



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0027175
(43) 공개일자 2011년03월16일

(51) Int. Cl.

G06F 7/52 (2006.01) G06F 7/53 (2006.01)

(21) 출원번호 10-2009-0085148

(22) 출원일자 2009년09월10일

심사청구일자 2009년09월10일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가 1

(72) 발명자

장남수

서울 마포구 성산2동 200-84 2층

김창한

서울시 강남구 대치동 은마아파트 21-507

(뒷면에 계속)

(74) 대리인

특허법인충현

전체 청구항 수 : 총 4 항

(54) 비트-병렬 다항식 기저 곱셈 방법

(57) 요약

본 발명은 유한체(Finite Field)상에서 비트-병렬 다항식 기저 곱셈을 위한 기약다항식(irreducible polynomial)으로서 반복형 기약다항식(RFP: Repeated Form Polynomial)을 이용하는 비트-병렬 다항식 기저 곱셈 방법에 관한 것이다.

본 명세서에서 개시하는 비트-병렬 다항식 기저 곱셈 방법은 유한체(finite field)상에서의 비트-병렬 다항식 기

저 곱셈을 위한 기약다항식 $f(x)$ 가 $f(x) = \sum_{i=0}^s p(x)x^{i \cdot s+v} + l(x)$, $\deg(l(x)) \leq v$, $\deg(p(x)) < s$ 을 만족하는지의 여부로 상기 $f(x)$ 가 반복형 기약다항식임을 판단하여 비트-병렬 다항식 기저 곱셈이 이루어진다. $f(x)$ 가 반복

형 기약다항식임을 판단하는 것은 (a)상기 유한체상의 임의의 다항식 $q(x) = x^r + 1 + \sum_{i=1}^{r-1} q_i x^i$ 에 대해 $x^s = q(x)p(x) + 1$ 을 만족하는 최소의 양의 정수 s 와 상기 $p(x)$ 가 존재하는지 판단하는 단계; (b)상기 $p(x)$ 가 존재하

는 경우, 상기 $q(x)$ 와 상기 유한체상의 임의의 다항식 $h(x)$ 에 대하여 $x^{n_0+r} = q(x)f_{n_0}(x) + h(x)$ 을 만족하는 최

소의 양의 정수 n_0 와 다항식 $f_{n_0}(x) (n_0 = \deg(f_{n_0}(x)))$ 가 존재하는지 판단하는 단계; 및 (c)상기 n_0 와 상기

$f_{n_0}(x)$ 가 존재하는 경우, 상기 $f(x)$ 의 차수(degree) n 에 대하여 $x^{n+r} = q(x)f(x) + h(x)$ 를 만족하는지 판단하는 단계를 포함하여 이루어진다.

대표도 - 도4

$x^{n+r} = h(x) \bmod f(x)$			Space Complexity		Time Complexity Upper Bound
r	$\deg(h(x)) = k_1$	$H(h(x))$	#AND	#XOR	
2	$\leq \lfloor (n+5)/2 \rfloor$	2	n^2	$\leq n^2 + n - 6$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
3	≤ 5	2	n^2	$\leq n^2 + n - 2$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
4	≤ 5	2	n^2	$\leq n^2 + n - 1$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
2	$\lfloor (n+5)/2 \rfloor < k_1 \leq \lfloor (2n+7)/3 \rfloor$	2	n^2	$\leq n^2 + n - 4$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
3	$6 \leq k_1 \leq \lfloor (n+7)/2 \rfloor$	2	n^2	$\leq n^2 + n - 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
4	$6 \leq k_1 \leq \lfloor (n+9)/2 \rfloor$	2	n^2	$\leq n^2 + n + 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
5,6,7,8	≤ 9	2	n^2	$\leq n^2 + n - 2r + e - 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
1	$\leq \lfloor (n+3)/2 \rfloor$	3	n^2	$\leq n^2 + n + k_2 + H(f(x)) - 8$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
2,3,4	≤ 5	3	n^2	$\leq n^2 + 2n - 2r + e - 3$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$

※ 만약 $H(h(x)) = 2$ 인 경우, $h(x) = x^{k_1} + 1$ ($k_1 > 0$).

만약 $H(h(x)) = 3$ 인 경우, $h(x) = x^{k_1} + x^{k_2} + 1$ ($k_1 > k_2 > 0$).

(72) 발명자

홍석희

서울특별시 도봉구 도봉동 한신아파트 103동 107호

김현민

서울특별시 성북구 길음2동 대림아파트 414동 1801호

특허청구의 범위

청구항 1

유한체(finite field)상에서의 비트-병렬 다항식 기저 곱셈을 위한 기약다항식(irreducible polynomial) $f(x)$

가 $f(x) = \sum_{i=0}^s p(x)x^{i \cdot s+v} + l(x)$, $\deg(l(x)) \leq v$, $\deg(p(x)) < s$ --- 식(1)을 만족하는지 판단하되,

(a)상기 유한체상의 임의의 다항식 $q(x) = x^r + 1 + \sum_{i=1}^{r-1} q_0 x^i$ 에 대해 $x^s = q(x)p(x) + 1$ 을 만족하는 최소 자연수 s 와 상기 $p(x)$ 가 존재하는지 판단하는 단계;

(b)상기 s 와 상기 $p(x)$ 가 존재하는 경우, 상기 $q(x)$ 와 상기 유한체상의 임의의 다항식 $h(x)$ 에 대하여 $x^{n_0+r} = q(x)f_{n_0}(x) + h(x)$ 을 만족하는 최소의 양의 정수 n_0 와 다항식 $f_{n_0}(x) (n_0 = \deg(f_{n_0}(x)))$ 가 존재하는지 판단하는 단계;

(c)상기 n_0 와 상기 $f_{n_0}(x)$ 가 존재하는 경우, 상기 $f(x)$ 의 차수(degree) n 에 대하여 $x^{n+r} = q(x)f(x) + h(x)$ 를 만족하는지 판단하는 단계; 및

(d)상기 $x^{n+r} = q(x)f(x) + h(x)$ 을 만족하는 경우, 상기 $f(x)$ 가 상기 식(1)을 만족하는 것으로 간주하고 상기 유한체상의 임의의 두 다항식의 곱셈을 상기 $f(x)$ 를 기약다항식으로 하여 수행하는 단계를 포함하고,

상기 $x^{n+r} = q(x)f(x) + h(x)$ 을 만족하는 상기 $f(x)$ 는 상기 $f_{n_0}(x)$ 자체 또는 $l(x) = f_{n_0}(x)$ 인 반복형 기약다항식인 것을 특징으로 하는 비트-병렬 다항식 기저 곱셈 방법.

청구항 2

제 1 항에 있어서,

상기 r 은 1이고, 상기 $h(x)$ 의 0이 아닌 항(non-zero term)의 개수가 3인 것을 특징으로 하는 비트-병렬 다항식 기저 곱셈 방법.

청구항 3

제 1 항에 있어서,

상기 r 은 2 이상 8 이하이고, 상기 $h(x)$ 의 0이 아닌 항의 개수가 2 또는 3인 것을 특징으로 하는 비트-병렬 다항식 기저 곱셈 방법.

청구항 4

제 1 항 내지 제 3 항 중 어느 한 항의 방법을 컴퓨터에서 실행하기 위한 프로그램을 기록하는 컴퓨터 판독 가능한 기록 매체.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 비트-병렬 다항식 기저 곱셈 방법에 관한 것으로, 보다 상세하게는 유한체(Finite Field)상에서 비트-병렬 다항식 기저 곱셈을 위한 기약다항식(irreducible polynomial)으로서 반복형 기약다항식(RFP: Repeated Form Polynomial)을 이용하는 비트-병렬 다항식 기저 곱셈 방법에 관한 것이다.

배경 기술

[0002] 암호 시스템에 적용되는 유한체에서 주로 사용되는 기약다항식은 All One Polynomial(AOP), Equally Spaced Polynomial(ESP), 삼항 기약다항식(Trinomial), 오항 기약다항식(Pentanomial)이 있다. 유한체의 특정 차수(degree)에서 비트-병렬 다항식 기저 곱셈의 효율성(곱셈 소요시간 및 곱셈에 필요한 게이트 수)을 고려한 기약다항식의 순서는 위와 같다. 그러나 암호 알고리즘에서 사용되는 유한체의 차수는 대부분 소수(prime number)이므로 AOP와 ESP는 거의 사용되지 않으며, 대부분 Trinomial과 Pentanomial이 사용된다.

