Comm. Korean Math. Soc. 7 (1992), No. 2, pp. 347-364

갈로아체에서의 고속연산법개발 및 응용

조인호, 임종인, 서광석, 김창한

1. 서론

갈로아 체(Galois field) $GF(p^m)$은 수학적 중요성외에 디지털 신호처리, 스윗칭회로, 오류정정부호, 각종 암호 시스템 및 이산로그 문제 등에 중요하게 응용되고 있는 분야이다. [3, 6] 갈로아체 중 특히 하드웨어 구현상의 편리함때문에 $GF(2^m)$이 주목을 받고 있다. 응용에 있어서의 효율성은 갈로아 체에서의 연산, 특히 곱셈, 역원계산, 멱승(exponentiation)의 연산속도에 비례하므로 연산속도를 빠르게 하기위한 연구가 활발히 진행되어 왔다. [1, 11] 1981년 Massey 와 Omura는 $GF(2^m)$ 에서의 정규기저(normal basis)를 이용하여 연산속도를 획기적으로 개선할 수 있는 방법을 개발하였다. [7] 이 방법은 큰 반응을 일으켰으며 1985년 C.C.Wang 등이 8 bit Massey-Omura 연산자 VLSI chip 을 설계하였다. [10] Massey-Omura 의 방법은 사용하는 정규기저에 따라 복잡도가 다르기 때문에 적은 복잡도를 가지는 정규기저를 찾으려는 노력이 따르게 되었다. Wang[10]은 self-dual 정규기저의 경우 복잡도가 줄어든다는 것을 보이고 이를 찾는 확률적 방법을 제시하였다. 본 논문은 이 방법을 완성하여 컴퓨터 구현하는 것을 목적으로 하고 있다.

한편 최근 Vanstone등이 제시한 최적 정규기저(optimal normal basis) 개념과[2, 8] self-dual 정규기저와의 관계를 살펴보는 것이 본 논문의 후속 과제라고 본다. 현재까지의 컴퓨터 실행결과는 두 개념사이에 밀접한 관계가 있음을 보여주고 있다. 2절에서는 $GF(2^m)$ 에서의 일반 성질들과 Massey-Omura 고속연산법을 제시하려 한다. 3절에서는 Wang 이 발견한[10] 정규기저와 self-dual 정규기저의

Received January 3, 1992.
이 논문은 1991년도 문교부지원 한국학술진흥재단의 지방대 육성 학술연구조성비에 의하여 연구되었음.

성질들을 제시하고 self-dual 정규기저를 찾는 방법을 제시하였다. 이 알고리즘의 프로그램 및 최적 정규기저 여부를 확인하는 프로그램을 부록으로 제시하고 있다. 본 논문의 결과는 최종 완성시 특히, 1988년 본 연구팀이 컴퓨터로 구현한바 있는 [13] 이산대수 알고리즘인 Coppersmith 알고리즘의 속도개선에 이바지 함으로서 암호통신 분야에의 역할이 클 것으로 생각된다.

2. 갈로아체의 일반 성질들과 Massey-Omura 고속연산법

갈로아체 중 $GF(2^m)$ 은 하드웨어 구현시의 편리함 때문에 공학적으로 특히 중요한 것이다. $GF(2^m)$ 을 만들기 위해서는 $GF(2)$ 상의 m 차 기약다항식 $p(x)$ 를 택하여[14] $GF(2)$ 의 확대체 $GF(2^m)[x]/(p(x))$ 를 만들면 된다는 것은 잘 알려져 있는 사실이다.

$\alpha \in GF(2^m)$ 를 $p(x)$의 한 근이라고 하면 $GF(2^m)$ 의 모든 원소는 $\{1, \alpha, \cdots, \alpha^{m-1}\}$ 의 일차결합으로 표시된다. 즉 $\{1, \alpha, \cdots, \alpha^{m-1}\}$ 은 $GF(2^m)$ 을 $GF(2)$ 상의 m 차 벡터공간으로 보았을때 기저(basis)가 보통 관용기저(conventional basis)라 부른다. 관용기저를 사용하면 덧셈 연산시는 편리하지만 곱셈등의 연산시는 시간이 많이 걸리는 단점이 있다. $\beta \in GF(2^m)$ 에 대해 $Tr(\beta) = \beta + \beta^2 + \cdots + \beta^{2^{m-1}}$ 와 같이 정의 하면 우리는 다음의 두 개념을 도입할 수 있다.

정의 2.1. $GF(2^m)$ 의 두 기저 $\{\beta_1, \cdots, \beta_m\}, \{\gamma_1, \cdots, \gamma_m\}$ 이 dual 하다는 것은 $Tr(\beta_i, \gamma_j) = \delta_{ij}$ 가 성립하는 것이다.

정의 2.2. $\beta \in GF(2^m)$ 에 대해 $\{\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{m-1}}\}$ 이 기저가 될 때 이것을 정규기저 (normal basis)라 한다.

갈로아체는 항상 정규기저를 가지고 있으며 정규기저의 dual 기저는 정규기저라는 사실은 잘 알려져 있다. [6]

정의 2.3. 정규기저 $\{\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{m-1}}\}$ 가 self-dual 정규기저라는 것은 이것의 dual 기저가 자기자신인 것이다. 즉, $Tr(\beta^{2^i} \cdot \beta^{2^j}) = \delta_{ij}$ 가 성립하는 것이다.

