通信情報保護學會論文誌
第 11 券 第 1 號, 2001. 2

GF(q)상의 원시다항식 생성에 관한 연구

최 희 봉*, 원 동 호**

On algorithm for finding primitive polynomials over GF(q)

Hee-bong Choi*, Dong-ho Won**

요 약

*GF(q)*상의 원시다항식은 스크램블러, 에러정정 부호 및 복호기, 난수 발생기 그리고 스트림 암호기 등 여러 분야에 걸쳐 많이 사용되고 있다. *GF(q)*상의 원시다항식을 생성하는 효율적인 알고리즘이 A.D. Porto에 의하여 제안되었으며, 그 알고리즘은 한 원시다항식을 이용하여 다른 원시다항식을 구하는 방법을 반복 사용하여 원시다항식 수열을 생성하는 방법이다. 이 논문에서는 A.D. Porto가 제안한 알고리즘을 개선한 알고리즘을 제안하였다. A.D. Porto의 알고리즘의 running time은 $O(km^2)$이고, 개선된 알고리즘의 running time은 $O(m(m+k))$이다. 여기서 k는 $\gcd(k, q^m-1)$이다. m차 원시다항식을 구하고자 할 때 $k, m \gg 1$ 조건에서는 개선된 알고리즘을 사용하는 것이 효율적이다.

ABSTRACT

The primitive polynomial on *GF(q)* is used in the area of the scrambler, the error correcting code and decode, the random generator and the cipher, etc. The algorithm that generates efficiently the primitive polynomial on *GF(q)* was proposed by A.D. Porto. The algorithm is a method that generates the sequence of the primitive polynomial by repeating to find another primitive polynomial with a known primitive polynomial. In this paper, we propose the algorithm that is improved in the A.D. Porto algorithm. The running time of the A.D. Porto algorithm is $O(km^2)$, the running time of the improved algorithm is $O(m(m+k))$. Here, k is $\gcd(k, q^m-1)$. When we find the primitive polynomial with m oder, it is efficient that we use the improved algorithm in the condition $k, m \gg 1$.

keyword : *primitive polynomial, linear feedback shift register, Berlekamp-Massey algorithm*

Ⅰ. 서 론

원시다항식은 난수 발생기, 에러 정정 부호 및 복호기 그리고 암호기와 통신 장비를 설계하는데 매우 유용하게 사용하고 있다. 공개키 암호체계에서는 암호화 및 복호화 과정의 곱셈과 제곱의 계산량을 작도록 설계하기 위해 적절한 원시다항식을 이용한 연산회로 구성방법에 대한 연구를 많이 하고 있다.[1] 스트림 암호체계에서도 원시다항식을 특성다항식으로 하는 linear feedback shift register가 많이 사용되고 있다.[2] 원시다항식을 찾는 알고리즘은 일반적으로 두 가지 방법으로 분류된다.[3] 첫째 방법은 임의로 주어진 다항식이 원시다항식인가 아닌가를 판정하는 알고리즘을 이용한 것이고, 둘째 방법은 알고 있는 한 개의 원시다항식으로부터 새로운 원시다항식을 얻는 알고리즘을 이용하는 것이다. 첫째 방법을 이용하여 원시다항식을 구하는 알고리즘을 참고문헌[4,5,6]등에서 찾을 수 있다. 대부분의 알고리즘의 실행여부는 q^m-1를 소인수 분해할 수 있는가 하는 문제에 달려 있다.[7] 이 문제에 관해서

* 국가보안기술연구소(hbchoi@etri.re.kr)
** 성균관대학교 전기전자 및 컴퓨터공학부 교수(dhwon@ssku.ac.kr)

Shoup가 논문을 발표하였다.[8] 그러나 이 방법으로는 원시다항식을 구하는 시간이 많이 걸리기 때문에 다양한 응용분야에 활용하기에 충분한 종류를 구할 수 없다. 둘째 방법을 이용하면 필요한 충분한 양의 원시다항식을 가능한 시간 내에 구해낼 수 있다.