[0003] 유한체의 차수가 1000 이하인 것들 중 443개의 차수에 대해서는 AOP, ESP, Trinomial을 사용할 수 없고 Pentanomial만 사용할 수 있었는데, Pentanomial을 사용함으로써 야기되는 곱셈의 비효율성을 개선하기 위해 Pentanomial 아닌 세 가지 종류의 기약다항식이 제안된 바 있다(H. Wu, "Bit-Parallel polynomial Basis Multiplier for New Classes of Finite Fields," IEEE Trans. Computers, vol. 57, no. 8). 그러나 제안된 기약다항식은 Pentanomial보다 효율적이지만 상기 443개의 차수 중 10개의 차수에 대해서만 사용할 수 있다. 따라서 나머지 433개의 차수에 대해서는 여전히 비효율적인 Pentanomial을 사용할 수밖에 없다. 특히 이 433개의 차수를 갖는 유한체(다항식)은 암호 시스템에서 많이 활용되는 차수가 많이 포함되어 있는바, 상기와 같은 문제는 결국 암호 시스템의 효율적 구현에 장애가 된다.

발명의 내용

해결 하고자하는 과제

[0004] 본 발명이 해결하고자 하는 과제는 유한체의 차수가 1,000 이하인 차수 중에서 Pentanomial을 사용하는 경우보다 더 효율적이고 아울러 상기 443개의 차수에 사용될 수 있는 반복형 기약다항식을 이용한 비트-병렬 다항식 기저 곱셈 방법을 제공하는 것이다.

과제 해결수단

[0005] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 비트-병렬 다항식 기저 곱셈 방법에 의하면

[0006] 유한체(finite field)상에서의 비트-병렬 다항식 기저 곱셈을 위한 기약다항식 $f(x)$ 가 $f(x) = \sum_{i=0}^s p(x)x^{i \cdot s+v} + l(x)$, $\deg(l(x)) \leq v$, $\deg(p(x)) < s$ 을 만족하는지의 여부로 상기 $f(x)$ 가 반복형 기약다항식임을 판단하여 비트-병렬 다항식 기저 곱셈이 이루어진다. $f(x)$ 가 반복형 기약다항식임을 판단하는 것

은 (a)상기 유한체상의 임의의 다항식 $q(x) = x^r + 1 + \sum_{i=1}^{r-1} q_0 x^i$ 에 대해 $x^s = q(x)p(x) + 1$ 을 만족하는 최소의 양의 정수 s 와 상기 $p(x)$ 가 존재하는지 판단하는 단계; (b)상기 $p(x)$ 가 존재하는 경우, 상기 $q(x)$ 와 상기 유한체상의 임의의 다항식 $h(x)$ 에 대하여 $x^{n_0+r} = q(x)f_{n_0}(x) + h(x)$ 을 만족하는 최소의 양의 정수 n_0 와 다항식 $f_{n_0}(x) (n_0 = \deg(f_{n_0}(x)))$ 가 존재하는지 판단하는 단계; 및 (c)상기 n_0 와 상기 $f_{n_0}(x)$ 가 존재하는 경우, 상기 $f(x)$ 의 차수(degree) n 에 대하여 $x^{n+r} = q(x)f(x) + h(x)$ 를 만족하는지 판단하는 단계를 포함하여 이루어진다.

[0007] 이때, 상기 r 은 1로 하고, 상기 $h(x)$ 의 0이 아닌 항의 개수를 3으로 하는 것이 본 발명의 과제를 해결함에 바람직하다. 또는 상기 r 은 2 이상 8 이하로 하고, 상기 $h(x)$ 의 0이 아닌 항의 개수를 2 또는 3으로 하는 것이 본 발명의 과제를 해결함에 바람직하다.

효과

[0008] 본 발명에서 제안하는 반복형 기약다항식을 이용한 비트-병렬 다항식 기저 곱셈 방법에 의하면, 특히 유한체의 차수가 1000 이하의 차수 중 AOP, ESP, Trinomial을 사용할 수 없는 443개의 차수에 대해서도 모두 적용가능하므로, 이들 443개의 차수에 대해 오항 기약다항식을 적용함으로써 야기되었던 곱셈의 비효율성을 개선시킬 수 있다. 즉, 차수 n 인 반복형 기약다항식을 사용하는 경우 오항 기약다항식을 사용하는 경우보다 대략 n 개의 XOR 게이트 수(공간 복잡도)를 줄일 수 있고, XOR 게이트의 Time Delay를 현저히 줄일 수 있어 비트-병렬 다항식 기저 곱셈 연산에 소요되는 시간을 획기적으로 줄일 수 있다. 따라서 상기 443개의 차수에 오항 기약다항식을 적용함으로써 야기되었던 비트-병렬 다항식 기저 곱셈의 비효율성을 개선시킬 수 있다.

발명의 실시를 위한 구체적인 내용

[0009] 이하, 본 발명이 해결하고자 하는 과제의 해결 방안을 명확하게 하기 위한 발명의 구성을 본 발명의 바람직한 실시예에 근거하여 첨부 도면을 참조하여 상세히 설명하되, 도면의 구성요소들에 참조번호를 부여함에 있어서 동일 구성요소에 대해서는 비록 다른 도면에 있더라도 동일 참조번호를 부여하였으며 당해 도면에 대한 설명 시 필요한 경우 다른 도면의 구성요소를 인용할 수 있음을 미리 밝혀둔다. 아울러 본 발명과 관련된 공지 기능 혹은 구성에 대한 구체적인 설명 그리고 그 이외의 제반 사항이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그 상세한 설명을 생략한다.

[0010] 1. 제안하는 반복형 기약다항식

[0011] 유한체에서 비트-병렬 다항식 기저 곱셈의 효율성을 고려한 기약다항식의 순서는 상기한 바와 같이 ESP, Trinomial, Pentanomial의 순서를 가지며, 유한체의 차수(n, degree)가 정해지면 상기의 순으로 기약다항식을 선택하여 사용하게 된다. 그러나 n이 1000 이하인 경우에는 77개의 n에서만 ESP를 사용할 수 있고 543개의 n에서만 Trinomial을 사용할 수 있으며, ESP와 Trinomial을 같이 사용할 수 있는 경우까지 고려하면 총 557개의 n에서만 ESP 또는 Trinomial을 사용할 수 있으므로 나머지 443개의 n에서는 가장 비효율적인 Pentanomial을 사용하여야 한다. 따라서 본 발명에서는 이러한 문제를 해결하기 위해 Pentanomial을 사용하는 경우보다 효율적인 새로운 기약다항식 즉 반복형 기약다항식을 제안한다.

[0012] 유한체 F_{2^r} 위에서 임의의 두 다항식 $p(x)$ 와 $l(x)$ 에 대하여 $f(x)$ 가 아래의 식(1)을 만족하면, 그 $f(x)$ 는 반복형 기약다항식이다.

[0013]
$$f(x) = \sum_{i=0}^s p(x)x^{i \cdot s + v} + l(x), \quad \deg(l(x)) \leq v, \deg(p(x)) < s \quad \text{--- 식(1)}.$$

[0014] 이 반복형 다항식은 s-ESP를 포함하며, 임의의 다항식 $h(x) (\in F_{2^r})$ 가 주어졌을 때 식 $x^{n+r} = h(x) \bmod f(x)$ 를 만족하는 $f(x)$ 가 RFP임을 보이기 위하여 다음의 정리(theorem)들을 이용한다.

[0015] <Theorem 1> F_{2^r} 위에서 임의의 다항식 $q(x) = x^r + 1 + \sum_{i=1}^{r-1} q_i x^i$ 에 대하여 $x^s = q(x)p(x) + 1$ 을 만족하는 최소의 양의 정수 s와 다항식 $p(x)$ 가 존재할 때, 임의의 양의 정수 m에 대하여 x^m 이 $x^m = q(x)p_m(x) + 1$ 을 만족하면 양의 정수 i에 대하여 $m = i \cdot s$ 이다.

[0016] Proof of Theorem 1) $x^m = q(x)p_m(x) + 1$ 을 만족하는 m의 순서집합(ordered set)을 $\Gamma = \{m_i \mid x^{m_i} = q(x)p_{m_i}(x) + 1, \text{ 만약 } i > j \text{ 이면 } m_i > m_j\}$ 라 하자. 그러면 가정에 의하여 순서집합 Γ 의 최소값(m_0)은 s이다. 다음으로 s보다 큰 순서집합 Γ 의 최소값을 m_1 라 하면 $x^{m_1} = q(x)p_{m_1}(x) + 1$ 이므로 $F_2[x]$ 에서,

[0017]
$$x^{m_1} - x^s = (q(x)p_{m_1}(x) + 1) - (q(x)p(x) + 1)$$

[0018]
$$\Rightarrow x^s(x^{m_1-s} + 1) = q(x)[p_{m_1}(x) + p(x)]$$

이다.

[0019] 이때 가정에 의하여 $q(x)|(x^{m_1-s} + 1)$ 이므로 m_1-s 는 Γ 의 원소이다. m_1 은 s보다 큰 순서집합 Γ 의 최소값이며, $m_1 > m_1 - s = s$ 이므로 $m_1 = 2s$ 이고 이를 일반화하면 $m_i = i \cdot s$ 이다. 즉, 순서집합 Γ 의 원소는 양의 정수 i에 대하여 $m = i \cdot s$ 이다.

