Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.3, Jun. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.3.515

224비트 소수체에서 효율적인 모듈러 곱셈*

장 남 수†‡
세종사이버대학교 정보보호학과

Efficient Modular Multiplication for 224-bit Prime Field*

Nam Su Chang†‡
Department of Information Security, Sejong Cyber University

요 약

타원곡선 상수배 연산은 사영좌표계를 기반으로 대부분 모듈러 곱셈으로 계산되므로 모듈러 곱셈의 효율성은 타원곡선암호의 성능에 크게 영향을 미친다. 본 논문에서는 FIPS 186-4의 224비트 소수체에서 효율적인 모듈러 곱셈 방법을 제안한다. 제안하는 방법은 Karatsuba 곱셈과 새로운 모듈러 감산을 수행한다. 제안하는 모듈러 곱셈은 기존방법에 비하여 25%정도 빠르며, 모듈러 감산만 비교하면 기존 방법보다 50% 연산으로 계산이 가능하다.

ABSTRACT

The performance of Elliptic Curves Cryptosystem(ECC) is dominated by the modular multiplication since the elliptic curve scalar multiplication consists of the modular multiplication in projective coordinates. In this paper, we propose a new method that combines the Karatsuba-Ofman multiplication method and a new modular reduction algorithm in order to improve the performance of the modular multiplication for NIST p224 in the FIPS 186-4 standard. The proposed method leads to a running time improvement for computing the modular multiplication about 25% faster than the previous methods. The results also show that the method can reduce the arithmetic complexity by half when compared with traditional implementations on the standpoint of the modular reduction.

Keywords: Elliptic Curve Cryptosystem, Modular Multiplication, Finite Field Arithmetic

I. 서 론

타원곡선암호는 기존의 공개키암호와 비교하였을 때 여러 장점을 가진다. 다른 공개키 암호와 같은 보안강도를 기준으로 비교하였을 때 짧은 키 길이를 가지고 요구되는 계산량도 비교적 적다. 따라서 경량화가 요구되는 환경에서 타원곡선은 강점을 가진다.

타원곡선 암호는 타원곡선 상수배(scalar multiplication) 연산으로 구성된다. 그리고 타원곡선의 상수배 알고리즘은 반복적인 타원곡선 점 덧셈 연산(point-addition)과 두 배 연산(point-doubling)으로 구성되어 있는데, 두 가지 연산은 타원곡선을 정의하는 유한체의 덧셈, 곱셈, 역원 연산으로 분류할 수 있다. 이러한 유한체 연산 중 역원 연산은 다른 연산과 비교하여 상대적으로 계산량이 많기 때문에 상수배 연산은 역원 연산이 필요한 아핀 좌표계(affine coordinate)보다 역원 연산이 없는 사영좌표계(projective coordinate)를 많이 사용한다. 따라서 타원곡선 상수배 알고리즘과 유한체 곱셈이 효율성이 타원곡선암호의 성능을 좌우한다[1-4].

본 논문은 FIPS 186-4에서 제시된 타원곡선 224비트 소수체에서 효율적인 모듈러곱셈 방법을 제

Received(05. 02. 2019), Accepted(05. 16. 2019)
* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2017-0-00380,차세대 인증 기술 개발)
† 주저자, nschang@sjcu.ac.kr
‡ 교신저자, nschang@sjcu.ac.kr(Corresponding author)

안한다. 제안하는 방법은 Karatuba 곱셈과 새로운 모듈러 감산 방법을 수행한다. 제안하는 방법은 기존 방법에 비하여 25%정도 빠르며, 모듈러 감산만 비교하면 기존 방법의 절반으로 계산이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 224비트 소수체에서 기존 모듈러 곱셈 방법에 대하여 기술한다. 3장에서는 제안하는 karatsuba 과 모듈러 감산 방법을 기술한다. 4장에서는 기존 결과와 비교하고 결론을 내린다.