A. D. Porto[9]는 둘째 방법으로 이미 알고 있는 원시다항식을 이용하여 선형방정식을 유도하고 Berlekamp-Messey 알고리즘[8]을 적용하여 새로운 원시다항식을 생성하였다.

이 논문에서는 A. D. Porto가 제안한 알고리즘 중 선형방정식을 유도한 계산방법을 변형하여 개선된 원시다항식 생성 알고리즘을 제안하였다. A. D. Porto의 알고리즘의 running time 은 $O(km^2)$이고 개선된 알고리즘의 running time 은 $O(m(m+k))$이다. 여기서 k는 $\gcd(k, q^m-1)=1$를 만족한다. running time에서 볼 수 있듯이 m차 원시다항식을 구하고자 할 때 $k, m \gg 1$ 조건에서는 개선된 알고리즘을 사용하는 것이 효율적이다.

이 논문의 구성은 다음과 같다. Ⅱ장에서는 원시다항식의 기본개념을 설명하고 A. D. Porto가 제안한 알고리즘을 설명하며, Ⅲ장에서는 개선된 새로운 알고리즘 제안하고, Ⅳ장에서는 제안한 알고리즘과 A. D. Porto의 알고리즘과 비교 분석하고, Ⅴ장에서 결론을 내린다.

Ⅱ. 원시다항식

2.1 기본 개념

체(field)는 덧셈, 곱셈, 나눗셈이 가능한 원소들의 집합이다. 즉 체(F, +, ·)는 F상의 이항 연산기 +, ·를 갖으며 어떤 원소 $a, b, c \in F$에 대해 다음과 같은 성질을 만족하는 집합이다[3].

(1) $(a+b)+c=a+(b+c)$(덧셈의 결합법칙)
(2) $a+b=b+a$(덧셈의 교환법칙)
(3) $a+0=a$이 성립하는 $0 \in F$이 존재한다.(덧셈의 항등원)
(4) $a+(-a)=0$이 성립하는 $-a \in F$이 존재한다.(덧셈의 역원)
(5) $(a \times b) \times c=a \times (b \times c)$(곱셈의 결합법칙)
(6) $a \times b=b \times a$(곱셈의 교환법칙)
(7) $a \times 1=a$(곱셈의 항등원)
(8) $a \times a^{-1}=1$이 성립하는 $a \neq 0$이고 $a^{-1} \in F$이 존재한다.(곱셈의 역원)
(9) $a \times (b+c)=a \times b+a \times c$(배분법칙)

원소의 개수가 유한개로 구성되는 체를 유한체라 한다. 유한체는 Galois Field로도 불린다. Galois Field는 GF(q)로 쓰며 q는 소수(prime) 또는 소수의 멱승(power)이다. 원소의 개수가 무한개로 구성되는 체를 무한체라 한다. 무한체의 예로서는 유리수, 실수를 들 수 있다.

원시다항식의 개념을 알기 위하여 다음과 같은 사항들을 설명한다.

(정의 1) $GF(q)$를 소수 q의 residue set {0, 1, 2, ... , q-1} 인 유한체로 정의한다.[3]

(정의 2) 0을 제외한 유한체의 모든 원소를 지수성으로 표시할 수 있는 원소를 원시원소(primitive element)라 한다.[3]

아래의 (정리 1)부터 (정리 2)까지는 증명없이 요약하였으며 원시다항식을 구성하는데 사용된다.

(정리 1) $p=bc$일 때 b가 상수다항식이거나 c가 상수다항식일 때 p를 $GF(q)$상의 기약다항식irreducible polynomial)이라 한다. 여기서 $p, b, c \in GF(q)[x]$이다.[3]

(정리 2) $GF(q^m)$를 기초체 (base field) $GF(q)$상의 확대체 (extension field)라 하면 $GF(q^m)$는 $GF(q)$상의 m차 기약 다항식의 residue set 이다. β를 $GF(q^m)$의 한 원소라 하자. $GF(q)$상에서 $g(\beta)=0$가 되는 가장 낮은 차수의 기약다항식 g(x)를 β의 최소다항식(minimal polynomial)이라 한다. 또한 $GF(q^m)$의 모든 원소에 대한 최소다항식은 항상 존재하고 유일하다.[3]

(정의 3) 확대체 $GF(q^m)$의 원시원소를 근으로 하는 최소다항식은 모두 m차 원시다항식이다.[3]

(정리 3) α를 확대체 $GF(q^m)$의 원시원소라 하자. $\gcd(k, q^m-1)=1$(gcd : greatest common divisor)인 모든 정수 k에 대해 α^k는 $GF(q^m)$의 원시원소이고, 이 원소의 최소다항식 $g_k(x)=$

$x^m+\sum_{i=0}^{m-1}c_ix^i$은 m차 원시다항식이다. 여기서 c_i는 $GF(q)$의 원소이다.