[0020] <Theorem 2> 상기 다항식 $q(x)$ 와 임의의 다항식 $h(x)$ 에 대하여 $x^{n_0+r} = q(x)f_{n_0}(x) + h(x)$ 을 만족하는 최소의 양의 정수 n_0 와 다항식 $f_{n_0}(x) (n_0 = \deg(f_{n_0}(x)))$ 가 존재하면, 임의의 n에 대하여 $x^{n+r} = q(x)f(x) + h(x)$ 를 만족

하는 $f(x)$ 는 $f_{n_0}(x)$ 또는 $l(x)=f_{n_0}(x)$ 인 반복형 다항식이다.

[0021] Proof of Theorem 2) $x^{n_i+r}=q(x)f_{n_i}(x)+h(x)$ 를 만족하는 n_i 의 순서집합을 $\Gamma=\{n_i \mid x^{n_i+r}=q(x)f_{n_i}(x)+h(x), \text{ 만약 } i_j > i_g \text{ 이면 } m_{i_j} > m_{i_g}\}$ 라 하자. 가정에 의하여 $x^{n_0+r}=q(x)f_{n_0}(x)+h(x)$ 을 만족하는 최소 양의 정수 n_0 와 다항식 $f_{n_0}(x)$ 가 존재하고, Γ 의 원소인 n_0 보다 큰 최소값을 n_1 이라 하면 $x^{n_1+r}=q(x)f_{n_1}(x)+h(x)$ 이다. 그러면 $x^{n_1+r}+x^{n_0+r}=x^{n_0+r}(x^{n_1-n_0}+1)=q(x)(f_{n_1}(x)-f_{n_0}(x))$ 이고 $q(x)=x^r+1+\sum_{i=1}^{r-1}q_0x^i$ 이므로 $q(x) \mid (x^{n_1-n_0}+1)$ 이다. 따라서 <Theorem 1>에 의하여 n_1-n_0 는 $i \cdot s$ 이며 n_1 이 n_0 보다 큰 최소값이므로 $n_1-n_0=s$ 이다. 실제로

$$x^{n_0+r}=q(x)f_{n_0}(x)+h(x)$$

$$\begin{aligned} \Rightarrow (x^{n_0+r})x^s &= (q(x)f_{n_0}(x)+h(x)) \cdot x^s \\ &= (q(x)f_{n_0}(x)+h(x)) \cdot (q(x)p(x)+1) \\ &= q(x)\{(q(x)f_{n_0}(x)+h(x))p(x)+f_{n_0}(x)\}+h(x) \\ &= q(x)\{x^{n_0+r}p(x)+f_{n_0}(x)\}+h(x) \\ &= q(x)f_1(x)+h(x) \end{aligned}$$

[0022]

[0023] 이고, 순서집합 Γ 의 원소에 대하여 일반화하면 $n_{i+1}-n_i=s$ 이다. 따라서 $x^{n_i+r}=x^{i \cdot s+n_0+r}=q(x)f_{n_i}(x)+h(x)$ 이다. 다음으로 $f_{n_{i+1}}(x)=x^{i \cdot s+n_0+r}p(x)+f_{n_i}(x)$ 이므로 $f_{n_0}(x)$ 와 (n_0+r) 을 각각 $l(x)=f_{n_0}(x)$, $n_0+r=v$ 라 하면 $f_{n_{i+1}}(x)=\left(\sum_{j=0}^i p(x)x^{j \cdot s+v}\right)+l(x)$ 이다. 따라서 <Theorem 1>의 다항식 $q(x)$ 와 임의의 다항식 $h(x) (\in F_2[x])$ 에 대하여, $x^{n+r}=q(x)f(x)+h(x)$ 를 만족하는 임의의 n 이 존재하면 $f(x)$ 는 $f_{n_0}(x)$ 또는 $l(x)=f_{n_0}(x)$ 인 반복형 다항식이다.

[0024]

2. RFP를 이용한 비트-병렬 다항식 기저 곱셈

[0025]

상기 <Theorem 2>로부터 임의의 $q(x)$, $h(x)$ 에 대하여 $x^{n+r}=q(x)f(x)+h(x) \Rightarrow x^{n+r}=h(x) \bmod f(x)$ 를 만족하는 최소 차수의 $f_{n_0}(x)$ 가 주어졌을 때, $x^{n+r}=q(x)f(x)+h(x)$ 를 만족하는 나머지 $f(x)$ 는 모두 $f_{n_0}(x)$ 와 $p(x)$ 를 이용한 반복형 기약다항식임을 알 수 있다. 반복형 기약다항식 $f(x)$ 에 의하여 정의된 F_x^r 의 임의의 두 원소를 $a(a), b(a)$ 라 하고, $f(x)$ 가 <Theorem 2>를 만족한다면 $x^{n+r} \equiv h(x) \bmod f(x)$ 이다. 따라서

[0026]

$$\alpha^{n+i} \bmod f(x) = \begin{cases} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-1-j}, & 0 \leq i \leq r-1 \\ h(x) \alpha^{i-r}, & r \leq i \leq n-2, \deg(h(x))+i-r \geq n \\ \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-1-j}, & r \leq i \leq n-2, \deg(h(x))+i-r < n \end{cases}$$

[0027] 이다.

[0028] 이때, 만약 $\deg(h(x) \alpha^{i-r}) \geq n$ 이면, $h(x) \alpha^{i-r}$ 는 모듈러 감산(modulo reduction)되어야 한다. 따라서 $c(\alpha) = a(\alpha) \cdot b(\alpha)$ 는 $f(x)$ 에 의하여 다음과 같다.

$$\begin{aligned} c(\alpha) &\equiv \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=n}^{2n-2} s_i \alpha^i \bmod f(x) \right) = \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=n}^{n+r-1} s_i \alpha^i + \sum_{i=n+r}^{2n-2} s_i \alpha^i \bmod f(x) \right) \\ &= \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=n}^{n+r-1} s_i \alpha^i + \left(\sum_{i=n+r}^{2n-2} s_i \alpha^{i-n-r} \right) h(x) \bmod f(x) \right) \\ &= \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^{r-1} s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-1-j} \right) + \left\{ \left(\sum_{i=n+r}^{2n-2} s_i \alpha^{i-n-r} \right) h(x) \bmod f(x) \right\} \Rightarrow \text{식(2)}. \end{aligned}$$

[0029]

$$\text{식(2)에서 } \left(\sum_{i=0}^{r-1} s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-1-j} \right) \text{ 항을 식(2.1), } \left\{ \left(\sum_{i=n+r}^{2n-2} s_i \alpha^{i-n-r} \right) h(x) \bmod f(x) \right\} \text{ 항을 식(2.2)로 한다.}$$

[0030]

[0031] 이때, 만약 $\deg(h(x)) \geq r+2$ 이면 식(2.2)은 다시 모듈러 감산되어야 한다. 감산행렬 T에서 식(2.2)과 같이 $h(x)$ 로 표현되는 행을 REDUCTION BLOCK2라 하고 나머지 행을 REDUCTION BLOCK1라고 하면 RFP의 감산행렬 T는 도 1과 같다. $c(\alpha)$ 의 계산에서 REDUCTION BLOCK1의 효율성은 $h(x)$ 의 0이 아닌 비트수에 의존하고 REDUCTION BLOCK2의 효율성은 $h(x)$ 와 r 에 의존한다.

[0032] 본 발명이 제안하는 반복형 기약다항식을 이용하여 식(2)의 간단한 예를 들면 다음과 같다.

[0033] $q(x)$, $h(x) (\in \mathbb{F}_2[x])$ 를 각각 $q(x) = x^3 + x^2 + x + 1$, $h(x) = x^5 + x^2 + 1$ 이라 하면, $x^{4+3} = q(x)(x^4 + x^3 + x^2 + x + 1) + h(x)$ 이므로, < Theorem 2>에 의하여 $f_{n_0}(x) = x^4 + x^3 + x^2 + x + 1$ 이고, $x^n = q(x)f(x) + h(x)$ 를 만족하는 반복형 기약다항식 $f(x)$ 는

$$f(x) = \left(\sum_{i=0}^7 (x+1)x^{i \cdot 4+7} \right) + x^4 + x^3 + x^2 + x + 1 \quad \text{이다.}$$

[0034]

[0035] 이때 $i=1$ 인 $f(x) = x^8 + x^7 + x^4 + x^3 + x^2 + x + 1$ 은 기약다항식이고 이를 이용하여 F_{2^8} 에서 $c(\alpha) = a(\alpha) \cdot b(\alpha)$ 을 계산하면,

$$\begin{aligned} c(\alpha) &\equiv \sum_{i=0}^7 s_i \alpha^i + \left(\sum_{i=8}^{14} s_i \alpha^i \bmod f(x) \right) \\ &= \sum_{i=0}^7 s_i \alpha^i + \left(\sum_{i=0}^2 s_{n+i} \sum_{j=0}^7 t_{i,j} \alpha^{n-1-j} \right) + \left(\sum_{i=11}^{14} s_i \alpha^{i-n-r} \right) h(x) \bmod f(x) \\ &= \sum_{i=0}^7 s_i \alpha^i + \left(\sum_{i=11}^{13} s_i \alpha^{i-11} \right) h(x) + \left(\sum_{i=0}^2 s_{8+i} \sum_{j=0}^7 t_{i,j} \alpha^{7-j} \right) + \left(s_{14} \sum_{j=0}^{n-1} t_{14,j} \alpha^{7-j} \right) \end{aligned}$$

[0036] 이다. 그리고 이때 7×8 곱셈 행렬(감산 행렬) $T = (t_{i,j})_{7 \times 8}$ 는 다음과 같다.