우리는 m이 4의 배수가 아니면 $GF(2^m)$ 은 항상 self-dual 정규기저를 가진다는 것을 알고있다. [8]

이제 Massey-Omura의 정규기저를 이용한 고속연산법을 제시하려한다. [7] 정규기저 $\{\alpha, \alpha^2, \alpha^{2^2}, \cdots, \alpha^{2^{m-1}}\}$ 이 주어졌다고 하자. 그러면 모든 $\beta \in GF(2^m)$ 은 유일하게

$$\beta = b_0\alpha + b_1\alpha^2 + b_2\alpha^{2^2} + \cdots + b_{m-1}\alpha^{2^{m-1}}, b_i \in GF(2)$$

와 같이 표현 되므로

$$\begin{aligned}\beta^2 &= b_0\alpha^2 + b_1\alpha^{2^2} + \cdots + b_{m-2}\alpha^{2^{m-1}} + b_{m-1}\alpha^{2^m} \\ &= b_{m-1}\alpha + b_0\alpha^2 + \cdots + b_{m-2}\alpha^{2^{m-1}}\end{aligned}$$

이 성립한다. 즉 $\beta = (b_0, b_1, b_2, \cdots, b_{m-1})$ 과 같이 벡터표현을 도입하면 $\beta^2 = (b_{m-1}, b_0, \cdots, b_{m-2})$ 가 되어 β^2 은 β의 cyclic shifting 에 의해 구해진다. 두 원소 $\beta = (b_0, b_1, b_2, \cdots, b_{m-1})$, $\gamma = (c_0, c_1, c_2, \cdots, c_{m-1})$ 가 주어졌을때 $\delta = (d_0, d_1, d_2, \cdots, d_{m-1})$ 의 마지막 항인 d_{m-1} 은 적당한 이항함수 (binary function) f에 대해

$$d_{m-1} = f(b_0, b_1, b_2, \cdots, b_{m-1}; c_0, c_1, c_2, \cdots, c_{m-1})$$

가 되는 것을 알 수 있다.

$$\begin{aligned}\delta^2 &= \beta^2 \cdot \gamma^2 = (b_{m-1}, b_0, \cdots, b_{m-2}) \cdot (c_{m-1}, c_0, \cdots, c_{m-2}) \\ &= (d_{m-1}, d_0, \cdots, d_{m-2}) \text{ 이므로} \\ d_{m-2} &= f(b_{m-1}, b_0, \cdots, b_{m-2}; c_{m-1}, c_0, \cdots, c_{m-2})\end{aligned}$$

임을 알 수 있다. 같은 방법으로 우리는 함수 f 에 대해 β와 γ를 cyclic shifting 하여 대입함으로서 $\delta = (d_0, d_1, d_2, \cdots, d_{m-1})$를 구할 수 있다는 것을 알 수 있다.

이것이 Massey-Omura 가 발견한 고속 곱셈법(Massey-Omura multiplier)으로서 현재 미국에 특허화되어 있다. [7] 위 방법을 쓰면 정규기저가 주어지면 이에 종속되는 곱셈함수(product function) f 를 찾을 수 있게 되고, 이로부터 곱셈은 자동적으로 이루어질 수 있게 된다. 이 방법은 하드웨어 구현시 관용기저 사용시보다 훨씬 간편하다는 것이 밝혀졌으나, 곱셈함수가 n 개의 항을 가지고 있으면 chip 설계시 $n-1$ 개의 EX-OR gate 와 n 개의 AND-gate 가 요구된다. 따라서 효율적인 Massey-Omura 고속 곱셈법을 구현 하기 위해서는 첫째, 정규기저를 쉽게 발견할 수 있어야하고 둘째, 되도록 적은 항을 갖는 곱셈함수를 가져오는 정규기저를 발견하여야 한다. 이 문제는 3절에서 다루겠다.

예 2.4. $p(x)=x^4+x^3+1$ 을 사용하면 $GF(2^4)$ 을 만들 수 있다. 이의 한 근을 α 라 하면 $\{\alpha,\alpha^2,\alpha^4,\alpha^8\}$ 은 정규기저가 되고 이때 $\beta=(b_0,b_1,b_2,b_3)$, $\gamma=(c_0,c_1,c_2,c_3)$, $\delta=\beta\gamma=(d_0,d_1,d_2,d_3)$ 라 하면 $\alpha^4=\alpha^3+1$ 이므로

$$\begin{aligned} d_3 &= f(b_0,b_1,b_2,b_3;c_0,c_1,c_2,c_3) \\ &= b_2c_2+b_3c_2+b_2c_3+b_3c_1+b_1c_3+b_3c_0+b_0c_3+b_1c_0+b_0c_1 \end{aligned}$$

임을 알 수 있다. 즉 f 는 9개의 항으로 이루어져 있다. 또다른 기약다항식 x^4+x+1 로 시작하면 발생하는 곱셈함수는 7개의 항을 갖게 된다.

곱셈함수를 나타내는 효율적인 방법은 $m\times m$ Boolean 행렬 $\Omega=(\rho_{i,j})_{ij=0}^{m-1}$ 을 이용하는 것이다. 즉 곱셈함수에서 b_ic_j 항이 있으면 $\rho_{ij}=1$ 이고, 없으면 $\rho_{ij}=0$ 으로 한다. 위의 예 2.4의 경우

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

이 된다.

이제 $\beta\in GF(2^m)$ 의 역원 β^{-1} 를 구하는 문제를 살펴보자. 1988년 Itoh 가 발견한 더욱 효율적인 방법이 있지만 [4] 여기에서는 Wang 의 방법을 제시하였다 [10]. $\beta^{2^m-1}=\beta\cdot\beta^{2^m-2}=1$ 이므로 $\beta^{-1}=\beta^{2^m-2}$ 이다.

$2^m-2=2+2^2+\cdots+2^{m-1}$ 이므로 $\beta^{-1}=\beta^2\cdot\beta^{2^2}\cdots\beta^{2^{m-1}}$ 이다. 따라서 β^{-1} 는 $m-1$ 번의 제곱(squaring)과 $m-2$ 번의 곱셈으로 구할 수 있다. 정규기저 사용시 이것들은 앞에서 보았듯이 효율적으로 수행될 수 있으므로 역원을 어렵지 않게 구할 수 있다.