II. 224비트 소수체에서 기존 모듈러 곱셈

FIPS 186-4 타원곡선에서 사용하는 224비트 유한체 소수는 다음과 같다.

$$p224 = 2^{224} - 2^{96} + 1$$

이다. 정수 표현을 위한 워드 크기를 $B = 2^{32}$으로 가정하면 소수체의 원소 a는 다음과 같이 표현된다.

$$a = a_6 B^6 + a_5 B^5 + \cdots + a_1 B + a_0.$$

2.1 기존 소수체 원소의 곱셈

본 소절에서는 기존의 정수 곱셈 방법에 대하여 기술한다. 두 원소 $a,b \in GF(p)$의 모듈러 곱셈을 정수 곱셈과 모듈러 감산으로 구분하면 다음과 같다.

$$a = a_6 B^6 + a_5 B^5 + \cdots + a_1 B + a_0,$$
$$b = b_6 B^6 + b_5 B^5 + \cdots + b_1 B + b_0,$$
$$c = a \cdot b = c_{13} B^{13} + c_{12} B^{12} + \cdots + c_1 B + c_0,$$
$$s = c \bmod p224$$

타원곡선 상수배 연산의 효율성은 모듈러 곱셈 연산이 많은 부분을 차지한다. 따라서 효율적인 연산을 위하여 karatsuba 곱셈을 사용할 수 있다. n개의 워드로 구성된 정수를 곱하는 경우 n^2번의 곱셈 연산을 수행하지만 karatsuba 곱셈을 사용하면 $n^{1.58}$번으로 줄일 수 있다. karatsuba 곱셈을 한번만 적용하는 경우 곱셈은 다음과 같이 계산된다.

$$aH = a_6 B^3 + \cdots + a_3,\ aL = a_2 B^2 + \cdots + a_0,$$
$$bH = b_6 B^3 + \cdots + b_3,\ bL = b_2 B^2 + \cdots + b_0,$$
$$a = aH \cdot B^3 + aL,\ b = bH \cdot B^3 + bL$$
$$c = a \cdot b = (aH \cdot B^3 + aL)(bH \cdot B^3 + bL),$$
$$= aHbHB^6 + (aHbL + aLbH)B^3 + aLbL$$
$$= aHbHB^6 + aLbL + \{aHbH + aLbL - (aH - aL)(bH - bL)\}B^3$$
$$s = c \bmod p224 \quad (1)$$

224비트 소수체에서 식(1)과 같이 karatsuba 곱셈을 사용하면 워드 단위 곱셈을 47번에서 41번으로 줄일 수 있다. 그러나 감소하는 단위 곱셈 연산량에 비해 추가적으로 발생하는 덧셈, 뺄셈이 많고 계산을 위한 데이터 저장 공간도 늘어나는 단점이 있다. RSA와 같이 크기가 비교적 큰 환경에서는 효과적이나 일반적으로 타원곡선과 작은 환경에서는 구현 복잡도와 난이도에 비하여 크게 효과가 없다.

2.2 기존의 모듈러 감산

본 소절에서는 FIPS 186-4 타원곡선에서 사용하는 유한체 소수 $p224$ 기반의 모듈러 감산 방법을 살펴본다. Fig.1은

$$t2^{224} \equiv t(2^{96} - 1) \bmod p224$$

를 이용하여 몇 번의 덧셈, 뺄셈으로 모듈러 감산을 빠르게 수행한다[5-8].

Fig.1을 살펴보면 2번의 덧셈, 2번의 뺄셈과 더불어 추가적인 모듈러 연산을 수행하여 모듈러 감산이 된다. $\bmod p224$ 의 경우 사전 계산된 $p224$의 상수배를 사용하면 최대 두 번의 덧셈 또는 뺄셈으로 계산된다.

Input : $c = (c13, \cdots, c2, c1, c0)$ in base 2^{32}
Output : $s = c \bmod p224$

(1) s1 = (c6, c5, c4, c3, c2, c1, c0),
s2 = (c10, c9, c8, c7,0,0,0),
s3 = (0, c13, c12, c11,0,0,0),
s4 = (c13, c12, c11, c10, c9, c8, c7),
s5 = (0,0,0,0, c13, c12, c11).

(2) Return((s1+s2+s3-s4-s5 mod p224).