[0036]

$$T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

[0037]

[0038] 3. 효율적인 RFP와 그 비트-병렬 다항식 기저 곱셈의 복잡도

- [0039] 오항 기약다항식 $f(x) = x^n + x^{k_3} + x^{k_2} + x^{k_1} + 1$ 을 이용한 비트-병렬 다항식 기저 곱셈기의 복잡도는 도 2에 제시되어 있다. 도 2를 통해 알 수 있듯이, 본 오항 기약다항식을 이용하는 경우에 비트-병렬 다항식 기저 곱셈기의 AND 게이트 수(#AND)는 모두 n^2 이며, $[k_1=1$ 이고 $k_3-k_2=1]$ 인 경우를 제외하면 비트-병렬 다항식 기저 곱셈을 위한 XOR 게이트 수(#XOR)와 시간 지연(Time Delay, 시간 복잡도(Time Complexity))은 3가지 경우로 구분된다. 따라서 본 발명에서 제안하는 반복형 기약다항식이 오항 기약다항식보다 차수에 따라 시간 또는 공간 복잡도 측면에서 더 효율적이기 위해서 요구되는 XOR 게이트 수(#XOR)와 시간 복잡도(Time Complexity)는 다음의 3 경우(Case 1, Case 2, Case 3)와 같다.
- [0040] Case 1 : $\#XOR \approx n^2 + n$, Time Delay = $T_A + ([\log_2(n)] + 3)T_X$.
- [0041] Case 2 : $\#XOR \approx n^2 + n$, Time Delay = $T_A + ([\log_2(n)] + 4)T_X$.
- [0042] Case 3 : $\#XOR \approx n^2 + 2n$, Time Delay = $T_A + [\log_2(n) + 3]T_X$.
- [0043] T_A 는 AND 게이트의 게이트 지연(gate delay)을, T_X 는 XOR 게이트의 게이트 지연을 의미한다. 이때, 상기 배경기술에서 소개된 세 가지 타입의 RFP는 Case 1에 속한다(이때, $2r+2 < k_1 < (n+3r+1)/2$).
- [0044] 본 발명이 제안하는 RFP를 이용한 비트-병렬 다항식 기저 곱셈 방법에 의한 시간 복잡도 및 공간 복잡도를 제시하면 다음과 같다.
- [0045] $h(x)$ 의 0이 아닌 항(non-zero term)의 개수를 $H(h(x))$ 라 하면, RFP 기반의 비트-병렬 다항식 기저 곱셈의 효율성은 다항식 $q(x)$ 의 차수 r 과 $H(h(x))$ 에 의존한다. 따라서 효율적인 비트-병렬 다항식 기저 곱셈을 위한 r 과 $h(x)$ 에 대하여 고려하여야 한다.
- [0046] 우선 임의의 r 에 대하여 $h(x)=1$ 인 경우를 살펴보면, 이 경우 RFP $f(x) = \sum_{i=0}^r p(x)x^{i \cdot s + v} + l(x)$ 는 <Theorem 2>의 증명에 의해 $f_{n_0}(x)$ 는 <Theorem 1>의 $p(x)$ 가 된다. 따라서 $p(x)=1(x)$ 이고 항상 $f(x) = p(x) \cdot (\sum_{i=0}^r x^{i \cdot s + 1})$ 이다. 결국 $p(x)$ 가 1이 아닌 경우 $f(x)$ 는 기약다항식이 될 수 없으며, $p(x)=1$ 인 경우 $f(x)$ 는 ESP가 된다. 따라서 $p(x)=1$ 인 경우를 제외하면 임의의 r 에 대하여 최소 해밍 웨이트(hamming weight)인 $H(h(x))=2$ 인 경우가 가장 효율적이다.
- [0047] 다음으로 $H(h(x))=2$ 인 경우의 r 을 살펴보면 다음과 같다. $r=1$ 인 경우 <Theorem 2>에서 $q(x)$ 는 가정에 의하여 $q(x) = x+1$ 이다. 따라서 $x^{nr} + h(x) = q(x)f(x)$ 에서 $q(x)f(x)$ 의 해밍 웨이트는 항상 짝수이다. 그러나 $(x^{nr} + h(x))$ 의 해밍 웨이트는 항상 홀수이므로 $r=1$ 과 $H(h(x))=2$ 를 만족하는 $f(x)$ 는 존재할 수 없다. 따라서 $H(h(x))=2$ 인 경우에는 r 이 2 이상인 경우에만 반복형 기약다항식 $f(x)$ 가 존재한다.
- [0048] 그러면 r 과 $H(h(x))$ 의 최대값을 고려해야 하는데, 이에 대해서는 다음과 같다. r 이 9 이상인 경우는 식(2.1)의 계산에서만 $3 \cdot T_X$ 가 소요되어 위에서 정의한 Case를 만족할 수 없다. 또한 $H(h(x))$ 가 4 이상인 경우는 식(2.2)가 다시 모듈러 감산되지 않는다 하여도 $\#XOR \geq n^2 + 3n$ 이 되어 역시 상기 Case를 만족할 수 없다. 따라서 Case 1 내지 Case 3을 모두 만족하는 조건(r 과 $H(h(x))$ 의 범위)은 다음과 같다.
- [0049] $[r=1$ 이고 $H(h(x))=3]$ 또는 $[2 \leq r \leq 8$ 이고 $2 \leq H(h(x)) \leq 3]$.
- [0050] r 과 $H(h(x))$ 에 따른(각 Case에 따른) RFP와 그 비트-병렬 다항식 기저 곱셈기의 공간 및 시간 효율성(공간 및 시간 복잡도)을 상기 각 Case에 대해 살펴보면 다음과 같다. 이하에서, $\sum_{i=u}^v ()$ 에서 $u > v$ 또는 $u < 0$ 인 경우 $\sum_{i=u}^v () = 0$ 이다.
- [0051] 3.1. Case 1인 경우

[0052] 1) $r=2$ 이고 $H(h(x))=2$ 인 경우

[0053] $H(h(x))=2$ 이므로 $h(x) = x^{k_1} + 1$ 이다. 도 3의 (a)의 REDUCTION BLOCK1의 임의의 열에서 1의 개수가 네 개 이하가 되어야 $3 \cdot T_x$ 내에 곱셈 연산이 가능하므로 k_1 의 범위는 $k_1 \leq (n+5)/2$ 이다. 따라서 k_1 이 최대일 때 $(n+5)/2$, 식(2) 내지 식(2.2)는 다음과 같이 전개된다.

$$\begin{aligned} & \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^1 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) + \left\{ \left(\sum_{i=n+2}^{2n-2} s_i \alpha^{-n+i-2} \right) (x^{k_1} + 1) \bmod f(x) \right\} \\ &= \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^1 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) \\ &+ \left\{ \left(\sum_{i=0}^{n-4} s_{n+i+2} \alpha^i \right) + \left(\sum_{i=k_1}^{n-1} s_{n-k_1+i+2} \alpha^i + \sum_{i=0}^1 s_{2n-k_1+i+2} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} + \sum_{i=n+2}^{n+k_1-4} s_{n-k_1+i+2} \alpha^i \right) \bmod f(x) \right\} \\ &= \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=k_1-5}^{n-4} s_{n+i+2} \alpha^i \right) \Rightarrow \text{식(3.1)} \end{aligned}$$

$$+ \left(\sum_{i=0}^1 (s_{n+i} + s_{2n-k_1+i+2}) \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) \Rightarrow \text{식(3.2)}.$$

$$+ \left\{ (1 + \alpha^{k_1}) \left(\sum_{i=0}^{k_1-6} (s_{n+i+2} + s_{2n-k_1+i+4}) \alpha^i \right) + \left(\sum_{i=2k_1-5}^{n-1} s_{n-k_1+i+2} \alpha^i \right) \right\} \Rightarrow \text{식(3.3)}$$

[0054]

[0055] 따라서 모듈러 감산에 소요되는 XOR 게이트 수(공간 복잡도)는 식(3.1)에서 $(n-k_1+2)$ 개, 식(3.2)에서 2개, 식(3.3)에서 (k_1-3) 개, 식(3.1)+식(3.3)에서 $(n-3)$ 개, (식(3.1)+식(3.3))+식(3.2)에서 최대 n 개이므로 전체 $(3n-2)$ 개이다.