갈로아 체에서 멱승(exponentiation)은 ElGamal 공개키 암호법[3]에서 효율성을 결정짓는 가장 중요한 연산이다. β^k 을 구하기 위해 k 를 이항전개 (binary expansion)하면 역원에서와 같이 멱승문제는 제곱과 곱셈의 문제로 귀착되므로 쉽게 구할 수 있다.

3. Self-dual 정규기저와 location 알고리즘

2절에서는 정규기저를 이용하여 곱셈함수 또는 boolean 행렬을 만들어서 $GF(2^m)$ 에서의 고속연산을 수행할 수 있다는 것을 알았다. 정규기저는 항상 존재하고 Berlekamp, Lidl-Niederreiter [5]는 정규기저의 갯수를 계산하는 공식까지 만들어 내었지만 실제로 정규기저를 찾는 문제(location of normal basis)와는 별개의 문제였다. Wang은 m 이 솟수이고 2를 원시근(primitive root mod m)으로 가질 때에는 m 차 기약 다항식 $P(x)$의 $m-1$ 차 계수가 1일 경우 $P(x)$의 근인 $\{\alpha,\alpha^2,\cdots,\alpha^{2^{m-1}}\}$ 이 정규기저가 되고 계수가 0일 경우 $\beta=1+\alpha$ 라 할 때 $\{\beta,\beta^2,\cdots,\beta^{2^{m-1}}\}$ 가 정규기저가 된다는 것을 발견하였다 . [10] 실제로 1000이하의 솟수 167개중 57개가 2를 원시근으로 가지고 있기에 이것은 정규기저를 찾는 매우 중요한 사실이다. 정규기저 N 에 대응되는 Boolean 행렬의 non-zero 성분의 갯수를 이 정규기저의 복잡도 C_N 이라 하며, 이것이 $C_N\geq 2m-1$ 을 만족한다는 것이 알려져 있다. $C_N=2m-1$ 일 때 이 정규기저는 최적정규기저(optimal normal basis)라 하며 Wang 은 N 이 self-dual 인 경우 최적이 되는가 하는 문제를 제기 하였다. [8] 본 논문의 목적은 이러한 질문에 대한 답으로서 self-dual 정규기저를 찾아 이들의 복잡도를 알아보고 최적 여부를 알아보는데 있다.

이 경우 생성 기약 다항식 $p(x)$를 변화시키기 보다는 이것을 고정시킨후 dual 기저를 이용하여 원소를 변화시키면서 self-dual 정규기저를 찾아내는 Wang 의 방법이 우리의 구현 결과 매우 효율적인 것으로 판명되었다.

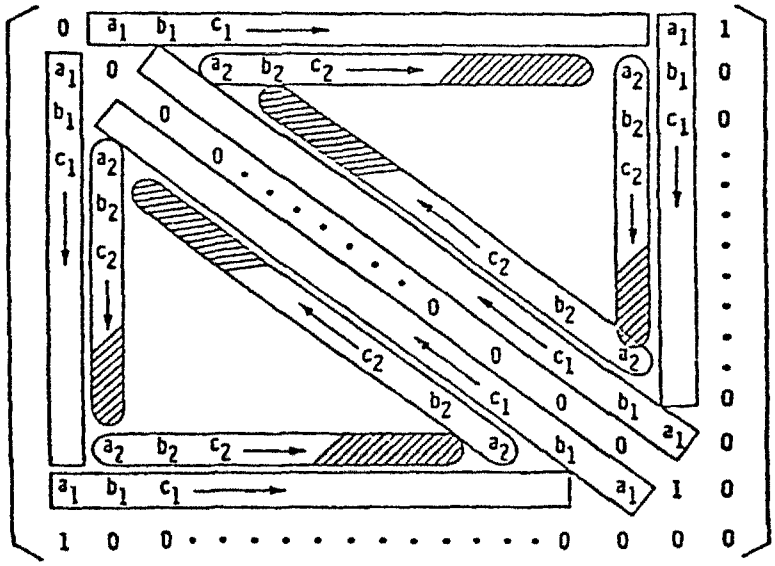
$\{\alpha,\alpha^2,\cdots,\alpha^{2^{m-1}}\}$ 이 $GF(2^m)$ 의 한 정규기저라 하고 $\{\beta,\beta^2,\cdots,\beta^{2^{m-1}}\}$

을 dual 기저라 할 때 대응되는 Boolean 행렬 $\Omega = (\rho_{ij})_{i,j=0}^{m-1}$ 의 (i,j) 성분 ρ_{ij} 는 $\rho_{ij} = Tr(\alpha^{2^i} \cdot \alpha^{2^j} \cdot \beta^{2^{m-1}})$ 이 된다는 것이 알려져 있다[10]. 이로부터 Ω 가 대칭 행렬이고 대각성분은 $m-2$ 번째를 제외하고는 0이고 마지막 $m-1$ 번째의 행 또는 열을 제외하고는 짝수개의 1을 갖는다는 것은 자명하다. 이제 $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$ 이 self-dual 정규기저라 하면 대응되는 Boolean 행렬 $\Omega = (\rho_{ij})_{i,j=0}^{m-1}$ 에서 $\rho_{ij} = \mathrm{Tr}(\alpha^{2^i}, \alpha^{2^j}, \cdots, \alpha^{2^{m-1}})$ 이 된다. 이 경우 정규기저에서는 성립하지 않았던 다음의 성질이 성립한다[10].

정리 3.1. $\rho_{i,m-1} = \rho_{m-1,i} = \delta_{i,0}$,

$\rho_{ij} = \rho_{(m-1+i-j)(m-j-2)} = \rho_{(j-i-1)(m-i-2)}$.

위의 정리 3.1은 self-dual 정규기저를 사용하였을 때의 Boolean 행렬이 갖는 특징을 보여주고 있다. 첫번째 성질은 이 행렬의 마지막 행과 열이 첫번째 성분을 제외하고는 0이라는 것을 나타내고 있으며 두번째 성질은 다음의 그림 3.1에서 보는 바와 같은 삼각형 대칭구조를 나타내고 있다.



