인수분해 공식과 정규기저를 이용한 $GF(2^m)$ 상의 고속 곱셈 역원 연산 알고리즘

A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ using Factorization Formula and Normal Basis

장 용 희 † 권 용 진 ††
(Yong-Hee Jang) (Yong-Jin Kwon)

요 약 Diffie-Hellman 키분배 시스템과 타원곡선 암호시스템과 같은 공개키 기반 암호시스템은 $GF(2^m)$ 상에서 정의된 연산, 즉 덧셈, 뺄셈, 곱셈 및 곱셈 역원 연산을 기반으로 구축되며, 이들 암호시스템을 효율적으로 구현하기 위해서는 위 연산들을 고속으로 계산하는 것이 중요하다. 그 중에서 곱셈 역원이 가장 time-consuming하여 많은 연구 대상이 되고 있다. Fermat 정리에 의해 $\beta \in GF(2^m)$의 곱셈 역원 β^{-1}은 $\beta^{-1} = \beta^{2^m-2}$이므로 $GF(2^m)$의 임의의 원소에 대해 곱셈 역원을 고속으로 계산하기 위해서는, 2^m-2을 효율적으로 분해하여 곱셈 횟수를 감소시키는 것이 가장 중요하며, 이와 관련된 알고리즘들이 많이 제안되어 왔다. 이 중 Itoh와 Tsujii가 제안한 알고리즘[2]은 정규기저를 사용해서 필요한 곱셈 횟수를 $O(\log m)$까지 감소시켰으며, 또한 이 알고리즘을 향상시킨 몇몇 알고리즘들이 제안되었지만, 분해과정이 복잡하다는 등의 단점이 있다[3,5]. 본 논문에서는 실제 어플리케이션에서 주로 많이 사용되는 $m=2^n$인 경우에, 인수분해 공식 $x^3-y^3=(x-y)(x^2+xy+y^2)$와 정규기저를 이용해서 곱셈 역원을 고속으로 계산하는 알고리즘을 제안한다. 본 논문의 알고리즘은 곱셈 횟수가 Itoh와 Tsujii가 제안한 알고리즘 보다 적으며, 2^m-2의 분해가 기존의 알고리즘 보다 간단하다.

키워드 : 암호시스템, $GF(2^m)$ 체, 곱셈 역원, Fermat 정리, 인수분해 공식, 정규기저

Abstract The public-key cryptosystems such as Diffie-Hellman Key Distribution and Elliptical Curve Cryptosystems are built on the basis of the operations defined in $GF(2^m)$:addition, subtraction, multiplication and multiplicative inversion. It is important that these operations should be computed at high speed in order to implement these cryptosystems efficiently. Among those operations, as being the most time-consuming, multiplicative inversion has become the object of lots of investigation. Fermant's theorem says $\beta^{-1} = \beta^{2^m-2}$, where β^{-1} is the multiplicative inverse of $\beta \in GF(2^m)$. Therefore, to compute the multiplicative inverse of arbitrary elements of $GF(2^m)$, it is most important to reduce the number of times of multiplication by decomposing 2^m-2 efficiently. Among many algorithms relevant to the subject, the algorithm proposed by Itoh and Tsujii[2] has reduced the required number of times of multiplication to $O(\log m)$ by using normal basis. Furthermore, a few papers have presented algorithms improving the Itoh and Tsujii's. However they have some demerits such as complicated decomposition processes[3,5]. In this paper, in the case of 2^m-2, which is mainly used in practical applications, an efficient algorithm is proposed for computing the multiplicative inverse at high speed by using both the factorization formula $x^3-y^3=(x-y)(x^2+xy+y^2)$ and normal basis. The number of times of multiplication of the algorithm is smaller than that of the algorithm

· 본 논문은 과학기술부 · 한국과학재단지정 「한국항공대학교 인터넷정보검색연구센터」의 연구비 및 IDEC의 지원으로 수행되었음.
† 학생회원 : 한국항공대학교 정보통신공학과 대학원
yhjang@mail.hangkong.ac.kr
논 문 접 수 : 2002년 10월 23일
심 사 완 료 : 2003년 3월 4일
†† 정 회 원 : 한국항공대학교 전자 · 정보통신 · 컴퓨터공학부 교수
yjkwon@tikwon.hangkong.ac.kr

proposed by Itoh and Tsujii. Also the algorithm decomposes 2^m-2 more simply than other proposed algorithms.