Fig. 1. Fast reduction modulo NIST p224

III. 제안하는 모듈로 곱셈 방법

본 절에서는 모듈로 감산의 효율성을 고려하면 유한체 원소의 표현방법을 제안하고, 제안한 표현방법 기반의 모듈로 곱셈에 대하여 기술한다. 제안하는 원소 표현 방법은 $p224 \cdot B^{-3}$을 모듈러로 사용한다. 따라서 유한체의 원소 a의 제안하는 원소 표현 방법은 다음과 같다.

$$a = a_6 B^3 + a_5 B^2 + \cdots + a_1 B^{-2} + a_0 B^{-3},$$
$$a_i \in \{0, 2^{32} - 1\}.$$

a_i의 범위와 표현법이 기존과 같기 때문에 위의 표현 방법으로 연산하는 경우 덧셈, 뺄셈, 곱셈 연산은 기존과 동일하다. 그러나 모듈러 감산의 경우 모듈러 연산하는 항이 기존과 달라진다.

3.1 제안하는 karatsuba 모듈로 곱셈

본 소절에서는 karatsuba 곱셈 방법에 제안하는 원소 표현 방법을 적용하여 효율적인 모듈로 곱셈방법을 제안한다. 먼저 기존 원소 표현을 적용한 식(1)을 살펴보자. karatsuba곱셈과 Fig.1의 모듈러 감산을 같이 살펴보면 $aHbHB^6$와

$$\{aHbH+aLbL-(aH-aL)(bH-bL)\}B^3$$

의 계산에 모듈러 감산해야 하는 항이 포함되어 있다. 따라서 식 (1)과 같이 karatsuba 곱셈을 수행한 결과에 모듈러 감산한다. 그러나 제안하는 원소 표현법을 기반으로 karatsuba 모듈로 곱셈을 수행하면 Fig.2와 같이 한번에 carry 처리를 할 수 있다. 제안하는 방법은 다음과 같다.

$$aH=a_6B^3+\cdots+a_3,\ aL=a_2B^2+\cdots+a_0$$
$$bH=b_6B^3+\cdots+b_3,\ bL=b_2B^2+\cdots+b_0$$
$$c=a\cdot b=(aH+aL\cdot B^{-3})(bH+bL\cdot B^{-3}),$$
$$=aHbH+(aHbL+aLbH)B^{-3}+aLbLB^{-6}$$
$$=aHbH+aLbLB^{-6}$$
$$+\{aHbH+aLbL-(aH-aL)(bH-bL)\}B^{-3}$$
$$s=c \bmod p224\cdot B^{-3}$$

제안하는 원소 표현법으로 위와 같이 계산하면 모듈러 감산해야 하는 항은

$$aHbH, aLbLB^{-6}$$

에만 존재한다. 그리고

$$\{aHbH+aLbL-(aH-aL)(bH-bL)\}B^{-3}$$

은 $(aHbL+aLbH)B^{-3}$과 같으므로 제안하는 원소의 표현 범위 크기의 값 연산이 된다. 따라서 기존에 모듈로 감산 대상인 값과 더하여 carry 처리하지 못했던 문제가 해결된다. 제안하는 방법은 Fig.1과 같이 karatsuba 곱셈에서 부가적으로 발생하는 덧셈과 모듈러 감산에서 발생하는 덧셈을 Fig.2의 (3)과 같이 한 번에 계산할 수 있다.

Fig.2에서 y = (t, y) - x mod p224·B^{-3}

는 식을 풀어보면 $(aHbL+aLbH)B^{-3}+s1+s2$와 같다. 따라서 Fig.2의 (3)에서 음수는 발생하지 않는다. 또한 mod p224·B^{-3} 연산은 사전 계산된 p224·B^{-3}의 상수배를 사용하면 최대 두 번의 뺄셈으로 계산된다.