$a_i, b_i, C_i \in GF(2)$

그림 3.1. The Structure of a Boolean Matrix Corresponding to a Self-Dual Normal Basis

즉, self-dual 정규기저를 사용할 경우 일반 정규기저를 사용할 경우보다 약 1/3의 trace 계산을 통하여 Boolean 행렬을 만들 수 있다는 것을 알 수 있다.

이제 self-dual 정규기저를 찾는 방법을 제시하겠다. m은 이제부터 홀수 이다. 이 방법은 Wang 이 제시한 방법을 완성한 것으로 (정리 3.2) 우리는 알고리즘 완성 및 Fortran언어로 프로그램화에 성공하였다. 이것은 부록에서 제시하겠다. 여기에 사용되는 정규기저를 찾는 방법은 [9, 10]에 제시되어있다. $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$을 정규기저라 하고 $\{\beta, \beta^2, \cdots, \beta^{2^{m-1}}\}$을 self-dual 정규기저라 하자. 그러면

(1)
$$\begin{pmatrix} \alpha \\ \alpha^2 \\ \vdots \\ \alpha^{2^{m-1}} \end{pmatrix} = \begin{pmatrix} b_0 & b_1 & \cdots & b_{m-1} \\ b_{m-1} & b_0 & \cdots & b_{m-2} \\ \vdots & \vdots & & \vdots \\ b_1 & b_2 & \cdots & b_0 \end{pmatrix} \begin{pmatrix} \beta \\ \beta^2 \\ \vdots \\ \beta^{2^{m-1}} \end{pmatrix} = \mathbf{B} \begin{pmatrix} \beta \\ \beta^2 \\ \vdots \\ \beta^{2^{m-1}} \end{pmatrix}_{b_i \in GF(2)}$$

이 성립한다. (1)의 transpose 를 취하고 이것을 (1)에 다시 곱하면 우리는

(2)
$$\begin{pmatrix} \alpha \\ \alpha^2 \\ \vdots \\ \alpha^{2^{m-1}} \end{pmatrix} \cdot [\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}] = \mathbf{B} \begin{pmatrix} \beta \\ \beta^2 \\ \vdots \\ \beta^{2^{m-1}} \end{pmatrix} \cdot [\beta, \beta^2, \cdots, \beta^{2^{m-1}}] \cdot {}^tB$$

(2) 의 양변에서 trace를 취하면 $Tr(\beta^{2^i} \cdot \beta^{2^j}) = \delta_{ij}$ 이므로

(3) $$F(\alpha) = [F_{ij}]_{i,j=0}^{m-1} = [Tr(\alpha^{2^i+2^j})]_{i,j=0}^{m-1} = B \cdot {}^tB$$

가 성립한다. 정규기저 $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$으로 부터 self-dual 정규기저 $\{\beta, \beta^2, \cdots, \beta^{2^{m-1}}\}$를 구하기 위해서는 (3)으로 부터 B를 구해야 한다.

$$F_{ij} = F_{\overline{i-1}\,\overline{j-1}} (\overline{i} = i \mod m)$$

이므로 실제로는 (3)식에서 앞쪽행렬의 첫번째 행만 생각하면 된다. 이로부터

$$(4)\qquad \begin{cases} Tr(\alpha^2) = b_0^2 + b_1^2 + \cdots + b_{m-1}^2 \\ Tr(\alpha^3) = b_0 b_{m-1} + b_1 b_0 + \cdots + b_{m-1} b_{m-2} \\ Tr(\alpha^{2^j+1}) = \sum_{k=0}^{m-1} b_k b_{\overline{m-j+k}} \\ Tr(\alpha^{2^{m-1}+1}) = b_0 b_1 + b_1 b_2 + \cdots + b_{m-1} b_0 \end{cases}$$

가 성립한다. $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$가 정규기저이므로 $Tr(\alpha^2) = Tr(\alpha) = 1$이고 $b_i \in GF(2)$이므로 ${b_i}^2 = b_i$이다. 따라서 (4)식의 첫번째 식은 $\sum_{k=0}^{m-1} b_i = 1$이 된다. 즉 $\{b_0, b_1, \cdots, b^{m-1}\}$은 홀수개의 1을 가진다.

$Tr(\alpha^{2^j+1}) = Tr(\alpha^{2^{m-j}+1})(1 \le j < m/2)$ 이므로 (4)식의 첫번째를 제외한 나머지 식들은 처음 절반이 나중 절반의 식과 동일하다는 것을 알 수 있다. 따라서 (4)에는 단지 $(m-1)/2 + 1 = (m+1)/2$ 개의 방정식만 존재하게 된다. 미지수가 m개이므로 해는 유일하지 않다. 만일 모든 j에 대해 $F_{0j} = 0$이면 $b = (1, 0, \cdots, 0)$이 해가 된다. 이제 적어도 한개의 0이 아닌 F_{0j}가 존재할 때 우리는 다음의 해법을 유도하였다.

정리 3.2. F_{0j} 가운데에서 1의 숫자를 $L+1$ 이라 하자. $n_k (k = 0, 1, \cdots, L)$를 $F_{0i} = 1$ 이 되는 $L + 1 - k$ 번째의 큰 i 라하자. 자명하게 $0 = n < n_0 < n_1 < \cdots < n_L$이 된다. 그러면 다음의 벡터 $b = (b_0, b_1, \cdots, b_{m-1})$은 식 (4)의 해가 된다.

a) $b_0 = 1, b_{n_L} = 1$

b) n_L이 짝수이면

1) $b_{n_L/2} = 1$

2) $F_{0n_i} = 1$ 인 모든 짝수 n_i에 대해, $b(n_L + n_i)/2 = 1$

3) $F_{0n_i} = 1$ 인 모든 홀수 n_i에 대해, $b(m + n_L \pm n_i)/2 = 1$

c) n_L 이 홀수이면

1) $b(m+n_L)/2=1$

2) $F_{0n_i}=1$ 인 모든 짝수 n_i 에 대해, $b(m+n_L \pm n_i)/2=1$

3) $F_{0n}=1$ 인 모든 홀수 n_i 에 대해, $b(n_L \pm n_i)/2=1$

d) 나머지는 모두 *0*으로 한다.