Key words : Cryptosystems, Finite field $GF(2^m)$, Multiplicative inverse, Fermat's theorem, Factorization Formula, Normal basis

1. 서 론

Galois 체 $GF(2^m)$은 암호시스템과 에러정정코드와 같은 어플리케이션에서 많이 사용된다. 이들 어플리케이션은 $GF(2^m)$ 상에서 정의된 덧셈, 뺄셈, 곱셈 및 곱셈 역원 연산을 기반으로 구축되므로, 이런 어플리케이션을 효율적으로 구현하기 위해서는 이들 연산을 고속으로 계산하는 것이 중요하다. 그리고 이들 어플리케이션 대부분은 큰 수의 m을 갖는 $GF(2^m)$ 상에서 구축되므로, 이들 어플리케이션의 수행 시간은 주로 곱셈 및 곱셈 역원 연산에 좌우된다. 그래서 이들 연산을 고속으로 수행하기 위한 알고리즘을 개발하는 것이 매우 중요하며, 특히 곱셈 역원 연산은 곱셈 연산 보다 시간 복잡도가 더 커서 많은 연구의 대상이 되고 있다 [1,2,3,4,5].

Fermat 정리로부터 $GF(2^m)$의 임의의 원소 β에 대한 곱셈 역원은 $\beta^{-1}=\beta^{2^m-2}$이므로[1,2,3,4], β^{-1}은 β를 $(2^m-2)-1$번 곱셈하면 된다. 그러나 정규기저를 사용해서 β를 표현할 경우, β^2은 cyclic shift로 간단히 계산될 수 있으며, 이것은 곱셈에 의한 계산 보다 매우 고속이다[2,3,4,5]. 따라서 이러한 사실을 곱셈 역원을 계산하는데 이용하면, 필요한 곱셈 연산을 cyclic shift로 치환할 수 있어서 곱셈 역원을 계산하는데 필요한 곱셈 횟수를 상당히 감소시킬 수 있다.

Fermat 정리를 기반으로 하고 $GF(2^m)$의 임의의 원소를 정규기저를 사용해서 표현해서, 곱셈 역원 연산을 계산하는데 필요한 곱셈 횟수를 감소시키는 알고리즘이 많이 제안되어 왔다. 이들 중 Itoh와 Tsujii가 제안한 알고리즘은 필요한 곱셈 횟수를 $O(\log m)$까지 감소시켰으며[2], Chang 등은 $m-1$을 두 개의 인수로 분해하여 몇몇 m에 대해서 Itoh와 Tsujii의 알고리즘을 향상시켰다[3]. 그러나 Chang 등이 제안한 알고리즘은 $m-1$이 소수이면 적용할 수 없고, 인수분해를 어떻게 하냐에 따라서 곱셈 횟수가 차이가 나는 단점이 있다. 그래서 최근에 Takagi 등은 Chang 등의 알고리즘을 보완해서 $m-1$이 소수이어도 적용할 수 있는 새로운 알고리즘을 제안하였지만, 곱셈 횟수를 최소로 하는 $m-1$에 대한 최적 분해를 미리 exhaustive search로 찾아야 하는 단점이 있다[5].

본 논문은 Fermat 정리를 기반으로 하고, 인수분해 공식 $x^3-y^3=(x-y)(x^2+xy+y^2)$와 $GF(2^m)$에서 정규기저를 이용해서 곱셈 역원을 고속으로 계산하는 새로운 알고리즘을 제안한다. 본 논문의 알고리즘은 Itoh와 Tsujii가 제안한 알고리즘을 내부적으로 이용하며, 2^m-2을 복잡하게 분해한다든지 하는 절차 없이, $m-1$에 대한 특성을 이용해 간단한 분해 절차에 의해 필요한 곱셈 횟수를 감소시킨다.

