_6x_3'\vee x_1x_2x_3x_6x_4'x_5'\vee x_1x_2x_5x_4'x_6'\vee$
$x_2x_4x_5x_6\vee x_6x_1'x_3'x_4'x_5'\vee x_4x_5x_1'x_3'\vee$
$x_1'x_2'x_4'x_5'\vee x_1x_3x_2'x_5'x_6'\vee x_1x_3x_5x_6x_2'\vee$
$x_1x_5x_3'x_4'x_6'\vee x_1x_6x_2'x_3'x_5'\vee x_2x_3x_1'x_5'x_6'\vee$
$x_2x_3x_5x_6x_1'\vee x_2x_6x_1'x_3'x_5'\vee x_2x_5x_1'x_3'x_6'\vee$
$x_1x_2x_4x_3'x_5'x_6'$

$F_3 =$ $x_1x_2x_3'x_4'x_6'\vee x_1x_2x_4x_5x_6x_3'\vee x_5x_1'x_2'x_3'x_6'\vee$
$x_1x_3x_4x_2'\vee x_3x_4x_5x_6x_2'\vee x_4x_1'x_2'x_3'\vee$
$x_1x_4x_3'x_5'x_6'\vee x_2x_3x_6x_4'\vee x_2x_6x_1'x_4'\vee$
$x_4x_1'x_2'x_3'\vee x_1x_3x_5x_2'x_6'\vee x_2x_3x_5x_1'x_4'\vee$
$x_2x_1'x_3'x_4'x_5'\vee x_3x_4x_5x_6x_1'\vee x_1x_2x_3x_6x_5'\vee$
$x_1x_3x_4x_5x_6'\vee x_1x_6x_2'x_3'x_4'\vee x_4x_5x_1'x_3'x_6'\vee$
$x_3x_1'x_2'x_4'x_5'x_6'\vee x_2x_3x_4x_1'x_5'x_6'$

$F_4 =$ $x_1x_2x_5x_6x_3'\vee x_1x_2x_3x_4x_5'\vee x_2x_1'x_4'x_5'x_6'\vee$
$x_2x_4x_5x_1'x_6'\vee x_4x_6x_1'x_2'x_5'\vee x_2x_5x_6x_3'x_4'\vee$
$x_4x_6x_1'x_2'x_3'\vee x_2x_4x_1'x_3'x_5'\vee x_1x_2x_5x_4'x_6'\vee$
$x_3x_1'x_4'x_5'x_6'\vee x_1x_5x_6x_3'x_4'\vee x_1x_4x_2'x_3'x_6'\vee$
$x_3x_4x_5x_1'x_6'\vee x_1x_2'x_3'x_5'x_6'\vee x_3x_5x_6x_1'x_4'\vee$
$x_1x_3x_4x_5'x_6'\vee x_1x_2x_6x_3'x_4'\vee x_1x_3x_5x_4'x_6'\vee$
$x_1x_4x_2'x_3'x_5'\vee x_1x_3x_6x_2'x_4'x_5'\vee x_1x_3x_4x_5x_6x_2'\vee$
$x_5x_1'x_2'x_3'x_4'x_6'$

$S_8 - box$

$F_1 =$ $x_1'x_3'x_4'x_5'x_6'\vee x_1x_3x_5x_6x_2'\vee x_1x_3x_2'x_5'x_6'\vee$
$x_1x_2x_4x_3'x_6'\vee x_1x_4x_6x_2'x_5'\vee x_4x_1'x_2'x_5'x_6'\vee$
$x_4x_6x_1'x_2'x_3'\vee x_1x_3x_4'x_5'x_6'\vee x_2x_3x_4x_1'x_5'\vee$
$x_1x_2x_3x_4x_5\vee x_2x_4x_5x_1'x_3'\vee x_2x_1'x_3'x_4'x_6'\vee$
$x_2x_1'x_3'x_4'x_5'\vee x_3x_5x_2'x_4'x_6'\vee x_5x_6x_1'x_2'x_3'\vee$
$x_1x_2x_4x_3'x_5'\vee x_1x_2x_6x_3'x_4'\vee x_1x_5x_2'x_4'x_6'\vee$
$x_2x_3x_5x_6x_1'x_4'\vee x_3x_6x_1'x_2'x_4'x_5'$

$F_2 =$ $x_1'x_2'x_4'x_5'x_6'\vee x_4x_6x_1'x_3'x_5'\vee x_4x_5x_1'x_3'x_6'\vee$
$x_1x_3x_6x_2'x_4'x_5'\vee x_3x_4x_6x_1'x_2'\vee x_2x_4x_5x_1'x_6'\vee$
$x_1x_2x_5x_3'x_4'\vee x_1x_2x_3x_4x_5'\vee x_1x_3x_4x_5'x_6'\vee$
$x_3x_5x_2'x_4'x_6'\vee x_5x_6x_1'x_3'x_4'\vee x_2x_5x_6x_4'\vee$
$x_1x_4x_5x_6x_2'\vee x_2x_6x_3'x_4'\vee x_2x_3x_5'x_6'\vee$
$x_2x_4x_5x_3'x_6'\vee x_1x_4x_5x_6x_2'\vee x_1x_2'x_3'x_5'x_6'\vee$
$x_1x_4x_2'x_3'x_5'\vee x_1x_2'x_3'x_5'x_6'$

$F_3 =$ $x_3x_1'x_2'x_4'\vee x_1x_4x_6x_2'x_3'\vee x_3x_4x_2'x_5'x_6'\vee$
$x_1x_2x_5x_4'x_6'\vee x_2x_4x_5x_1'\vee x_1x_5x_3'x_4'x_6'\vee$
$x_2x_1'x_3'x_5'x_6'\vee x_3x_1'x_2'x_5'\vee x_1x_2x_6x_4'x_5'\vee$
$x_1x_2'x_3'x_4'x_5'\vee x_3x_5x_6x_2'x_4'\vee x_2x_4x_3'x_5'x_6'\vee$
$x_1x_2x_3x_4'x_6'\vee x_1x_2x_3x_4x_6\vee x_1x_3x_4x_2'x_6'\vee$
$x_2x_3x_5x_6x_1'$

$F_4 =$ $x_2x_5x_6x_1'x_3'\vee x_4x_6x_1'x_2'x_5'\vee x_1x_2'x_4'x_5'x_6'\vee$
$x_1x_5x_2'x_3'\vee x_5x_6x_2'x_3'x_4'\vee x_2'x_3'x_4'x_5'x_6'\vee$
$x_3x_4x_5x_1'x_6'\vee x_1x_2x_3x_4'\vee x_5x_6x_2'x_3'x_4'\vee$
$x_3x_4x_6x_1'x_5'\vee x_1x_4x_5x_3'x_6'\vee x_3x_4x_1'x_2'x_5'\vee$
$x_1x_2x_3x_5'x_6'\vee x_1x_3x_4x_5x_6\vee x_1x_2x_6x_3'x_5'\vee$
$x_2x_3x_4'x_5'x_6'\vee x_2x_5x_1'x_3'x_4'\vee x_3x_5x_1'x_2'x_4'\vee$
$x_1'x_2'x_3'x_4'x_5'\vee x_2x_4x_1'x_3'x_5'x_6'$