증명: a) ~ d)에서와 같이하여 대입하여보면 식 (4)가 성립하는 것은 자명하다.

위와 같은 과정을 거쳐 행렬 B 를 구하였다고 하자. 기본 행변환을 이용하여 B 가 정칙행렬(regular matrix)인지를 확인하자. B 가 정칙이면

$$\begin{pmatrix} \beta \\ \beta^2 \\ \vdots \\ \beta^{2^{m-1}} \end{pmatrix} = \mathbf{B}^{-1} \begin{pmatrix} \alpha \\ \alpha^2 \\ \vdots \\ \alpha^{2^{m-1}} \end{pmatrix}$$

으로부터 self-dual 정규기저를 구할 수 있게 된다. B 가 정칙이 아니면 정규기저를 다시 택하여 위의 과정을 반복한다.

4. 실험결과 및 고찰

우리는 3절에서의 self-dual 정규기저를 찾는 알고리즘을 프로그램화 하였다. 언어는 Fortran 으로하였으며 우선 IBM 386-PC에 구현하였다. 차후 큰 용량의 컴퓨터에 구현할 예정으로 있다.

표 4.1의 table은 $m \leq 18$의 경우에 대한 우리의 구현결과를 보여주고 있다. 즉 복잡도가 가장 작은 경우를 나타내고 있다. 위의 table은 대부분의 경우에 self-dual정규기저를 사용하면 Boolean 행렬의 복잡도가 가장 작아진다는 것을 보여주고 있으며 $GF(2^3), GF(2^5), GF(2^6), GF(2^9), GF(2^{11}), GF(2^{14})$ 등의 경우 최적개념과 self-dual 개념이 일치한다는 것을 보여주고 있다.

$GF(2^7)$의 경우 $\alpha=\lambda^{13}$ 에 대응되는 Boolean 행렬은 19개의 가장 작은 0이 아닌 성분을 가졌으나 $\alpha=\lambda^{43}$ 으로 하면 self-dual 정규기저를 가져오지만 21개

의 0이 아닌 성분을 갖게 된다. $GF(2^{19})$에서의 self-dual 정규기저는 $n = 14$의 짝수이므로 정규기저를 먼저 찾은 후 check 과정에서 발견한 것이다. 이상의 결과에서 보았듯이 self-dual 정규기저는 분명히 대부분의 경우에 복잡도가 가장 작은 Boolean 행렬을 가져다 주고 최적이 된다.

finite field	generator	primitive polynomial	# of nonzero terms	self-dual
$GF(2^3)$	x^3	x^3+x+1	5	S
$GF(2^4)$	x^3	x^4+x+1	7	
$GF(2^5)$	x^5	x^5+x^2+1	9	S
$GF(2^6)$	x^{23}	x^6+x+1	11	S
$GF(2^7)$	x^{13}	x^7+x+1	19	
$GF(2^8)$	x^{47}	$x^8+x^4+x^3+x^2+1$	21	
$GF(2^9)$	x^{41}	x^9+x^4+1	17	S
$GF(2^{10})$	x^{93}	$x^{10}+x^3+1$	19	
$GF(2^{11})$	x^{439}	$x^{11}+x^2+1$	21	S
$GF(2^{12})$	x^{315}	$x^{12}+x^6+x^4+x+1$	23	
$GF(2^{13})$	x^{401}	$x^{13}+x^4+x^3+x+1$	45	S
$GF(2^{14})$	x^{3511}	$x^{14}+x^5+x^3+x+1$	27	S
$GF(2^{15})$	x^{1359}	$x^{15}+x+1$	45	S
$GF(2^{16})$	x^{1117}	$x^{16}+x^5+x^3+x^2+1$	85	
$GF(2^{17})$	x^{615}	$x^{17}+x^3+1$	81	S
$GF(2^{18})$	x^{5301}	$x^{18}+x^5+x^2+x+1$	93	

표 4.1. Table of generators of normal basis and # of nonzero terms in Boolean matrix.

이것은 Vanstone등이 최근 보였듯이 [2, 8] 그들이 만든 두가지 형태의 최적기저중 type-II 최적정규기저가 self-dual 이기 때문이라고 해석된다. 또한 이들 두가지 형태의 최적정규기저만이 존재한다는 그들의 conjecture 가 사실일 가능성이 크다는 암시를 하고 있다고 생각된다. 앞으로 큰 용량의 컴퓨터 구현을 통하여 이 두 개념의 관계를 알아보는 것이 필요하다고 생각 된다.

References

[1] G.B.Agnew R.C.Mullin I.M. Onyszchut and S.A.Vanstone, *Ab implementation for a Fast Public-Key Crypt-System*, J. Cryptlogy **3** (1991), 63–79.

[2] D. Ash I Blake and S. Vanstone, *Low Complexity Normal Bases*, Discrete Applied Mathematics **25** (1989), 191–210.

[3] G. Brassard, *Lecture Notes in Computer Science 325*, Springer-Verlag, 1988.

[4] T. Itoh and S.T.Sujii, *A fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Bases*, Information and Computation **78** (1989), 171–177.

[5] R. Lidle and H. Niederreiter, *Finite Fields*, Addison-Wesley Publishing Company, 1953.

[6] F.J. MacWilliams and N.J.A., *Sloane The Theory of Error-Correction Codes*, North-Holland Publishing, New York, 1977.

[7] J.L. Massey and J.K. Omura, *Computational-Method and Apparatus for Finite Field Arithmetic.*

[8] R.C. Mullin I.M. Onyszchuk S.A.Vanstone and R.M. Wilson, *Optimal normal bases in $GF(p^n)$*, Discrete Applied Mathematics **22** (1988-89), 149–161.