다음 장에서 정규기저를 사용해서 $GF(2^m)$의 임의의 원소에 대한 곱셈 역원을 계산하는 알고리즘을 소개한다. 3장에서는 곱셈 역원을 계산하는 지금까지의 알고리즘에 대해서 요약하고, 4장에서 본 논문에서 제안한 알고리즘을 설명한다. 마지막으로 5장에서 결론을 맺는다.

2. 정규기저를 이용한 곱셈 역원

Galois 체 $GF(2^m)$의 임의의 원소 β는 $GF(2)$ 상에서 정규기저(Normal Basis), α^{2^0}, α^{2^1}, $\cdots$, $\alpha^{2^{m-1}}$ ($\alpha\in GF(2^m)$)를 사용해서 아래와 같이 표현할 수 있다.

$$\beta=\beta_0\alpha^{2^0}+\beta_1\alpha^{2^1}+\cdots+\beta_{m-1}\alpha^{2^{m-1}},\quad \beta_i\in GF(2)$$

또한 위 표현을 이용해서 β는 벡터,

$(\beta_0,\beta_1,\cdots,\beta_{m-1})$으로도 표현할 수 있다.

Fermat 정리로부터 $GF(2^m)$의 임의의 원소 β에 대해서 $\beta^{2^m}=\beta$이고, 곱셈에 대한 역원 β^{-1}은 $\beta^{-1}=\beta^{2^m-2}$이다.

$GF(2^m)$의 임의의 원소 β와 γ에 대해서

$(\beta+\gamma)^2=\beta^2+\gamma^2$ 이므로,

$\beta=\beta_0\alpha^{2^0}+\beta_1\alpha^{2^1}+\cdots+\beta_{m-1}\alpha^{2^{m-1}}$일 때

($\beta_i\in GF(2)$), β^2은

$$\begin{aligned}\beta^2&=(\beta_0\alpha^{2^0}+\beta_1\alpha^{2^1}+\cdots+\beta_{m-1}\alpha^{2^{m-1}})^2\\&=\beta_0\alpha^{2^1}+\beta_1\alpha^{2^2}+\cdots+\beta_{m-1}\alpha^{2^m}\\&=\beta_{m-1}\alpha^{2^0}+\beta_0\alpha^{2^1}+\cdots+\beta_{m-2}\alpha^{2^{m-1}}\end{aligned}$$

이다. β^2을 벡터 표현으로 바꾸면

$(\beta_{m-1},\beta_0,\beta_1,\cdots,\beta_{m-2})$이므로, β의 제곱(squaring)

은 β의 벡터 표현의 1-bit cyclic right shift로 간단히 계산된다. 그리고 β^{2^i}는 (i mod m)-bit cyclic right shift로 계산된다.

$GF(2^m)$의 임의의 원소 β의 곱셈 역원은 $\beta^{-1}=\beta^{2^m-2}$이므로 β의 곱셈 역원을 구하기 위해서는 β를 $(2^m-2)-1$번 곱해야 한다. 그러나 β^{2^m-2}를 β^{2^i}가 포함된 형태로 분해하면, β^{2^i}은 (i mod m)-bit cyclic right shift로 계산하고 각 β^{2^i}끼리 곱셈을 계산하면 되므로 그만큼 곱셈 횟수를 줄일 수 있다.

그래서 정규기저를 이용한 $GF(2^m)$의 임의의 원소에 대한 곱셈 역원을 계산하는 문제는 2^m-2를 어떻게 분해하느냐에 따라 곱셈 횟수가 결정되므로 2^m-2의 분해가 핵심이다. 다음 장에서 2^m-2를 분해하는 이전의 알고리즘에 대해서 설명한다.