2.2 원시다항식 유도

(정리 3)에서 원시다항식 $g_k(x)$는 아래식을 만족한다.

$$g_k(\alpha^k)=\alpha^{km}+c_{m-1}\alpha^{k(m-1)}+c_{m-2}\alpha^{k(m-2)}+\cdots+c_0=0 \quad (1)$$

또한 α^h를 아래 식과 같이 표시할 수 있다.

$$\alpha^h=a_{m-1}^{[h]}x^{m-1}+a_{m-2}^{[h]}x^{m-2}+\cdots+a_0^{[h]} \quad (2)$$

여기서 $0\le h\le q^m-1$이다. $a_i^{[h]}\in GF(q)$를 $\alpha^h(i=0,1,2,\cdots,m-1)$의 i번째 성분이라 한다.

식 (2)을 식 (1)에 대입하여 계산하면 α^{km}의 각 성분에 대한 방정식을 얻을 수 있다. 이 방정식은 m개의 미지수를 갖는 아래와 같은 m개 선형방정식이다.

$$\begin{aligned}
a_0^{[km]}&=-c_{m-1}a_0^{[k(m-1)]}-c_{m-2}a_0^{[k(m-2)]}-\cdots-c_0a_0^{[0]}\\
a_1^{[km]}&=-c_{m-1}a_1^{[k(m-1)]}-c_{m-2}a_1^{[k(m-2)]}-\cdots-c_0a_1^{[0]}\\
&\cdots\cdots\\
a_{m-1}^{[km]}&=-c_{m-1}a_{m-1}^{[k(m-1)]}-c_{m-2}a_{m-1}^{[k(m-2)]}-\cdots-c_0a_{m-1}^{[0]}
\end{aligned} \quad (3)$$

α를 근으로 하는 원시 다항식으로부터 α^k, α^{2k}, $\cdots$, α^{mk}를 계산하여 계수 $a_i^{[h]}(i=0,1,2,\cdots,m-1)$, $(h=0,k,2k,\cdots,mk)$를 얻는다. 계수 $a_i^{[h]}$를 식 (3)에 대입하여 $g_k(x)$의 계수 $c_0,c_1,\cdots,c_{m-1}$를 구한다. 이를 계산하는 일반적인 방법으로는 matrix inversion이 있다.

그런데 방정식(1)의 특수한 구조를 이용하여 계수 $c_0,c_1,\cdots,c_{m-1}$를 matrix inversion으로 구한 것보다 훨씬 효율적으로 구하는 방법을 A. D. Porto가 제안하였다.[9] 즉 식 (1)에 $\alpha^{uk}, u=0,1,2,\cdots$를 곱하고 이 곱한 식에 식 (2)를 대입하여 j번째 성분에 대해서만 정리하면 아래와 같은 식을 얻는다.

$$a_j^{[k(m+u)]}=-\sum_{t=1}^{m}c_{m-t}a_j^{[k(m+u-t)]},\ u=0,1,2,\cdots \quad (4)$$

$s_i=a_j^{[ki]}, i=0,1,2,\cdots$로 두고 식 (4)를 아래와 같이 다시 쓴다. 여기서 s_i는 알고 있는 원시다항식으로부터 구할 수 있다.

$$s_i=-\sum_{t=1}^{m}e_ts_{i-t},\ i=0,1,2,\cdots \quad (5)$$

여기서 e_t를 아래와 같은 등식으로 두었다.

$$e_t=c_{[m-t]} \quad (6)$$

식 (5)에서 $i=m,m+1,\cdots,2m-1$로 두면 다음과 같은 m개의 방정식을 얻을 수 있다.

$$s_{(m+v)}=-\sum_{t=1}^{m}e_ts_{m+v-t},\ v=0,1,2,\cdots,m-1 \quad (7)$$

방정식 (7)은 m개의 미지수 $e_t, t=1,2,\ldots,m$를 갖는 m개의 선형 방정식이며 Berlekamp-Massey 알고리즘을 이용하여 풀 수 있다. 식 (6)으로부터 원시다항식 $g_k(x)$을 얻을 수 있다. 식 (3)은 running time이 $O(m^3)$인 matrix inversion 알고리즘으로 구하는 데 비하여 식 (7)은 running time이 $O(m^2)$인 Berlekamp-Massey 알고리즘으로 구할 수 있으므로 매우 효율적이다.