[0056] 시간 복잡도의 경우 식(3.1)+식(3.3) 계산에서 $2 \cdot T_x$ 가 필요하고, 식(3.2)의 $(s_{n+i} + s_{2n-k_1+i+2})$ 에서 $1 \cdot T_x$ 와 $t_{0,j} + t_{1,j}$ 에서 $1 \cdot T_x$ 가 소요되므로 $2 \cdot T_x$ 가 소요된다. 따라서 모듈러 감산에서 최대 $(3n-5)$ 개의 XOR 게이트와 $3 \cdot T_x$ 가 추가적으로 필요하다.

[0057] 2) $r=3$ 이고 $H(h(x))=2$ 인 경우

[0058] 도 3의 (a)의 REDUCTION BLOCK1의 임의의 열에서 1의 개수가 네 개 이하가 되어야 $3 \cdot T_x$ 내에 곱셈 연산이 가능하므로 k_1 의 범위는 6보다 작다. 따라서 $\delta = \begin{cases} 1, k_1 = 5 \\ 0, k_1 < 5 \end{cases}$ 라 하고, k_1 의 최대값을 5로 가정하면 식(2) 내지 식(2.2)은 다음과 같이 전개된다.

$$\begin{aligned} & \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^2 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) + \left\{ \left(\sum_{i=n+3}^{2n-2} s_i \alpha^{-n+i-2} \right) (x^5 + 1) \bmod f(x) \right\} \\ &= (s_n + \delta \cdot s_{2n-2}) \sum_{j=0}^{n-1} t_{0,j} \alpha^{n-j-1} + \left(\sum_{i=1}^2 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) \Rightarrow \text{식(4.1)} \\ &+ \left\{ \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^{n-5} s_{n+i+3} \alpha^i \right) + \delta \cdot \left(\sum_{i=k_1}^{n+k_1-4} s_{n-k_1+i+3} \alpha^i \right) + (1-\delta) \cdot \left(\sum_{i=k_1}^{n+k_1-5} s_{n-k_1+i+3} \alpha^i \right) \right\} \end{aligned}$$

$\Rightarrow$ 식(4.2)

[0059]

[0060] 식(4.1)의 REDUCTION BLOCK1 $\left(\sum_{i=0}^2 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right)$ 의 XOR 게이트 수를 ε 이라 하면 $0 < \varepsilon < 2^r - r$ 이다. 따라

서 식(4.1)은 최대 4개($\delta=1$ 인 경우는 5개)의 XOR 게이트로 계산되며 식(4.2)는 $[n-4+(n-3)]$ ($\delta=1$ 인 경우 $n-4$)개의 XOR 게이트로 계산되고, 식(4.1)+식(4.2)은 n 개의 XOR 게이트로 계산되므로 모듈러 감산에서 최대 $(3n-3)$ 개의 XOR 게이트가 필요하다. 시간 복잡도의 경우 식(4.1)과 식(4.2)은 $2 \cdot T_X$ 내에 계산되므로 $3 \cdot T_X$ 의 시간 지연이 발생한다.

[0061] 3) $r=4$ 이고 $H(h(x))=2$ 인 경우

[0062] 이 경우는 상기한 경우들과 마찬가지로 전개되므로 결과만 정리하면,

[0063] $r=4$, $k_1 < 6$ 인 경우 $\#AND = n^2$, $\#XOR = n^2 + n + \varepsilon - 10$, Time Delay = $T_A + [\log_2(n) + 3]T_X$ 이다. $r > 4$ 인 경우 REDUCTION BLOCK1의 시간 지연만 $3 \cdot T_X$ 가 되어 Case 1에 해당하지 않는다.

[0064] 결론적으로 RFP는 다음의 경우이어야 Case 1을 만족한다.

[0065] a) $r=2$, $H(h(x))=2$, $k_1 < (n+6)/2$.

[0066] b) $r=3$, $H(h(x))=2$, $k_1 < 6$.

[0067] c) $r=4$, $H(h(x))=2$, $k_1 < 6$.

[0068] 3.2. Case 2인 경우

[0069] 1) $r=2$ 이고 $H(h(x))=2$ 인 경우

[0070] $h(x) = x^{k_1} + 1$ 이다. $(n+5)/2 < k_1 \leq (2n+7)/3$ 을 가정하자. 그러면 최대 $k_1 = (2n+7)/3$ 에서 식(2) 내지 식 (2.2)은 다음과 같이 전개된다.

$$\sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^1 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) + \left\{ \left(\sum_{i=n+2}^{2n-2} s_i \alpha^{-n+i-2} \right) (x^{k_1} + 1) \bmod f(x) \right\}$$

$$= \sum_{i=0}^1 (s_{n+i} + \underbrace{(s_{2n-k_1+i+2} + s_{3n-2k_1+i+4})}_{(a)}) \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \Rightarrow \text{식(5.1)}$$

$$+ \left(\sum_{i=0}^{n-1} s_i \alpha^i + \sum_{i=k_1-5}^{n-4} s_{n+i+2} \alpha^i \right) \Rightarrow \text{식(5.2)}$$

$$+ \left(\sum_{i=0}^{-n+2k_1-8} \underbrace{(s_{n+i+2} + \underbrace{(s_{2n-k_1+i+4} + s_{3n-2k_1+i+6}))}_{(b)}) \alpha^i}_{(c)} \right) + \left(\sum_{i=-n+2k_1-7}^{k_1-6} \underbrace{(s_{n+i+2} + s_{2n-k_1+i+4}) \alpha^i}_{(d)} \right)$$

$\Rightarrow$ 식(5.3)

$$+ \left(\sum_{i=k_1}^{-n+3k_1-8} \underbrace{(s_{n-k_1+i+2} + \underbrace{(s_{2n-2k_1+i+4} + s_{3n-3k_1+i+6}))}_{(f)}) \alpha^i}_{(e)} \right) + \left(\sum_{i=-n+3k_1-7}^{n-1} \underbrace{(s_{n-k_1+i+2} + s_{2n-2k_1+i+4}) \alpha^i}_{(g)} \right)$$

$\Rightarrow$ 식(5.4)

[0072] 우선 공간 복잡도(필요한 $\#XOR$)를 살펴본다. 식(5.3)의 (d)의 $(n-k_1+2)$ 개의 XOR 게이트로 (a), (b), (f), (g)는 모두 계산되고 식(5.3)의 (c)의 $(-n+2k_1-7)$ 개의 XOR 게이트로 (e)가 계산된다. 식(5.1)은 3개의 XOR 게이트로 계산되며, 식(5.2)은 $(n-k_1+2)$ 개의 XOR 게이트로 계산된다. 따라서 식(5.3)+식(5.4)은 연산량이 없으며 식(5.1)+식(5.2)은 n 개의 XOR 게이트, (식(5.1)+식(5.2))+(식(5.3)+식(5.4))는 $(n-5)$ 개의 XOR 게이트로 계산되므로 모듈러 감산은 $(3n-5)$ 개의 XOR 게이트로 계산되며 전체적으로 n^2 개의 AND 게이트와 (n^2+n-4) 개의 XOR 게이트가 소요된다.

[0074] 다음으로 시간 복잡도를 살펴본다. 식(5.1)의 계산에서 $3 \cdot Tx$, 식(5.2)의 계산에서 $1 \cdot Tx$, 식(5.3)+식(5.4)의 계산에서 $2 \cdot Tx$ 가 소요되므로, 모듈러 감산에 있어서 $4 \cdot Tx$ 가 추가적으로 필요하다.