3. 기존 알고리즘

$2^m-2=2^1+2^2+\cdots+2^{m-1}$이기 때문에,

$$\beta^{-1}=\beta^{2^m-2}=\beta^{2^1}\times\beta^{2^2}\times\cdots\times\beta^{2^{m-1}}$$

이다. 그래서 β^{2^m-2}은 제곱과 곱셈을 반복 적용하여 계산할 수 있다. 이 알고리즘은 Wang 등이 제안한 것으로서 $m-2$번의 곱셈과 $m-1$번의 제곱을 필요로 한다[1].

Itoh와 Tsujii는 $m-1$을 q-bit의 이진표현 $[1m_{q-2}\cdots m_1m_0]_2$으로 표현하고 아래와 같은 방법을 기반으로 해서 필요한 곱셈 횟수를 $O(\log m)$까지 감소시켰다[2,5].

$m-1=2^{q-1}+m_{q-2}2^{q-2}+\cdots+m_12^1+m_02^0$ 이므로,

$$\begin{aligned}2^{m-1}-1 &=(2^{2^{q-1}}-1)2^{[m_{q-2}\cdots m_1m_0]_2}+2^{[m_{q-2}\cdots m_1m_0]_2}-1\\ &=(1+2^{2^{q-2}})\cdots(1+2^{2^1})(1+2^{2^0})2^{[m_{q-2}\cdots m_1m_0]_2}\\ &\quad+2^{[m_{q-2}\cdots m_1m_0]_2}-1\end{aligned}$$

이고, 여기서 $2^{[m_{q-2}\cdots m_1m_0]_2}=2^{m_{q-2}2^{q-2}+\cdots+m_12^1+m_02^0}$이다. 더 나아가서

$$\begin{aligned}2^{[m_{q-2}\cdots m_1m_0]_2}-1 &=m_{q-2}(2^{2^{q-2}}-1)2^{[m_{q-3}\cdots m_1m_0]_2}\\ &\quad+2^{[m_{q-3}\cdots m_1m_0]_2}-1\\ &=m_{q-2}(1+2^{2^{q-3}})\cdots(1+2^{2^0})2^{[m_{q-3}\cdots m_1m_0]_2}\\ &\quad+2^{[m_{q-3}\cdots m_1m_0]_2}-1\end{aligned}$$

이다. 그래서

$$\begin{aligned}2^{m-1}-1= &((1+2^{2^{q-2}})2^{m_{q-2}2^{q-2}}+m_{q-2})(1+2^{2^{q-3}})\\ &\cdots(1+2^{2^1})(1+2^{2^0})2^{[m_{q-3}\cdots m_1m_0]_2}\\ &+2^{[m_{q-3}\cdots m_1m_0]_2}-1\end{aligned}$$

이며, 위의 감소 절차를 반복 적용하면,

$$\begin{aligned}2^{m-1}-1= &(((\cdots(((1+2^{2^{q-2}})2^{m_{q-2}2^{2^{q-2}}}+m_{q-2})\\ &(1+2^{2^{q-3}})2^{m_{q-3}2^{2^{q-3}}}+m_{q-3})\cdots)(1+2^{2^2})2^{m_22^2}+m_2)\\ &(1+2^{2^1})2^{m_12^1}+m_1)(1+2^{2^0})2^{m_02^0}+m_0\end{aligned}$$

이 된다. 그래서 곱셈 역원

$$\begin{aligned}\beta^{-1} &=\beta^{2^m-2}=(\beta^{2^{m-1}-1})^2\\ &=((((\cdots((\beta^{(1+2^{2^{q-2}})2^{m_{q-2}2^{2^{q-2}}}}\times\beta^{m_{q-2}})^{(1+2^{2^{q-3}})2^{m_{q-3}2^{2^{q-3}}}}\\ &\quad\times\beta^{m_{q-3}})\cdots)^{(1+2^{2^2})2^{m_22^2}}\times\beta^{m_2})^{(1+2^{2^1})2^{m_12^1}}\\ &\quad\times\beta^{m_1})^{(1+2^{2^0})2^{m_02^0}}\times\beta^{m_0})^2\end{aligned}$$

이 된다. 이 알고리즘은 $GF(2^m)$의 임의의 원소에 대한 곱셈 역원을 계산하는데 $l(m-1)+w(m-1)-2$ 번의 곱셈과 $l(m-1)+w(m-1)-1$(multiple-bit)번의 cyclic shift를 필요로 한다. 여기서 $l(m-1)$은 $m-1$을 이진표현 하는데 필요한 bit의 개수이며, $w(m-1)$은 $m-1$의 이진표현에서 1의 개수, 즉 Hamming weight를 나타낸다.