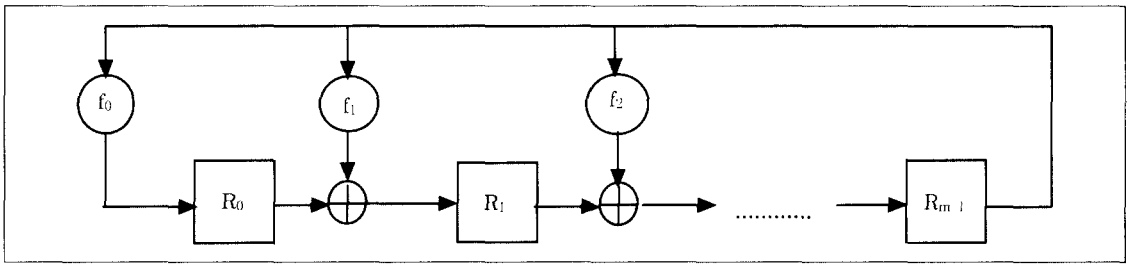
2.3 A. D. Porto의 원시다항식 생성 알고리즘

A. D. Porto가 제안한 원시다항식을 생성하는 알고리즘은 다음과 같다.

Step 1 : $GF(q)$상의 m차 원시다항식 $f(x)$에 대하여 $x^i \bmod f(x)$를 계산하기 위하여 특성 다항식이 $f(x)$인 [그림 1]의 shift register를 구성한다.

Step 2 : [그림 1]의 shift register state를 (1 0 0 $\cdots$ 0)로 초기화한다.

Step 3 : 임의의 정수 $j, 0\le j\le m-1$를 선택하여 $GF(q)$의 m개의 원소 $s=a_j^{[kh]}$, $h=0$,



〔그림 1〕 $x^i \bmod f(x)$를 계산하는 shift register

$1, \cdots, 2m-1$를 얻기 위해 〔그림 1〕의 shift register를 0에서부터 $2km-1$번 shift하여 k번 shift할 때마다 해당 R_j의 내용을 읽는다.

Step 4 : 미지수 $e_t, t=1,2,\ldots,m$를 갖는 선형 방정식 (7)를 Berlekamp-Massey 알고리즘으로 푼다. 식 (6)으로부터 원시다항식 $g_k(x)$의 계수 $c_0, c_1, \cdots, c_{m-1}$를 구한다.

Step 5 : 새로운 원시다항식을 구하기 위하여 $f(x)$를 $g_k(x)$로 두고 Step 1부터 다시 시작한다.

위와 같은 Step을 반복 수행하면 원시다항식열 $g_{k^i}(x), i=2,3,\cdots$을 얻는다. 그리고 이 원시다항식열(primitive polynomial sequence)은 주기성을 가지며 이 원시다항식열의 주기를 구할 수 있다.[13]

아래 〔표 1〕는 $q=2, m=5$에 대하여 한 원시다항식과 k를 $f(x)=x^5+x^2+1, k=3$으로 선택하여 앞의 원시다항식 생성 알고리즘을 이용하여 원시다항식을 구한 것이다. 〔표 1〕에서 α는 $GF(2^5)$상의 원시원소이며 $f(\alpha)=0$이다.

〔표 1〕 원시다항식과 k가 $f(x), k=3$ 일 때 생성되는 원시다항식들

i	$f(x)$	원시원소	$g_{k^i}(x)$
1	x^5+x^2+1	α^3	$x^5+x^4+x^3+x^2+1$
2	$x^5+x^4+x^3+x^2+1$	α^3	$x^5+x^4+x^2+x+1$
3	$x^5+x^4+x^2+x+1$	α^3	x^5+x^3+1
4	x^5+x^3+1	α^3	$x^5+x^3+x^2+x+1$
5	$x^5+x^3+x^2+x+1$	α^3	$x^5+x^4+x^3+x+1$
6	$x^5+x^4+x^3+x+1$	α^3	x^5+x^2+1

Ⅲ. 개선 알고리즘

3.1 개선 알고리즘 유도

이 절에서는 A. D. Porto의 원시다항식 생성 알고리즘을 개선한 알고리즘을 유도한다. m차 다항식 $f(x)=c_0+c_1x+c_2x^2+\cdots+x^m$대하여 $x^i \bmod f(x)$를 구하는 방법을 $GF(2)$상에서 수식으로 전개하면 다음과 같다. $m-1$차 이하 다항식 $g(x)=a_0+a_1x+a_2x^2+\cdots+a_{m-1}x^{m-1}$에 x를 곱하여 정리하면 다음과 같다.