[0075] 2) $r=3$, $H(h(x))=2$, $6 \leq k_1 \leq (n+7)/2$ 인 경우

[0076] k_1 이 최대 즉 $k_1=(n+7)/2$ 이면, 식(2) 내지 식(2.2)은 다음과 같이 전개된다.

$$\begin{aligned} & \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^2 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) + \left\{ \left(\sum_{i=n+3}^{2n-2} s_i \alpha^{-n+i-2} \right) (x^{k_1} + 1) \bmod f(x) \right\} \\ &= \sum_{i=0}^2 (s_{n+i} + s_{2n-k_1+3}) \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \Rightarrow \text{식(6.1)} \\ &+ \underbrace{\sum_{i=0}^{n-1} s_i \alpha^i + \sum_{i=k_1-7}^{n-5} s_{n+i+3} \alpha^i + \sum_{i=2k_1-7}^{n-1} s_{n-k_1+i+3} \alpha^i}_{(a)} \\ &+ \underbrace{\sum_{i=0}^{k_1-8} (s_{n+i+3} + s_{2n-k_1+i+5}) \alpha^i}_{(c)} + \underbrace{\sum_{i=k_1}^{2k_1-8} (s_{n-k_1+i+3} + s_{2n-2k_1+i+5}) \alpha^i}_{(d)} . \end{aligned}$$

[0077]

[0078] 우선 공간 복잡도(필요한 #XOR)를 살펴본다. 식(6.1)의 계산에서 $(\varepsilon+3)$ 개의 XOR 게이트 계산, (a)의 계산에서 $(2n-3k_1+10)$ 개의 XOR 게이트 계산 및 (c)의 (k_1-7) 개의 XOR 게이트 계산에 의해 (d)는 계산되었으므로 연산량이 없다. 그리고 (a)+(b)는 $(2k_1-14)$ 개의 XOR 게이트로, 식(6.1)+((a)+(b))는 n 개의 XOR 게이트로 계산되므로 모듈러 감산에서 $(3n+\varepsilon-8)$ 개의 XOR 게이트가 소요된다.

[0079] 다음으로 시간 복잡도를 살펴보면, 식(6.1)에서 $3 \cdot Tx$, (a)+(b)에서 $3 \cdot Tx$ 시간 지연이 발생되므로 모듈러 감산에서 $4 \cdot Tx$ 가 소요된다.

[0080] 3) $r=4$, $H(h(x))=2$, $6 \leq k_1 \leq (n+9)/2$ 인 경우

[0081] k_1 이 최대 즉 $k_1=(n+9)/2$ 이면, 식(2) 내지 식(2.2)은 다음과 같이 전개된다.

$$\begin{aligned} & \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=0}^3 s_{n+i} \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \right) + \left\{ \left(\sum_{i=n+4}^{2n-2} s_i \alpha^{-n+i-2} \right) (x^{k_1} + 1) \bmod f(x) \right\} \\ &= \sum_{i=0}^3 (s_{n+i} + s_{2n-k_1+4}) \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-j-1} \Rightarrow \text{식(7.1)} \\ &+ \underbrace{\sum_{i=0}^{n-1} s_i \alpha^i + \sum_{i=k_1-9}^{n-6} s_{n+i+4} \alpha^i + \sum_{i=2k_1-9}^{n-1} s_{n-k_1+i+4} \alpha^i}_{(a)} \\ &+ \underbrace{\sum_{i=0}^{k_1-10} (s_{n+i+4} + s_{2n-k_1+i+8}) \alpha^i}_{(c)} + \underbrace{\sum_{i=k_1}^{2k_1-10} (s_{n-k_1+i+4} + s_{2n-2k_1+i+8}) \alpha^i}_{(d)} . \end{aligned}$$

[0082]

[0083] 우선 공간 복잡도(필요한 #XOR)를 살펴보면, 식(7.1)의 계산에서 $(\varepsilon+3)$ 개의 XOR 게이트 계산, (a)의 계산에서 $(2n-3k_1+13)$ 개의 XOR 게이트 계산 및 (k_1-7) 개의 XOR 게이트 계산에 의해 (d)는 계산되었으므로 연산량이 없다. 그리고 (a)+(b)에서 $(2k_1-18)$ 개의 XOR 게이트, 식(7.1)+((a)+(b))에서 n 개의 XOR 게이트로 계산되므로 모듈러 감산에서 $(3n+\varepsilon-10)$ 개의 XOR 게이트가 소요된다.

[0084] 다음으로 시간 복잡도를 살펴보면, 식(7.1)에서 $3 \cdot Tx$, (a)+(b)에서 $3 \cdot Tx$, 시간 지연이 발생되므로 모듈러 감

산에서 $4 \cdot T_x$ 가 소요된다.

[0085] 위에서 가정한 경우 보다 k_1 이 큰 경우 REDUCTION BLOCK 1의 계산에만 $4 \cdot T_x$ 가 소요되므로 상기 어느 Case에도 해당하지 않는다. Case 2에 해당하기 위해서는 REDUCTION BLOCK 1의 행의 개수가 8 이하가 되어야 한다. 식

(2.1)의 REDUCTION BLOCK 1에서 행 개수는 r 이며, 식 (2.2)는 $\left\{ \left(\sum_{i=0}^{n-r-2} s_{n+r+i} \alpha^i \right) (x^{k_1} + 1) \bmod f(x) \right\}$ 이므로 $(n-r-2+k_1)-(n-1)+r \leq 8$ 이다. 따라서 $r > 4$ 인 경우 Case 2를 만족하기 위해서는 $r < 9$ 이고 k_1 은 9 이하여야 한다. 이 경우 모듈러 감산의 공간 복잡도는 $2(n-r-2)+\varepsilon+n=(3n-2r+\varepsilon-4)$ 개의 XOR 게이트이고 추가되는 시간 복잡도는 $4 \cdot T_x$ 이다. 따라서 다음 RFP만 Case 2를 만족한다.

[0086] a) $r = 2, H(h(x)) = 2, \lfloor (n+5)/2 < k_1 \leq \lfloor (2n+7)/3 \rfloor$

[0087] b) $r = 3, H(h(x)) = 2, 6 \leq k_1 \leq \lfloor (n+7)/2 \rfloor$

[0088] c) $r = 4, H(h(x)) = 2, 6 \leq k_1 \leq \lfloor (n+9)/2 \rfloor$

[0089] d) $4 < r < 9, H(h(x)) = 2, k_1 \leq 9.$

[0090] 3.3. Case 3인 경우

[0091] 1) $r=1, H(h(x))=3$ 인 경우

[0092] $h(x) = x^{k_1} + x^{k_2} + 1$ 이다($k_1 > k_2 > 1$). 우선 $k_1 \leq (n+3)/2$ 라 가정하면 최대 $k_1 = (n+3)/2$ 에서 식(2) 내지 식(2.2)은 다음과 같이 전개된다.

$$\begin{aligned} & \sum_{i=0}^{n-1} s_i \alpha^i + \left(s_n \sum_{j=0}^{n-1} t_{i,j} \alpha^{n-1-j} \right) + \left\{ \left(\sum_{i=n+1}^{2n-2} s_i \alpha^{-n+i-1} \right) (x^{k_1} + x^{k_2} + 1) \bmod f(x) \right\} = \\ & \sum_{i=0}^{n-1} s_i \alpha^i + \left(s_n + (s_{2n-k_1+1} + s_{2n-k_2+1}) \sum_{j=0}^{n-1} t_{0,j} \alpha^{n-1-j} \right) \\ & + \underbrace{\left(\sum_{i=0}^{n-3} s_{n+i+1} \alpha^i \right)}_{(b1)} + \underbrace{\left(\sum_{i=k_1}^{n-1} s_{n-k_1+i+1} \alpha^i \right)}_{(c1)} + \underbrace{\left(\sum_{i=k_2}^{n-1} s_{n-k_2+i+1} \alpha^i \right)}_{(d)} \\ & + \underbrace{\left(\sum_{i=0}^{k_1-4} s_{2n-k_1+i+2} \alpha^i \right)}_{(b2)} + \underbrace{\left(\sum_{i=k_2}^{k_1+k_2-4} s_{2n-k_1-k_2+i+2} \alpha^i \right)}_{(a1)} + \underbrace{\left(\sum_{i=k_1}^{2k_1-4} s_{2n-2k_1+i+2} \alpha^i \right)}_{(c2)} \Rightarrow \text{식(8)} \\ & + \underbrace{\left(\sum_{i=0}^{k_2-4} s_{2n-k_2+i+2} \alpha^i \right)}_{(e)} + \underbrace{\left(\sum_{i=k_2}^{2k_2-4} s_{2n-2k_2+i+2} \alpha^i \right)}_{(f)} + \underbrace{\left(\sum_{i=k_1}^{k_1+k_2-4} s_{2n-k_1-k_2+i+2} \alpha^i \right)}_{(a2)} \end{aligned}$$

[0093]

[0094] (a1)+(a2)는 $k_1 > k_2$ 이므로,

$$\underbrace{\left(\sum_{i=k_2}^{k_1+k_2-4} s_{2n-k_1-k_2+i+2} \alpha^i \right)}_{(a1)} + \underbrace{\left(\sum_{i=k_1}^{k_1+k_2-4} s_{2n-k_1-k_2+i+2} \alpha^i \right)}_{(a2)} = \sum_{i=k_2}^{k_1-1} s_{2n-k_1-k_2+i+2} \alpha^i \Rightarrow \text{식(9)}$$

[0095]

이고,

[0096] (b1)+(b2)와 (c1)+(c2)는

$$\begin{aligned} & \left(\sum_{i=0}^{n-3} s_{n+i+1} \alpha^i \right) + \left(\sum_{i=0}^{k_1-4} s_{2n-k_1+i+2} \alpha^i \right) = \left(\sum_{i=0}^{k_1-4} (s_{n+i+1} + s_{2n-k_1+i+2}) \alpha^i \right) + \left(\sum_{i=k_1-3}^{n-3} s_{n+i+1} \alpha^i \right) \\ & \left(\sum_{i=k_1}^{n-1} s_{n-k_1+i+1} \alpha^i \right) + \left(\sum_{i=k_1}^{2k_1-4} s_{2n-2k_1+i+2} \alpha^i \right) = \\ & \left(\sum_{i=k_1}^{2k_1-4} (s_{n-k_1+i+1} + s_{2n-2k_1+i+2}) \alpha^i \right) + \left(\sum_{i=2k_1-3}^{n-1} s_{n-k_1+i+1} \alpha^i \right) \end{aligned}$$

[0097]

[0098] 이므로, $(s_{n-k_1+i+1} + s_{2n-2k_1+i+2})$ 는 $(s_{n+i+1} + s_{2n-k_1+i+2})$ 과 중복연산이 되어 연산량이 없다.