Chang 등은 Itoh와 Tsujii가 제안한 알고리즘을 향상시켰으며, 몇몇 m에 대해서 곱셈 횟수가 더 감소됨을 보였다. 이 알고리즘은 $m-1$을 $m-1=s\times t$로 인수 분해하여 곱셈 역원을 구한다[3,5].

Chang 등의 알고리즘은 $(l(s)+w(s)-2)+(l(t)+w(t)-2)$번의 곱셈과 $(l(s)+w(s)-1)+(l(t)+w(t)-2)$번의 cyclic shift를 필요로 한다. 이 알고리즘을 Itoh와 Tsujii의 알고리즘과 비교해 볼 때, 이 알고리즘의 곱셈 횟수는 몇몇 m에 대해서 감소된다. 그러나 이 알고리즘의 곱셈 횟수는 $m-1$이 2개 이상의 인수를 가지고 있을 때에는, 인수분해 방법에 따라 그 곱셈 횟수가 달라 질 수 있으며, 또한 $m-1$이 소수이면 적용될 수 없는 단점이 있다.

Chang 등이 제안한 알고리즘은 효율적이지만, $m-1$이 소수가 되는 m에 대해서는 적용할 수 없다. 예를 들어 $m=2^n$일 때, $n=5,7,13,19,\cdots$인 경우에는 이 알고리즘을 사용할 수 없다[5].

Takagi 등이 제안한 알고리즘은 이러한 m에 대해서도 적용할 수 있는 알고리즘으로, 그 원리는 다음과 같다[5].

$$\begin{aligned}2^m-2 &=2^{m-1}+2^{m-1}-2\\ &=2^{m-1}+2^{m-2}+\cdots+2^{m-h}+2^{m-h}-2\end{aligned}$$

이므로, β의 곱셈 역원은

$$\beta^{-1}=\beta^{2^m-2}=\beta^{2^{m-1}}\times\beta^{2^{m-2}}\times\cdots\times\beta^{2^{m-h}}\times\beta^{2^{m-h}-2}$$

이다. $\beta^{2^{m-1}}$는 i-bit cyclic left shift에 의해서 계산할 수 있다. 그래서 β^{-1}은 $\beta^{2^{m-h}-2}$와 h번의 곱셈으로부터 계산할 수 있다. $\beta^{2^{m-h}-2}$는 m을 $m-h$로 치환하면 Itoh와 Tsujii 및 Chang 등의 알고리즘에 의해서 계

산할 수 있다.

예를 들어 $m=2^n=128$이면, $m-1=127$이므로 Chang 등의 알고리즘으로는 계산할 수 없다. 그래서 $m-1=127$을 $18\times7+1$로 분해하면

$$2^{m-1}-1=2^{127}-1=2^{18\times7+1}-1=2^{18\times7}+2^{18\times7}-1$$

이 된다. 여기서 $\beta^{2^{18\times7}-1}$을 Chang 등의 알고리즘을 이용해서 계산하면 9번의 곱셈을 필요로 한다. 따라서 $\beta^{2^{128}-2}$는 10번의 곱셈으로 계산될 수 있다. 그러나 Itoh와 Tsujii의 알고리즘은 12번의 곱셈을 필요로 한다.

Takagi 등의 알고리즘은 지금까지의 알고리즘 중에서 곱셈 횟수가 가장 적다. 그러나 이 알고리즘은 m이 주어졌을 때, exhaustive search로 $m-1$에 대한 최적 분해를 우선 찾아야 한다.