$$\begin{aligned} g(x)x &= a_0x+a_1x^2+a_2x^3+\ldots+a_{m-1}x^m \\ &= \left[\begin{array}{c}0\\ \oplus\\ c_0a_{m-1}\end{array}\right]+\left[\begin{array}{c}a_0\\ \oplus\\ c_1a_{m-1}\end{array}\right]x+\left[\begin{array}{c}a_1\\ \oplus\\ c_2a_{m-1}\end{array}\right]x^2 \\ &+\cdots+\left[\begin{array}{c}a_{m-2}\\ \oplus\\ c_{m-1}a_{m-1}\end{array}\right]x^{m-1} \end{aligned} \quad (8)$$

위 수식을 m단 shift register로 구성하면 register를 한번 shift하고 a_{m-1}이 1일 때 $c_0, c_1, \cdots, c_{m-1}$을 각 register에 exclusive-or하면 $g(x)x$를 구할 수 있다. 이 때 $g(x)$를 1로 하고 위 방법을 반복 적용하면 모든 양의 정수 i에 대하여 $x^i \bmod f(x)$를 구할 수 있다.

(정의 4) 임의의 다항식 $g(x)=a_0+a_1x+a_2x^2+\cdots+a_k^k$와 $m \le k$에 대하여 다항식 truncation 함수 T_m은 다음과 같다.

$$T_m[g(x)] = a_0+a_1x+a_2x^2+\ldots+a_{m-1}x^{m-1} \quad (9)$$

(정의 4)에 따라 $T_m[g(x)x]$는 $g(x)x$에서 m차 이상의 항을 버린 것이다. 식 (8)을 간단히 쓰면

$$g(x)x = T_m[g(x)x] + a_{m-1}(c_0 + c_1x + \ldots + c_{m-1}x^{m-1}) \quad (10)$$

이다.

임의의 m차 다항식 $f(x)$와 양의 정수 k에 대하여 $x^{ki} \bmod f(x)$를 구하는 보다 빠른 방법은 다음과 같다. 우선 $x^m \bmod f(x), x^{m+1} \bmod f(x), x^{m+2} \bmod f(x), \cdots, x^{m+k-1} \bmod f(x)$을 일반적인 방법으로 먼저 구한다. $x^{m+k-1} \bmod f(x) = h_i(x), i = 0, 1, \cdots, k-1$라 두면 $m-1$차 $g(x)x^k$ 다항식은 다음과 같이 표시할 수 있다.

$$g(x)x^k = a_0x^k + a_1x^{k+1} + a_2x^{k+2} + \cdots + a_{m-1}x^{m+k-1} \quad (11)$$

$$= T_m[g(x)x^k] + a_{m-k}h_0(x) + \cdots + a_{m-1}h_{k-1}(x) \quad (12)$$

식 (12)은 m단 shift register로 구성할 수 있는데 $x^m \bmod f(x), x^{m+1} \bmod f(x), x^{m+2} \bmod f(x), \cdots, x^{m+k-1} \bmod f(x)$를 구하여 shift register k개에 각각 저장한다. register를 k번 shift하고 $8a_i, (i = m-k, \cdots, m-1)$가 1일 때 그에 해당되는 m단 register를 서로 exclusive-or하면 $g(x)x^k$를 구할 수 있다. 이 때 $g(x)$를 1로 하고 이 방법을 반복적용하면 양의 정수 i에 대하여 $x^{ki} \bmod f(x)$를 구할 수 있다.

3.2 개선된 원시다항식 생성 알고리즘

원시다항식을 유도하는 것은 A.D. Porto의 알고리즘과 동일하지만 Step 1에서 Step 3 까지 사용되는 shift register를 [그림 2]와 같이 변형하여 특정 조건에서 시간상 개선된 생성 알고리즘이다. 그 알고리즘은 다음과 같다.