[9] P.K.S Wah and M.Z. Wang, *Realization and Application of the Massey-Omura Lock*, Preceedings of Internation Zurich Seminar March, 1984.

[10] C.C.Wang, *Exponentation in Finite Fields Ph. D dissertation*, School of Engineering and Applied Science U.C.L.A. June, 1985.

[11] 이창순, 문상재, *$GF(2^m)$* 의 정수기저를 사용한 *D-H*형 공용위의 분배시스템, KISC Proceeding (1991), 49–57.

[12] 박일환, *A Fast Multiplication and Inversion Method in $GF(2^m)$ Using Normal Bases*, 고려대 수학과 석사학위논문, 1990.

[13] 조인호, 임종인, 서광석, 유한체위의 이산대수와 암호계에서의 상용, WISC, 89 (1989), 277–287.

[14] 최윤서, *$GF(p)$* 의 유한확장체 구축을 위한 *$GF(p)$* 위에서의 기약다항식, 고려대 수학과 석사학위 논문, 1991.

This program is used for finding a generator of normal basis whose corresponding Boolean matrix contains the least number of I's and testing if it is a self-dul normal basis or not in GF(2m).

```
      INTEGER MAT(127, 254), PBT(127, 127)
      INTEGER XX(10001), MT(253), RDT(0)
      INTEGER CNT, DEG, TRACE, RHT,
     DXP, TRN1
      INTEGER TRN2, TERMS, RPT, END-
     CARD
      INTEGER NBSUM(100), DEG1, DEG2
      INTEGER MTA(127), CNTA, ODD
      DATA XX, MT, RDT/10001*0.253*0,10
     *0/
      DATA LEXP, STERMS/10000,10000/
      OPEN(6,FILE='LTM2.0UT')
C
      WRITE(*,1)
1     FORMAT(1X, 'Input Degree.')
      READ(*,2)DEG
2     FORMAT(13)
      WRITE(*,3)
3     FORMAT(1X,'Input Rdt. Poly.')
      READ(*,4)(RKT(1),I=1,10)
4     FORMAT(10I3)
      WRITE(*,5)
5     FORMAT(1X, 'Input Endcard.')
      READ(*, 6)ENDCARD
6     FORMAT(I9)
C
      CNT=1
      DO7I-2,10
      IF(FDT(I)EQ 0)GO TO 7
      CNT=CNT+1
      RDT(1)=RDT(1)+1
7     CONTINUE
C
      DO 999 EXP=1, ENDCARD, 2
C
      RHT=EXP+1
      IF(RHT, GT, 10001)RHT=MOD(EXP,100
     00)+1
      RPT=(EXT/10000)+1
C
      XX(RHT)=1
      DO 50 #=1, RPT
      IF(RHT, LE, DEG)CO TO 35
10    XX(RHT)=0
      DO20K=1, CNT
20    XX(RDT(K)+RHT-DEG-1)=
      MOD(SS(RDT(K)+RHT-DEG-1)+1,2)
30    RHT=RHT-1
      IF(XX(RHT),EQ,0)GO TO 30
      IF(FHT, GT, DEG)GO TO 10
35    IF(II EQ, RPT)GO TO 50
      DO 40 I IX1, RHT
      XX⑦10001-DEGV1⑧XXX(I)
40    XX(I)IX0
      RHTIX10001-(DEG-RHT)
50    CONTINUE
C
      DO 55IIX1, DEG
      DO 55 JIX1, DEG III2
55    MAT(I, J)IX0
      DO60JIX1,RHT
      MAT(1, j)IXXX(J)
60    SS(J)IX0
      DEG1IXRHT
C
      DO 100 iIX2, DEG
      DO 70 JIX1, RHT
      IF(MAT(I, 1, J)EQ,0)GO TO 70
      MAT(I,JIII2-1)IX1
70    CONTINUE
```

```
      RHT=RHT/2-1
      IF(RHT. LE. DEG)GO TO 100
   80 MAT(I. RHT)=0
      DO 85K=1. CNT
   85 MAT(I. RDT(K)/RHT-DEG-1)=
      MOD(MAT(I. RDT(K)/RHT-DEG-1)/2.2)
   90 RHT=RHT-1
      IF(MAT(I.RHT).EQ.0)GO TO 90
      IF(RHT. GT. DEG)GO TO 80
  100 CONTINUE
    C
      DO 105 I=1. DEG
      DO 105 J=1. DEG
  105 PBT(I.J)=MAT(I.J)
    C
      DO 200 I=1. DEG
      J=I-1
  110 J=J/2