[0099] 또한 식(9)와 (d)의 합은

$$\begin{aligned} & \left(\sum_{i=k_2}^{n-1} s_{n-k_2+i+1} \alpha^i \right) + \sum_{i=k_2}^{k_1-1} s_{2n-k_1-k_2+i+2} \alpha^i = \\ & \left(\sum_{i=k_2}^{k_1-1} (s_{n-k_2+i+1} + s_{2n-k_1-k_2+i+2}) \alpha^i \right) + \sum_{i=k_1}^{n-1} s_{n-k_2+i+1} \alpha^i \end{aligned}$$

[0100]

[0101] 이므로 이 또한 $(s_{n+i+1} + s_{2n-k_1+i+2})$ 과 중복연산이므로 연산량이 없다. 마지막으로 (e)= $\delta_1 + \delta_2$ 를 위하여 δ_1 , δ_2 를 정리하면,

$$\begin{aligned} \delta_1 &= \begin{cases} \sum_{i=k_1-3}^{2k_2-4} s_{2n-2k_2+i+2} \alpha^i, & 2k_2-4 \geq k_1-3 \\ 0, & 2k_2-4 < k_1-3 \end{cases} \\ \delta_2 &= \begin{cases} \sum_{i=k_2}^{k_1-4} s_{2n-2k_2+i+2} \alpha^i, & 2k_2-4 \geq k_1-3 \\ \sum_{i=k_2}^{2k_2-4} s_{2n-2k_2+i+2} \alpha^i, & 2k_2-4 < k_1-3 \end{cases} \end{aligned}$$

[0102]

[0103] 이고, 위의 내용을 적용하여 식(8)을 정리하면 다음과 같다.

$$\begin{aligned} & \left((s_n + (s_{2n-k_1+1} + s_{2n-k_2+1})) \sum_{j=0}^{n-1} t_{0,j} \alpha^{n-1-j} \right) \Rightarrow \text{식(10.1)} \\ & + \left\{ \sum_{i=0}^{n-1} s_i \alpha^i + \left(\sum_{i=k_1-3}^{n-3} s_{n+i+1} \alpha^i + \delta_2 + \sum_{i=0}^{k_2-4} s_{2n-k_2+i+2} \alpha^i \right) \right\} \Rightarrow \text{식(10.2)} \\ & + \left\{ \sum_{i=k_1}^{n-1} s_{n-k_2+i+1} \alpha^i + \left(\sum_{i=2k_1-3}^{n-1} s_{n-k_1+i+1} \alpha^i + \delta_1 \right) \right\} + \sum_{i=0}^{k_1-4} (s_{n+i+1} + s_{2n-k_1+i+2}) \alpha^i \end{aligned}$$

$\Rightarrow \text{식(10.3)}$

$$+ \sum_{i=k_1}^{2k_1-4} (s_{n-k_1+i+1} + s_{2n-2k_1+i+2}) \alpha^i + \sum_{i=k_2}^{k_1-1} (s_{n-k_2+i+1} + s_{2n-k_1-k_2+i+2}) \alpha^i \Rightarrow \text{식(10.4)}$$

[0104]

[0105] 식(10.1)의 s_{2n-k_1+1} 와 s_{2n-k_2+1} 는 각각 $\log_2(k_1-1)T_X$, $\log_2(k_2-1)T_X$ 의 시간 지연을 가지므로 가정 $2k_1 < n+4$ 에 의하여 모두 $\log_2(n/2)T_X$ 보다 작은 계산시간을 가지며 ($s_{2n-k_1+1} + s_{2n-k_2+1}$)는 S_{n-1} 과 동시에 계산되므로 식(10.1)의 계산은 $1 \cdot T_X$ 에 계산 가능하다. 또한 식(10.2) 내지 식(10.4)은 모두 $1 \cdot T_X$ 에 계산되므로 식(10.1) 내지 식(10.4)의 시간지연은 $1 \cdot T_X + ([\log_2 n] + 3)T_X$ 이다.

[0106] 다음으로 공간 복잡도를 살펴보면, 식(10.2)+식(10.3)은 $[(n-k_1+1)+(k_2-3)+(n-k_1)+(n-2k_1+3)+(k_2-3)]$ 개의 XOR 게이트와 $2(k_1-3)$ 개의 XOR 게이트의 합이므로 전체 $(3n-2k_1+2k_2-8)$ 개의 XOR 게이트로 연산된다. 또한 식(10.1)은 2개의 XOR 게이트로 연산되며, 식(10.4)은 식(10.3)에서 계산되어 연산량이 없으므로 (식(10.2)+식(10.3))+식(10.4)는 $(2k_1-k_2-3)$ 개의 XOR 게이트로 연산되며, (식(10.2)+식(10.3)+식(10.4))+식(10.1)은 $H(f(x))$ 와 같으므로 비트-병렬 다항식 기저 곱셈 연산을 위한 총 게이트수는 $n^2[\#AND] + ((n)^2 + (n+k_2+H(f(x))-8)[\#XOR])$ 이므로 $\#XOR \approx n^2 + 2n$ 이며 따라서 Case 3이다. 그리고 가정과 달리 $2k_1 \geq (n+4)$ 인 경우에는 식(10.1)의 시간 지연이 $2 \cdot T_X$ 가 되어 상기 어느 Case에도 해당하지 않는다.

[0107] Case 3의 경우 REDUCTION BLOCK2의 계산에서 $2 \cdot T_X$ 가 소요되므로 REDUCTION BLOCK1의 행의 개수가 4보다 작은 $r < 5$ 이다. 따라서 $1 \leq r \leq 4$ 인 경우 식 (2.2)을 고려했을 때 $(n-r-2+k_1)-(n-1)+r \leq 4$ 이므로 $k_1 \leq 5$ 이다. 이 경우 모듈러 감산의 공간 복잡도는 $3(n-r-2)+\varepsilon+n = (4n-2r+\varepsilon-4)$ 개의 XOR 게이트이고 추가되는 시간 복잡도는 $3T_X$ 이다. 따라서 다음 RFP만 Case 3을 만족한다.

[0108] a) $r = 1$, $H(h(x)) = 3$, $2k_1 \leq n+3$.

[0109] b) $1 < r < 5$, $H(h(x)) = 3$, $k_1 \leq 5$.

[0110] 본 발명은 암호 알고리즘에서 사용되는 1000 이하의 유한체 차수 중 AOP, ESP, Trinomial이 존재하지 않는 443개의 차수에서 모두 사용가능한 새로운 기약다항식인 반복형 기약다항식(RFP)을 제안하였으며, 도 2에서 보인 기존의 오항 기약다항식에 비해 차수에 따라 효율적인 세 가지 경우(Case 1, Case 2, Case 3)에서의 RFP를 제시하였다. 세 가지 경우의 기약다항식의 시간 및 공간 복잡도를 도 4에 r에 따라 정리하였다.

[0111] 한편 차수가 1000 이하인 경우의 적용가능한 기약다항식의 개수는 도 5에 제시되어 있다. 기존에 가장 효율적으로 알려진 기약다항식인 ESP와 Trinomial이 없는 443개의 차수 중 Case 1의 경우 181(40.8%)개가 존재하며, Case 2의 경우 232(52.4%)개, Case 3의 경우 443(100%)개 존재한다. 또한 case 3($r=1$)의 경우 $n = 1, 2, 16, 515$ 를 제외한 모든 차수에서 존재한다.

[0112] 본 발명에서 제안하는 RFP는 타원곡선 암호시스템의 표준(ANSI X9.62, IEEE P1363, SEC2)에서 제안되어 사용되는 유한체 중 오항 기약다항식을 사용하는 차수인 $n = 131, 163, 283, 571$ 모두에서 존재하는데(사용가능한데), 이를 살펴보면 다음과 같다.