Step 1 : $GF(q)$상의 m차 원시다항식 $f(x)$에 대하여 특성다항식이 $f(x)$인 [그림 1]의 shift register를 이용하여 [그림 2]의 shift register를 아래와 같이 구성한다. [그림 2]의 register $R_{m-k}, \cdots, R_m$으로부터 귀환되는 tap은 다음과 같이 구한다. register R_{m-k}에서 귀환되는 tap은 [그림 1]의 register R_{m-k}에 1로 하고 나머지 register에 0으로 초기화한 후 k번 shift하여 각 register에 포함된 내용으로 한다. register R_{m-k+1}에서 귀환되는 tap은 위에서 shift한 것을 한번 더 shift하여 각 register에 포함된 내용으로 한다. 이와 같이 하여 register R_{m-1}까지 귀환되는 tap을 구할 수 있다.

Step 2 : [그림 2]의 state를 (1 0 0 … 0)로 초기화한다.

Step 3 : 임의의 정수 $j, 0 \le j \le m-1$를 선택하여 $GF(q)$의 $2m$개의 원소 $s_j = a_j^{[kh]}, h = 1, 2, \cdots, 2m-1$를 얻기 위해 [그림 2]의 shift register를 0에서부터 $2m-1$번 shift하여 shift할 때 마다 해당 R_j register의 내용을 읽는다.

Step 4 : 미지수 $e_t, t = 1, 2, \cdots, m$를 갖는 선형방정식 (7)을 Berlekamp-Massey 알고리즘으로 푼다. 식 (6)으로부터 원시다항식 $g_k(x)$의 계수 $c_0, c_1, \cdots, c_{m-1}$를 구한다.

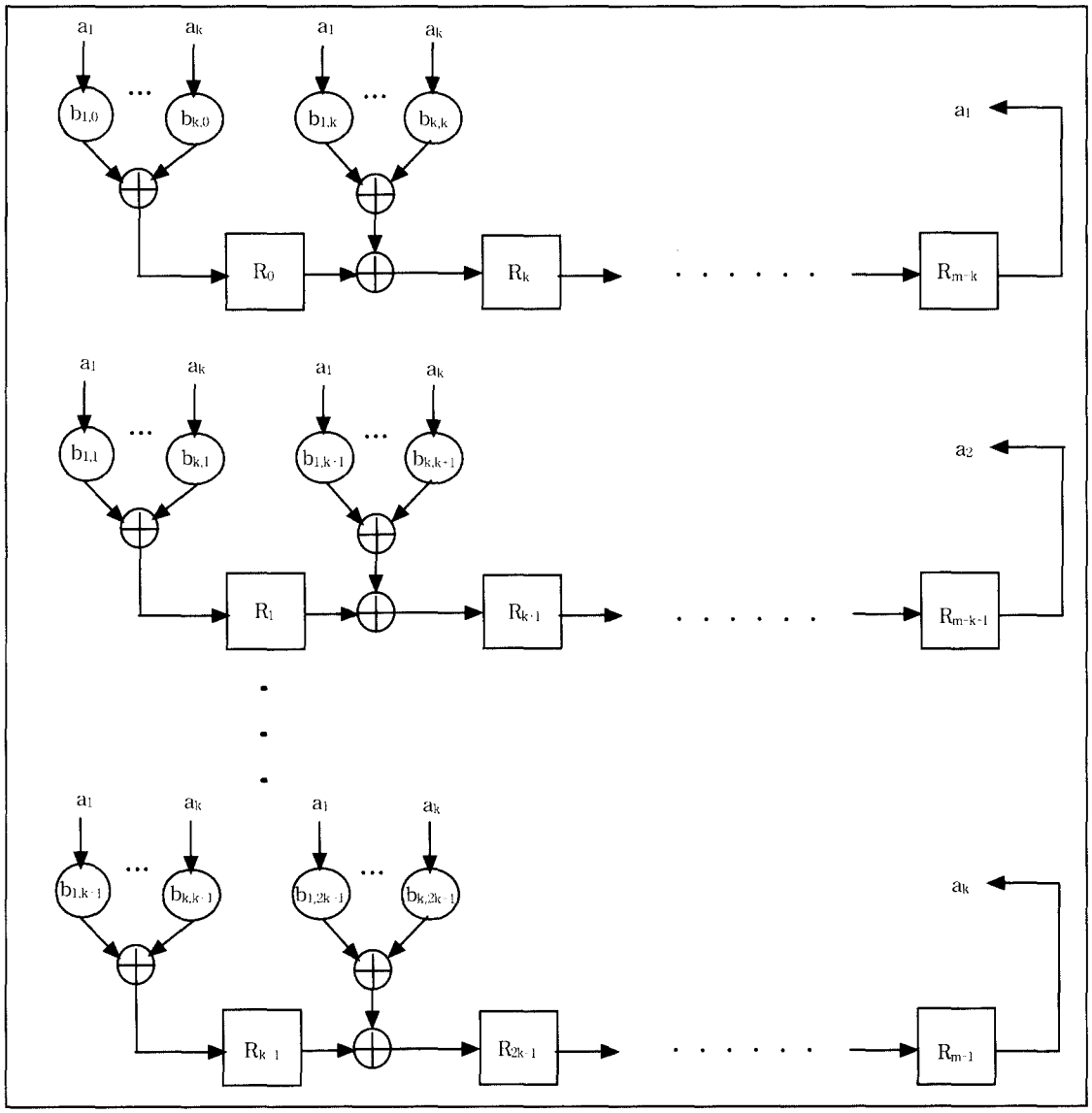
Step 5 : 새로운 원시다항식을 구하기 위하여 $f(x)$를 $g_k(x)$로 두고 Step 1부터 다시 시작한다.