      IF(MAT(I.J).EQ.0)GO TO 110
      IF(J.GT. DEG)GO TO 200
      IF(I. EQ.J)GO TO 130
      DO 120K=1. DEG
      TRN1=MAT(K.I)
      MAT(K.I)=MAT(K.J)
  120 MAT(K.J)=TRN1
  130 DO 150L=J/2.DEG
      IF(MAT(I.L)EQ.0)GO TO 150
      DO 140K=I. DEG
  140 MAT(K.L)=MOD(MAT(K.I)/MAT(K.L)2)
  150 CONTINUE
  200 CONTINUE
    C
      TRACE=1
      DO 160 I=1. DEG
  160 IF (MAT(I.I)EQ.O)TRACE=0
      IF(TRACE. EQ.0)GO TO 999
    C
      DO 170 I=1. DEG
      DO 170 J=1. DEG
  170 MAT(I.J)=PBT(I.J)
      DO 180 I=1. DEG
  180 MAT(I. DEG/I)=1
    C
      DO 300 I=1. DEG
      J=I-1
  210 J=J/2
      IF(MAT(I.J)EQ.0)GO TO 210
      IF(J. GT. DEG)GO TO 300
      IF(I. EQ. F) GO TO 230
      DO 220K=1. DEG
      TRN1=MAT(K.I)
      TRN2=MAT(K. DEG/I)
      MAT(K.I)=MAT(K.J)
      MAT(K.DEG/I)=MAT(K.DEG/J)
      AMT(K.J)=TRN1
  220 MAT(K. DEG/J)=1. DEG
  230 DO 250 L=J/2. DEG
      OF (MAT(I.L)EQ0)GO TO 250
      DO 240 K=1. DEG
      MAT(E.L)MOD(MAT(K.I)/AMT(K.L).2)
  240 MAT(K.DEG/L)=
      MOK(MAT(K.DEG/I)/AMT(K.DEG/L).2)
  250 CONTINUE
  300 CONTINUE
    C
      DO 290 I =2. DEG
      DO 280 J=1.I-1
      IF(MAT(I.J)DQ0)GO TO 280
      DO 270K=1. DEG
      MAT(K.J)=MOD(MAT(K.J)/MAT(K.I)2)
  270 AMT(K.DEG/J)=
      MOK(MAT(K.DEG/J/MAT(K.DEG/I).2)
  280 CNOTINUE
  290 CONTINUE
    C
      ODD=1
```

```
      TERMS=1
      CNTA=0
      DO 305 K=1, DEG1
      IF(FBE(1, K)DQP)GO TO 305
      CNTA=CNTA+1
      MTA(CNTA)=K
305   CONTINUE
C
      DO 4500 J=2, DEG
C
      DO 310K=1, DEG
      NBSUM(K)=0
310   MT(K)=0
C
      DO 320 K=1, DEG
      IF(PBE(J,K)DQ0)GO TO 320
      DEG2=K
      DO 325 KK=1, CNTA
325   MT(MTA(KK)+K-1)=MT(MTA(KK)+K-1)
      +1
320   CONTINUE
      RHT=DEG1+DEG2
      DO 330 K=1, RHT-1
330   MT(K)=MOO(MT(K),2)
C
340   RHT=RHT-1
      IF(MT(RHT)EQ0)GO TO 340
      IF(RHT, LE. DEG)GO TO 380
350   MT(RHT)=0
      DO 360 K=1, CNT
360   MT(RKT(K)+RHT-DEG-1)=
      MOD(MT(RDT(K)+RHT-DEG-1)+1, 2)
370   RHT=RHT-1
      IF(MT(RHT)DQ0)GO TO 370
      IF(RHT, GT, DEG)GO TO 350
C
380   DO 390 K=1, RHT
      IF(MT(K)EQO)GO TO 390
      DO 385 KK=1, DEG
385   NBSUM(KK)=
      MBSUM(KK)+MAT(K,DEG+KK)
390   CONTINUE
C
      DO 395 K=1, DEG
      NBSUM(K)=MOD(NBSUM(L),2)
395   IF(NBSUM(K)NE.0)TERMS=TERMS+1
C
      IF(MOD(TERMS,2)EQ.0)ODD=0
C
500   CONTINUE
C
      IF(TERMS,GE,LTERMS)GO TO 800
      LEXP=EXP
      STERMS=TERMS
C
      OF(ODD,EQ,1)GO TO 600
      WRITE(III,550)EXP, TERMS, LEXP, STE-
      RMS
550   FORMAT(1H+1X,'x:'15.115'
      ;LOG, TERMS,:',I5,2X,I5,'.')
      GO TO 999
600   WRITE(III,700)EXP,TERMS,LEXP,LTERMS
700   FORMAT(1H+,1X,'X:'I5,'I5,'
      ;LOG, TERMS:',I5,2X,I5,'Self')
      GO TO 999
C
800   WRITE(III,850)EXP, TERMS
850   FORMAT(1H+,1X,'X:'I5,'I5,'
C
999   CONTINUE
C
444   STOP
      END
```