[0113] $n=131$:

$$f(x) = \left(\sum_{i=67}^{131} x^i \right) + x + 1$$

[0114] $n=163$:

[0115]

$$f(x) = \left(\sum_{i=33}^{163} x^i \right) + \left(\sum_{i=0}^{21} x^i \right), \quad f(x) = \left(\sum_{i=63}^{163} x^i \right) + \left(\sum_{i=0}^5 x^i \right), \quad f(x) = \left(\sum_{i=63}^{163} x^i \right) + \left(\sum_{i=0}^{41} x^i \right), \quad f(x) = \left(\sum_{i=71}^{163} x^i \right) + \left(\sum_{i=0}^1 x^i \right),$$

$$f(x) = \left(\sum_{i=75}^{163} x^i \right) + \left(\sum_{i=0}^{29} x^i \right), \quad f(x) = \left(\sum_{i=77}^{163} x^i \right) + \left(\sum_{i=0}^{21} x^i \right)$$

[0116] $n=283$:

[0117]

$$f(x) = \left(\sum_{i=0}^{31} (x^4 + x^2 + x + 1) x^{(62+7 \cdot i)} \right) + (x^4 + x^2 + x + 1) x^{55} + \left(\sum_{i=0}^6 (x^4 + x^2 + x + 1) x^{(7 \cdot i)} \right)$$

$$f(x) = \left(\sum_{i=15}^{283} x^i \right) + \left(\sum_{i=0}^9 x^i \right), \quad f(x) = \left(\sum_{i=97}^{283} x^i \right) + \left(\sum_{i=0}^1 x^i \right), \quad f(x) = \left(\sum_{i=105}^{283} x^i \right) + \left(\sum_{i=0}^{29} x^i \right), \quad f(x) = \left(\sum_{i=123}^{283} x^i \right) + \left(\sum_{i=0}^5 x^i \right)$$

$$f(x) = \left(\sum_{i=207}^{571} x^i \right) + \left(\sum_{i=0}^{45} x^i \right)$$

n=571:

이다. n=283인 경우, Case 2(r=3)에서 하나 존재하며 나머지는 모두 Case3(r=1)인 경우에서 존재한다. 따라서 제안하는 RFP 기약다항식은 유한체(유한체) 차수 1000 이하에서 모두 존재하며 EPS 또는 Trinomial이 사용될 수 없는 차수에서 오항 기약다항식 보다 효율적이다.

본 방법발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 유무선 네트워크를 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

이제까지 본 발명에 대하여 그 바람직한 실시예를 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다.

그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 균등한 범위 내에 있는 모든 차이점은 본 발명에 포함 된 것으로 해석되어야 할 것이다.

도면의 간단한 설명

도 1은 본 발명에 의한 반복형 기약다항식의 감산행렬 T를 제시한 도면이다.

도 2는 오항 기약다항식 기반의 비트 병렬 다항식 기저 곱셈기의 시간 및 공간 복잡도를 제시한 도면이다.

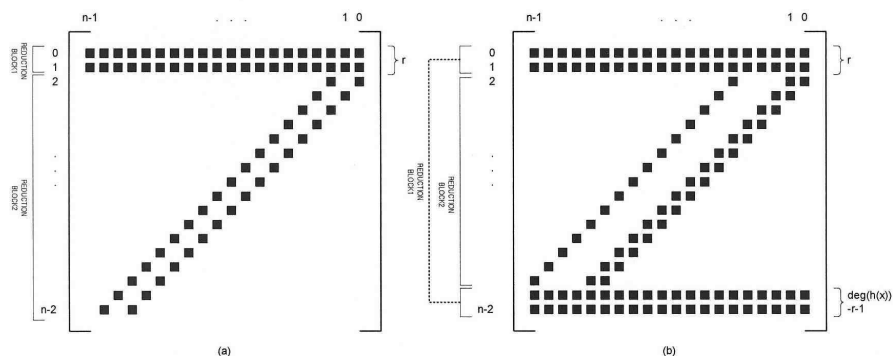
도 3은 본 발명에 의한 반복형 기약다항식의 감산행렬 T의 일례를 제시한 도면으로 $2 \leq r \leq 4$ 인 경우의 감산행렬 T이다.

도 4는 본 발명에 의한 반복형 기약다항식을 이용한 비트-병렬 다항식 기저 곱셈을 하는 곱셈기의 시간 및 공간 복잡도를 제시한 도면이다.

도 5는 차수가 1000 이하인 경우의 적용가능한 기약다항식의 개수를 제시한 도면이다.

도면

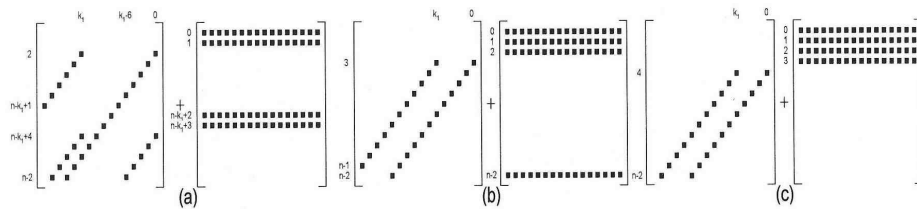
도면1



도면2

Special Case		Space Complexity		Time Complexity
		#AND	#XOR	Time Delay
$f(x) = x^n + x^{k_3} + x^{k_2} + x^{k_1} + 1, 1 < k_1 < k_2 < k_3 \leq n/2$				
$k_1 = 1$	$k_3 - k_2 \neq 1$	n^2	$n^2 + 2n - 3$	$T_A + (3 + \lceil \log_2(n-1) \rceil) T_X$
	$k_3 - k_2 = 1$	n^2	$n^2 + n$	$T_A + (3 + \lceil \log_2(n-1) \rceil) T_X$
$k_1 > 1$	$k_3 - k_2 \neq k_1$	n^2	$n^2 + 2n - 3$	$T_A + (4 + \lceil \log_2(n-1) \rceil) T_X$
	$k_3 - k_2 = k_1$	n^2	$n^2 + n + k_1 - 2$	$T_A + (4 + \lceil \log_2(n-1) \rceil) T_X$
$f(x) = x^n + x^{n-s} + x^{n-2s} + x^{n-3s} + 1$				
$k_1 > 1$	$\frac{n-1}{8} \leq s \leq \frac{n-1}{3}$	n^2	$\leq n^2 + n$	$T_A + (4 + \lceil \log_2(n-1) \rceil) T_X$

도면3



도면4

$x^{n+r} = h(x) \bmod f(x)$			Space Complexity		Time Complexity
r	$\deg(h(x)) = k_1$	$H(h(x))$	#AND	#XOR	Upper Bound
2	$\leq \lfloor (n+5)/2 \rfloor$	2	n^2	$\leq n^2 + n - 6$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
3	≤ 5	2	n^2	$\leq n^2 + n - 2$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
4	≤ 5	2	n^2	$\leq n^2 + n - 1$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
2	$\lfloor (n+5)/2 \rfloor < k_1 \leq \lfloor (2n+7)/3 \rfloor$	2	n^2	$\leq n^2 + n - 4$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
3	$6 \leq k_1 \leq \lfloor (n+7)/2 \rfloor$	2	n^2	$\leq n^2 + n - 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
4	$6 \leq k_1 \leq \lfloor (n+9)/2 \rfloor$	2	n^2	$\leq n^2 + n + 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
5,6,7,8	≤ 9	2	n^2	$\leq n^2 + n - 2r + e - 3$	$1T_A + \lceil \log_2(n) + 4 \rceil T_X$
1	$\leq \lfloor (n+3)/2 \rfloor$	3	n^2	$\leq n^2 + n + k_2 + H(f(x)) - 8$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$
2,3,4	≤ 5	3	n^2	$\leq n^2 + 2n - 2r + e - 3$	$1T_A + \lceil \log_2(n) + 3 \rceil T_X$

* 만약 $H(h(x)) = 2$ 인 경우, $h(x) = x^{k_1} + 1$ ($k_1 > 0$).

만약 $H(h(x)) = 3$ 인 경우, $h(x) = x^{k_1} + x^{k_2} + 1$ ($k_1 > k_2 > 0$)

도면5

기약다항식		기약다항식	1 ≤ n ≤ 1,000 기약다항이 존재하는			ESP 또는 Trinomial이 없는 차수의		
case	r	개수	차수의 개수			개수		
ESP		80	77			557		
Trinomial		1,513	543			(ESP 또는 Trinomial이 존재하는 차수의 개수)		
1	2	504	279	315	1,000	178	181	443
	3	43	40			5		
	4	11	11			3		
2	2	150	128	592		86	232	
	3	482	300			37		
	4	273	192			128		
	5	51	42			18		
	6	29	23			3		
	7	55	42			16		
	8	40	33			7		
3	1	117,858	996	999		442	443	
	2	71	65			23		
	3	70	59			27		
	4	82	68			29		