위와 같은 Step을 계속 수행하면 원시다항식 $g_k(x), i = 2, 3, \cdots$열을 얻는다.

Ⅳ. 원시다항식 생성 알고리즘 비교 분석

A. D. Porto 알고리즘의 running time은 다음과 같다. 2.3절의 알고리즘 Step 3에서 [그림 1]의 m단 shift register을 $2km$번 shift하기 때문에 running time은 $O(km^2)$이고 Step 4의 Berlekamp-Massey 알고리즘을 수행하는 running time은 $O(m^2)$이다.[8,9] 따라서 이 두 running time의 합은 $O(km^2)$이다.[9]

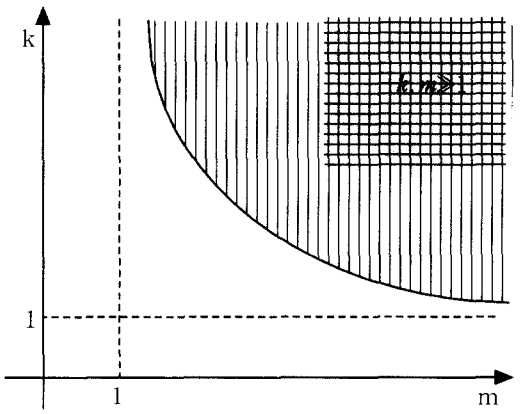
개선된 알고리즘의 running time은 다음과 같다. 3.2절의 알고리즘 Step 1에서 [그림 2]의 m단의 shift register의 feedback tap을 구하기 위한



〔그림 2〕 $x^{ki} \bmod f(x)$를 계산하는 shift register

계산으로서 〔그림 1〕의 m단 shift register중 register R_{m-k}에 1로 초기화하고 k번 shift하기 때문에 running time은 $O(km)$이고 추가로 한 번씩 shift하는 running time은 $O((k-1)m)$이 된다. 따라서 Step 1의 전체 running time은 $8(km)$이다. Step 3에서 〔그림 2〕의 k개의 shift register를 $2m$번 shift한다. 그런데 k개의 shift register들은 전체로 합하면 m단 register이기 때문에 running time은 $O(m^2)$이 된다. Step 4는 Berlekamp-Massey 알고리즘을 수행하기 때문에 running time은 $O(m^2)$이다.[8] 개선된 알고리즘의 전체 running time은 $O(km+m^2+m^2)$이다. 이것을 정리하면 running time은 $O(m(m+k))$이 된다. 〔그림 3〕은 두 알고리즘의 running time을 비교하기 위한 것이다. 〔그림 3〕에서 빗금 친 부분 즉 $k, m \gg 1$에서 개선된 알고리즘이 효율적임을 나타낸다.

실제로 $GF(q)$상의 141차 원시다항식을 2000개 생성하기 위하여, C-언어로 펜티엄III-800에서 A. D. Porto 알고리즘과 개선된 알고리즘을 구현하여 걸리는 시간을 〔표 2〕에 나타내었다. 〔표 2〕에서 보는



〔그림 3〕 개선된 알고리즘의 효율적인 영역

〔표 2〕 $GF(q)$상의 141차 원시다항식 생성 시간 비교
(단위: 초)

k	개선한 방법	Porto의 방법
3	8	6
5	12	12
23	38	44
59	88	108
109	154	196

바와 같이 k값이 크면 클수록 개선된 알고리즘이 효율적임을 알 수 있었다. 또한 m값이 클 경우도 개선된 알고리즘이 효율적임을 시뮬레이션을 통하여 확인하였다. 물론 $q=2,3,5,7,11,\cdots$ 에서 $GF(q)$상의 원시다항식을 생성하는 일반적인 경우에 대해서도 똑같이 적용된다.