4. 인수분해 공식과 정규기저를 이용한 새로운 알고리즘

대부분의 실용적인 어플리케이션에서, m은 주로 2의 거듭제곱을 많이 사용한다[5]. 본 논문에서는 $m=2^n$ 일 때, 2^m-2을 분해하는 새로운 알고리즘을 제안한다.

$m=2^n$이면, $m-1=2^n-1$ 이다. n이 짝수일 때, 2^n-1을 이진표현으로 변환하면 계수가 모두 1이고 1의 개수가 짝수인 n개이다. 예를 들어, $n=6$이면 $2^6-1=63=(111111)_2$ 이다. 그러나 n이 홀수일 때, 2^n-1을 이진표현으로 변환할 경우, 계수는 모두 1이지만 1의 개수는 홀수이다. 예를 들어, $n=7$이면 $2^7-1=127=(1111111)_2$ 이다.

우선 n이 짝수인 6, 즉 $m=2^6$인 경우를 예를 들어 $2^{2^6}-2$를 분해하는 방법에 대해서 설명해 보자. $n=6$이면 $m-1=2^6-1=63=(111111)_2$ 이므로,

$(111111_2)=3(4^2+4^1+4^0)$이다. 그래서 이것과 인수분해 공식 $x^3-y^3=(x-y)(x^2+xy+y^2)$을 분해하는데 이용하면

$$\begin{aligned}2^{m-1}-1 &= 2^{2^6-1}-1\\ &= 2^{(111111)_2}-1\\ &= 2^{3(4^2+4^1+4^0)}-1\\ &= (2^{4^2+4^1+4^0})^3-1^3\\ &= (2^{4^2+4^1+4^0}-1)(2^{(4^2+4^1+4^0)2}+2^{4^2+4^1+4^0}+1)\end{aligned}$$

가 되고, 따라서 β의 곱셈 역원은

$$\begin{aligned}\beta^{-1} &= \beta^{2^{2^6}-2}=(\beta^{2^{2^6-1}-1})^2=(\beta^{2^{63}-1})^2\\ &= (\beta^{(2^{4^2+4^1+4^0}-1)(2^{(4^2+4^1+4^0)2}+2^{4^2+4^1+4^0}+1)})^2\\ &= (\beta^{(2^{63/3}-1)(2^{(63/3)2}+2^{63/3}+1)})^2\\ &= (\beta^{(2^{21}-1)(2^{21\times2}+2^{21}+1)})^2\end{aligned}$$

이 된다. 여기서 $\beta^{2^{21}-1}$은 Itoh와 Tsujii의 알고리즘을 이용해서 6번의 곱셈으로 계산할 수 있다. 그러므로 $\beta^{2^{2^6}-2}$ 는 8=6+2 번의 곱셈 횟수를 필요로 한다. 이것은 Itoh와 Tsujii의 알고리즘만을 사용할 경우인 10번의 곱셈 횟수 보다 적다.

다음으로 n이 홀수인 7인 경우에 대해서 살펴보자. $n=7$이면 $m-1=2^7-1$이므로

$$2^{m-1}-1=2^{2^7-1}-1=2^{127}-1=2(2^{63}-1)(2^{63}+1)+1$$

이다. 따라서 β의 곱셈 역원, β^{-1}은

$$\begin{aligned}\beta^{-1} &= \beta^{2^{2^7}-2}\\ &= (\beta^{2^{2^7-1}-1})^2\\ &= (\beta^{2^{127}-1})^2\\ &= (\beta^{2(2^{63}-1)(2^{63}+1)+1})^2\end{aligned}$$

이다. 여기서 $\beta^{2^{63}-1}$은 위에서 설명한 대로 8번의 곱셈으로 계산된다. 그래서 $\beta^{2^{128}-2}$은 10=8+1+1 번의 곱셈으로 계산 가능하다. 이것은 Itoh와 Tsujii 의 알고리즘만을 사용할 경우인 12번의 곱셈 횟수 보다 적다.