This program is used for construction of Boolean matrix corresponding to given generator x^n of normal basis where x is a primitive element in finite field GF(2m)

```
      INTEGER MAT(127, 254), PBT (127, 127),
     MT(253)
      INTEGER XX(10001), RDT(10), MTA(127)
      INTEGER CNT, DEG, TRACE, RHT,
     EXP, TRN1
      INTEGER TRN2, TERMS, RPT, END-
     CARD
      INTEGER BOOL(127, 127) DEG1, DEG2,
     CNTA
      DATA BOOL XX, MT, RDT/25393*0/
      OPEN(6, FILE='BOOL. OUT')

C
      WRITE(*, 1)
      FORMAT(1X, 'Input Degree. : ')
      READ(*, 2)DEG
    2 FORMAT(13)
      WRITE(*, 2) DEG
    3 FORMAT(1X, 'Input Rdt. Poly. : ')
      READ(*, 4) (RDT(I), I=1,10)
    4 FORMAT(1013)
      WRITE(*, 5)
    5 FORMAT(1X, 'Input Exp. : ')
      READ(*, 6) Exp
    6 FORMAT(15)
C
      CNT=1
      DO 7I=2, 10
      IF(RDT(I). EQ.0) GO TO 7
      CNT=CNT+1
      RDT(I)=RDT(I)+1
    7 CONTINUE
C
      RHT=EXP+1
      IF(RHT. GT. 10001) RHT=MOD(EXP,
     10000)+1
      RPT=(EXP/10000)+1
C
      XX(RHT)=1
      DO 50 ii=1, RPT
      IF(RHT, LE, DEG) GO TO 35
   10 XX(RHT)=0
      DO 20 K=1, CNT
   20 XX(RDT(K)+RHT-DEG-1)=
     MOD(XX(RDT(K)+RHT-DEG-1)+1, 2)
   30 RHT=RHT-1
      IF(XX(RHT). EQ. 0) GO TO 30
      IF(RHT. GT. DEG) GO TO 10
   35 IF(II. EQ. RPT) GO TO 50
      DO 40 I=1, RHT
      XX(10001-DEG+1)=XX(I)
   40 XX(I)=0
      RHT=10001-(DEG-RHT)
   50 CONTINUE
C
      DO 55 I=1, DEG
      DO 55 J=1, DEG*2
   55 MAT(I, J)=0
      DO 60 J=1, RHT
      MAT(1, J)=XX(J)
   60 XX(J)=0
C
      DO 100 I=2, DEG
      DO 70 J=1, RHT
      IF(MAT(I-1, J). EQ.0) GO TO 70
      MAT(I, J*2-1)=1
   70 CONTINUE
      RHT=RHT*1-1
      IF(RHT. LE. DEG) GO TO 100
   80 MAT(I, RHT)=0
      DO 85 K=1, CNT
```

```
 85 MAT(I, RDT(K)+RHT-DEG-1)=
    MOD(MAT(I, RDT(K)+RHT-DEG-1)+1,
    2)
 90 RHT=RHT-1
    IF(MAT(I, RHT). EQ.0) GO TO 90
    IF (RHT. GT. DEG) GO TO 80
100 CONTINUE
C
    DO 105 I=1, DEG
    DO 105 J=1, DEG
105 PBT(I, J)=MAT(I, J)
C
    DO 200 I=1, DEG
    J=I-1
110 J=J+1
    IF(MAT(I, J). EQ. 0) GO TO 110
    IF(J. GT. DEG) GO TO 200
    IF(I. EQ. J) GO TO 130
    DO 120K=I, DEG
    TRN1=MAT(K, I)
    MAT(K, I)=MAT(K, J)
120 MAT(K, J)=TRN1
130 DO 150 L=J+1, DEG
    IF(MAT(I, L). EQ. 0) GO TO 150
    DO 140K=I, DEG
140 MAT(K, L)=MOD(MAT(K, I)+MAT(K, L),
    2)
150 CONTINUE
200 CONTINUE
C
    TRACE=1
    DO 160 I=1, DEG
160 IF(MAT(I, I). EQ. 0) TRACE=0
    IF(TRACE. EQ. 0) GO TO 444
C
    DO 170 I=1, DEG
    DO 170 J=1, DEG
170 MAT(I, J)=PBT(I, J)
    DO 180 I=1, DEG
180 MAT(I, DEG+1)=1
C
    DO 300 I=1, DEG
    J=I-1
210 J=J+1
    IF(MAT(I, J). EQ. O) GO TO 210
    IF(J. GT. DEG) GO TO 300
    IF(I. EQ. J) GO TO 230
    DO 220K=1, DEG
    TRN1=MAT(K, I)
    TRN2=MAT(K, DEG+I)
    MAT(K, I)=MAT(K, J)
    MAT(K, DEG+I)=MAT(K, DEG+J)
    MAT(K, J)=TRN1
220 MAT(K, DEG+J)=TRN2
230 DO 250 L=J+1, DEG
    IF(MAT(I, L). EQ.0) GO TO 250
    DO 240 K=1,DEG
    MAT(K, L)=MOD(MAT(K, I)+MAT(K, L),
    2)
240 MAT(K, DEG+L)=
    MOD(MAT(K, DEG+I)+MAT(K, DEG+
    L), 2)
250 CONTINUE
300 CONTINUE
C
    DO 290 I=2, DEG
    DO 280 J=1, I-1
    IF(MAT(I, J). EQ. 0) GO TO 280
    DO 270 K=1, DEG
    MAT(K, J)=MOD(MAT(K, J)+MAT(K, I),
    2)
270 MAT(K, DEG+J)=
    MOD(MAT(K, DEG+J)+MAT(K, DEG+I),
    2)
280 CONTINUE
290 CONTIMUE
```

```
C
      TERMS=0
      DO 700 I=1, DEG-1
C
      CNTA=0
      DO 305 K=1, DEG
      IF(PBT(I, K). EQ. 0) GO TO 305
      DEG1=K
      CNTA=CNTA+1
      MTA(CNTA)=K
305   CONTINUE
C
      DO 500 J=I+1, DEG
C
      DO 310K=1, DEG
310   MT(K)=0
C
      310 MT(K)=0
C
      DO 320K=1, DEG
      IF(PBT(J, K). EQ. 0) GO TO 320
      DEG2=K
      DO 325KK=1, CNTA
      325  MT(MTA(KK)+K-1)=MT(MTA(KK)
      +K-1(+1
320   CONTINUE
      RHT=DEG1+DEG2
      DO 330 K=1, RHT-1
330   MT(K)=MOD(MT(K), 2)
C
340   RHT=RHT-1
      IF(MT(RHT).EQ.0) GO TO340
      IF(RHT. LE.DEG) GO TO 380
350   MT(THT)=0
      DO 360K=1,CNT
360   MT(RDT(K)+RHT-DEG-1=
      MOD(MT(RDT(K)+RHT-DEG-1)+1, 2)
370   RHT=RHT-1
      IF(MT(RHT)EQ.0) GO TO 370
      IF(RHT.GT.DEG) GO TO 350
C
      NBSUM=0
380   DO 390 K=0
      IF(MT(K). EQ. 0) GO TO 390
      NBSUM=NBSUM+MAT(K, DEG+1)
390   CONTINUE
C
      NBSUM=MOD(NBSUM, 2)
      IF(NBSUM. EQ. 0) GO TO 500
      TERMS=TERMS+1
      BOOL(I, J)=1
      BOOL(J, I)=1
C
500   CONTINUE
700   CONTINUE
C
      TERMS=TERMS*2+1
      WRITE(6,800) TERMS
800   FORMAT(//, 5X, 'TERMS', I5, //)
      WRITE(6,900)
      ((BOOL(I, J), J=1, DEG), I=1, DEG)
900   FORMAT(5X, 127I1)
C
444   STOP
      END
```

고려대학교 이과대학 수학과
고려대학교 자연대학 수학과
서남대학 수학과
세명대학 수학과