V. 결 론

본 논문에서는 A. D. Porto가 제시한 원시다항식 생성 알고리즘을 개선하여 A. D. Porto의 원시다항식 생성 알고리즘과 비교하였다. A. D. Porto 알고리즘의 running time이 $O(km^2)$이고 개선된 알고리즘의 running time이 $O(m(k+m))$이기 때문에 이 k, m 값이 크면 매우 효율적임을 알 수 있다. 개선된 알고리즘을 이용하여 고차 원시다항식을 빠르게 많이 생성하여 스트림 암호기나 기타 통신시스템에 편리하게 사용할 수 있다. 그러나 프로그램 개발시 메모리용량이 더 많이 필요하다. 이에 대한 개선방법이 요구된다.

참 고 문 헌

〔1〕 Thomas Beth and Dieter Gollmann, "Algorithm Engineering for Public Key Algorithms", *IEEE J. on Selected Areas in Communications*, Vol. 7, No. 4, pp. 458 ~466, May 1989.

〔2〕 W. Meier and O. Staffelbach, "Fast Correlation Attacks on Certain Stream Cipher", *J. Cryptology (1)*, pp. 159~176, 1989.

〔3〕 R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Application*, Cambridge University Press, Cambridge, 1986.

〔4〕 Wayne Stahnke, "Primitive Binary Polynomials", *Mathematics of Computation*, Vol. 27, No. 124, pp. 977~980, October 1973.

〔5〕 Niel Zierler, "On Primitive Trinomial (Mod 2)", *Inform. and Control*, Vol. 13, pp. 541~554, 1968.

〔6〕 Solomon W. Golomb, *Shift Register Sequences*, Aegean Park Press, California, 1982.

〔7〕 L. Rudolf, "Computational Problems in the Theory of Finite Fields", *AAECC.*, Vol. 2, pp. 81~89, 1991.

〔8〕 V. Shoup, "New Algorithms for Finding Irreducible Polynomials Over Finite Fields", *Math. Comp.*, Vol. 54, pp. 435 ~447, 1990.

〔9〕 A. D. Porto, F. Guida and E. Montolivo, "Fast Algorithm for Finding Primitive Polynomials over $GF(q)$", *Elec. Lett.*, B28, (2), pp. 118~120, 1992.

〔10〕 J. L. Massey, "Shift Register Synthesis and BCH Decoding", *IEEE Trans. on Inform. Theory*, Vol. 15, No. 1, pp. 122 ~127, 1969.

〔11〕 W. W. Adams and L. J. Goldstein, *Introduction to Number Theory*, Prentice-Hall, Inc., Englewood Cliffs, 1976.

〔12〕 J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman and Jr. S. S.

Wagstaff, *Contemporary Mathematics, Factorizations of* $b^n \pm 1$, $b=2,3,4,7,10,11,12$ up to high powers, American Mathematical Society, Providence, 1983.

[13] B. Park, H. Choi, T. Chang, and K. Kang, "Period of Sequences of Primitive Polynomials", *Electron. Lett.*, Vol. 29, pp. 390~391, 1993.

〈著者紹介〉

최 희 봉 (Hee-bong Choi) 정회원
1984년 2월 : 부산대학교 전기공학과 졸업
1987년 2월 : 부산대학교 전기공학과 석사
1997년 3월~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 박사과정
1987년 2월~2000년 1월 : 국방과학연구소 선임연구원
2000년 2월~현재 : 국가보안기술연구소 선임연구원
〈관심분야〉 암호이론, 네트워크보안, 보안시스템 설계

원 동 호(Dong-ho Won) 종신회원
1976년 2월 : 성균관대학교 전자공학과 졸업
1978년 2월 : 성균관대학교 전자공학과 석사
1988년 2월 : 성균관대학교 전자공학과 박사
1978년 4월~1980년 3월 : 한국전자통신연구원 연구원
1985년 9월~1986년 8월 : 일본 동경공대 객원연구원
1982년~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 정교수
〈관심분야〉 암호이론, 정보이론