위의 내용을 바탕으로 n이 $n=2k$(k는 양의 정수)인 경우와 $n=2k+1$(k는 양의 정수)에 대해서, $m=2^n$일 때, 2^m-2을 분해하는 방법을 일반화시키면 다음과 같다.

· $n=2k$($m=2^{2k}$)인 경우

$$\begin{aligned}\beta^{-1} &= \beta^{2^m-2}\\ &= \beta^{2^{2^{2k}}-2}\\ &= (\beta^{2^{2^{2k}-1}-1})^2\\ &= (\beta^{2^{3(4^{k-1}+4^{k-2}+\cdots+4+4^0)}-1})^2\\ &= (\beta^{(2^{4^{k-1}+4^{k-2}+\cdots+4+4^0}-1)^3-1^3})^2\\ &= (\beta^{(2^{4^{k-1}+4^{k-2}+\cdots+4+4^0}-1)(2^{(4^{k-1}+4^{k-2}+\cdots+4+4^0)\times2}+2^{(4^{k-1}+4^{k-2}+\cdots+4+4^0)}+1)})^2\\ &= (\beta^{(2^{\frac{2^{2k}-1}{3}}-1)(2^{\frac{2^{2k}-1}{3}\times2}+2^{\frac{2^{2k}-1}{3}}+1)})^2\\ &= (\beta^{(2^{\frac{m-1}{3}}-1)(2^{\frac{m-1}{3}\times2}+2^{\frac{m-1}{3}}+1)})^2\end{aligned}$$

$$\therefore\ \beta^{-1}=(\beta^{(2^{\frac{m-1}{3}}-1)(2^{\frac{m-1}{3}\times2}+2^{\frac{m-1}{3}}+1)})^2$$

· $n=2k+1$($m=2^{2k+1}$)인 경우

$$\begin{aligned}\beta^{-1} &= \beta^{2^m-2}\\ &= \beta^{2^{2^{2k+1}}-2}\\ &= (\beta^{2^{2^{2k+1}-1}-1})^2\\ &= (\beta^{2(2^{2^{2k}-1}-1)(2^{2^{2k}-1}+1)+1})^2\\ &= (\beta^{2(2^{\frac{m}{2}-1}-1)(2^{\frac{m}{2}-1}+1)+1})^2\end{aligned}$$

$$\therefore\ \beta^{-1}=(\beta^{2(2^{\frac{m}{2}-1}-1)(2^{\frac{m}{2}-1}+1)+1})^2$$

위 분해 방법을 바탕으로 본 논문에서는 아래와 같은 알고리즘을 제안한다.

Algorithm

function IT_Algo(β, s) /* calculating β^{2^s-2} using Itoh & Tsujii's Algorithm */

{

```
    /* Assume s=[1m_{q-2}m_{q-3}...m_1m_0]_2 */
    γ := β;
    for i:=q-2 to 0 do
    {
            γ := γ × γ^{2^i};
            if m_i=1 then γ := γ^{2^i} × β;
    }

    return γ;
}

function Even(β, t)   /* calculating β^{2^t-1}, only
when t=2^k-1(k is an positive integer) */
{
    ρ := IT_Algo(β, t/3);
    δ := ρ^{2^{t/3+2}} × ρ^{2^{t/3}} × ρ;

    return δ;
}

main
{
    /* Assume n= log_2 m */

    if n%2=0 then
    {
            ρ := Even(β, m-1);
            β^{-1} := ρ^2;
    }
    else
    {
            α := Even(β, m/2-1);
            ζ := α^2;
            ρ := ζ^{2^{m/2-1}} × ζ × β;
            β^{-1} := ρ^2;
    }
}
```