Input : $a=(a6,\cdots,a0)$, $b=(a6,\cdots,a0)$, base 2^{32}
Output : $s=ab \bmod p224\cdot B^{-3}$

(1)aH=(a6,a5,a4,a3), aL=(a2,a1,a0)
bH=(b6,b5,b4,b3), bL=(b2,b1,b0)

(2)u=aL·bL=(0,0,u5,u4,u3,u2,u1,u0)
v=aH·bH=(v7,v6,v5,v4,v3,v2,v1,v0)
x=(aH-aL)(bH-bL)=(x7,x6,x5,x4,x3,x2,x1,x0)
s1=(0,v3,v2,v1,v0,u5,u4,u3)
s2=(0,v7,v6,v5,v4,u2,u1,u0)
s3=(0,u2,u1,u0,v7,v6,v5,v4)

(3)t=0;
for i from 0 to 7 do
(t,y[i])=s1[i]+u[i]+v[i]+s2[i]
y = (t, y) - x mod p224·B^{-3}

(4) Return(y-s3 mod p224·B^{-3}).

Fig. 2. Proposed karatsuba modulo multiplication

3.2 제안하는 모듈로 감산 방법

본 소절에서는 제안하는 원소 표현 기반의 모듈로 감산에 대하여 기술한다. 제안하는 모듈로 감산방법은 기존과 비교하여 작은 계산량을 가진다. Fig.3과 같이 제안하는 모듈러 감산 방법을 사용하면 기존의 2번의 덧셈과 2번의 뺄셈이 1번의 덧셈과 1번의 뺄셈으로 감소한다. 또한 계산과정에서 필요한 저장 공간도 줄어든다.

Input : $c=(c13,\cdots,c2,c1,c0)$ in base 2^{32}
Output: $s=c \bmod p$

(1) s1=(c9, c8, c7, c6, c5, c4, c3),
s2=(c13, c12, c11, c10, c2, c1, c0),
s3=(c2, c1, c0, c13, c12, c11, c10).

(2) Return((s1+s2-s3) mod p224).

Fig. 3. Proposed fast reduction modulo p224

IV. 비교 및 결론

본 논문에서는 새로운 모듈러 곱셈 방법을 제시하

였다. 제안하는 방법은 새로운 원소 표현을 기반으로 karatuba곱셈과 모듈로 감산을 효율적으로 병합한다. Cortex-M3 75MHz에서 C로 구현한 결과 기존 방법은 모듈로 곱셈에서 3,232 clock cycle이 소요되고 제안하는 방법은 2,418 clock cycle이 소요된다. 또한 모듈로 감산만 비교하는 경우 4번에서 2번으로 덧셈, 뺄셈 연산이 줄어든다.

References

[1] D. E. Knuth, "The Art of Computer Programming," Addison-Wesley Publishing Company, Reading, MA, 1981

[2] H. Cohen, "`A Course in Computational Algebric Number Theory," Springer-Verlag, Berlin, Heidelberg, 1993

[3] American National Standard for Financial Services, "Public Key Cryptography for the financial services industry: ECDSA, X9.62," 1998

[4] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography," Springer, 2004

[5] M. Brown, D. Hankerson, A. Menezes, "Software Implementation of the NIST Elliptic Curves over Prime Fields", Proceedings of CT-RSA 2001, LNCS2020, Springer Verlag, pp.250-265, 2001

[6] N. S. Chang, C. H. Kim, S. Hong, Y. Park, "Efficeint Bit-Parallel Polynomial Basis Multipliers fo All Irreducible Trinomial," *Jonornal of The Korea Institute of information Security & Cryptology*, 19(2), pp. 49-61, Apr. 2009

[7] E. Kasper, "Fast elliptic curve cryptography in OpenSSL," Proceedings of the 2011 international conference on Financial Cryptography and Data Security, Mar, 2011

[8] J. Chung, M. A. Hasan, "Low-weight polynomial form integers for efficient modular multiplication" IEEE Transactions on Computers, 56(1), pp. 44 - 57, Jan. 2007

〈저자소개〉

장 남 수 (Nam Su Chang) 종신회원
2002년 2월: 서울 시립대학교 수학과 이학사
2004년 8월: 고려대학교 정보보호 대학원 공학석사
2010년 2월: 고려대학교 정보경영공학전문대학원 공학박사
2010년 2월~6월: 고려대학교 정보경영공학전문대학원 연구교수
2010년 7월~현재: 세종사이버대학교 정보보호학과 조교수
〈관심분야〉 암호칩 설계 기술, 부채널 공격, 공개키 암호 알고리즘, 공개키 암호 암호분석