위 알고리즘으로부터 $n=2k$일 때, 곱셈 역원을 계산하는 필요한 곱셈 횟수는 $l(\frac{2^{2k}-1}{3})+w(\frac{2^{2k}-1}{3})-2+2=l(\frac{m-1}{3})+w(\frac{m-1}{3})$이다. 그리고 $n=2k+1$일 때, 필요한 곱셈 횟수는 $l(\frac{2^{2k}-1}{3})+w(\frac{2^{2k}-1}{3})+2=l(\frac{m-2}{6})+w(\frac{m-2}{6})+2$이다. 본 논문의 알고리즘과 Itoh와 Tsujii의 알고리즘을 몇몇 $m=2^n$에 대해서 곱셈 횟수를 비교하면 표 1과 같다. 이 표로부터 본 논문의 알고리즘이 곱셈 역원을 계산하는데 필요한 곱셈 횟수가 Itoh와 Tsujii가 제안한 알고리즘 보다 적음을 알 수 있다.

표 4 곱셈 횟수 비교(단, $m=2^n(4\le n\le 16)$)

n	$m=2^n$	$m-1$	곱셈 횟수 (본 논문)	곱셈 횟수 (Itoh와 Tsujii)
4	16	15	5	6
5	32	31	6	8
6	64	63	8	10
7	128	127	10	12
8	256	255	11	14
9	512	511	13	16
10	1024	1023	14	18
11	2048	2047	16	20
12	4096	4095	17	22
13	8192	8191	19	24
14	16384	16383	20	26
15	32768	32767	22	28
16	65536	65535	23	30

5. 결 론

본 논문에서는 실용적으로 중요한 $m=2^n$일 때, Fermat 정리를 기반으로 하고, 인수분해 공식 $x^3-y^3=(x-y)(x^2+xy+y^2)$와 정규기저를 사용해서 $GF(2^m)$의 임의의 원소를 표현할 경우에, 곱셈 역원을 고속으로 계산하는 알고리즘을 제안했다. 본 논문의 알고리즘에서는 Itoh와 Tsujii가 제안한 알고리즘 보다 필요한 곱셈 횟수를 감소시켰으며, 또한 다른 이전의 알고리즘과는 다르게 2^m-2을 복잡하게 분해하는 과정이 필요 없이 규칙적이며 간단하다.

향후 연구과제로는 $GF(p^m)$(p는 홀수인 소수) 및 임의의 m 상에서 곱셈 역원을 고속으로 계산하는 알고리즘을 개발하는 것이다.

참 고 문 헌

[1] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, "VLSI Architecture for Computing Multiplications and Inverses in $GF(2^m)$," *IEEE Trans. on Computers*, vol. 34, no. 8, pp. 709-716, Aug. 1985.

[2] T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Basis," *Information and Computing*, vol. 78, pp. 171-177, 1988.

[3] T. Chang, E. Lu, Y. Lee, Y. Leu, and H. Shyu, "Two Algorithms for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Basis," accepted by *Information Processing Letters*.

[4] L. Gao and G. E. Sobelman, "Improved VLSI Designs for Multiplication and Inversion in $GF(2^m)$ over Normal Basis," *Proceeding of ASIC/SOC Conference 2000*, pp. 97-101.

[5] N. Takagi, J. Yoshiki, and K. Takagi, "A Fast Algorithm for Multiplicative Inversion in $GF(2^m)$ Using Normal Basis," *IEEE Trans. on Computers*, vol. 50, No. 5, pp. 394-398, May 2001.

장 용 희

1996년 2월 한국항공대학교 항공통신정보공학과 졸업. 1998년 2월 한국항공대학교 정보통신공학과 대학원 졸업(공학석사). 1998년 3월~현재 한국항공대학교 정보통신공학과 대학원 박사과정 재학중. 관심분야는 논리회로 설계 및 합성, 정보보호, 암호이론

권 용 진

1986년 2월 한국항공대학교 항공전자공학과 졸업. 1990년 3월 일본쿄토대학 정보공학과 대학원 졸업(공학석사). 1994년 3월 일본쿄토대학 정보공학과 대학원 졸업(공학박사). 1994년 3월~현재 한국항공대학교 전자·정보통신·컴퓨터공학부 부교수. 관심분야는 정보보호, 논리설계 및 합성, 정보검색