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.2019.29.3.511

NIST 소수 P-256에서 효율적인 모듈러 감산 방법*

장 남 수†‡
세종사이버대학교 정보보호학과

Efficient Modular Reduction for NIST Prime P-256*

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

요 약

타원곡선암호시스템(ECC)은 같은 보안강도일 때 상대적으로 작은 키 길이를 가지며, 암호시스템의 효율성은 기존의 공개키 암호시스템과 같이 유한체 연산에 의존한다. 타원곡선 암호시스템의 경우 주로 이진체 또는 소수체에서 고려되며 유한체 연산에서 모듈러 곱셈 연산이 효율성에 가장 큰 영향을 미친다. 본 논문은 NIST P256에서 효율적인 모듈러 감산 방법을 제안한다. 제안하는 방법을 소프트웨어로 구현하면 결과 기존 대비 대략 25% 빨라진다.

ABSTRACT

Elliptic Curves Cryptosystem(ECC) provides the same level of security with relatively small key sizes, as compared to the traditional cryptosystems. The performance of ECC over GF(2m) and GF(p) depends on the efficiency of finite field arithmetic, especially the modular multiplication which is based on the reduction algorithm. In this paper, we propose a new modular reduction algorithm which provides high-speed ECC over NIST prime P-256. Detailed experimental results show that the proposed algorithm is about 25% faster than the previous methods.

Keywords: Elliptic Curve Cryptosystem, Fast Reduction, Finite Field Arithmetic

I. 서 론

타원곡선을 이용한 공개키 암호시스템은 1985년에 Miller와 Koblitz가 제안하였다. 기존의 유한체 기반 암호시스템과 비교하여 계산량, 키 사이즈 등에서 장점을 가진다. 또한, 연산을 수행해 주는 효율적인 알고리즘을 가지고 있어 암호학적 응용이 용이하다. 타원곡선 암호시스템은 기존의 이산대수에서 사용하는 유한체의 곱셈 군을 타원곡선 군으로 대치한 암호시스템으로 주로 전자서명 및 키 공유에 활용한다.

Received(05. 02. 2019), Accepted(05. 16. 2019)
* 본 연구는 고려대 암호기술 특화연구센터(UD170109ED)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다
† 주저자, nschang@sjcu.ac.kr
‡ 교신저자, nschang@sjcu.ac.kr(Corresponding author)

NIST에서 제안하는 유한체 소수는 일반적으로 나눗셈 기반의 모듈러 감산보다 빠르게 동작한다. 그러나 NIST P256 소수의 경우 비교적 복잡도가 높아 효율적으로 구현하지 않으면 256-bit 정수 곱셈 속도보다 느리다. 효율적으로 구현한 경우에도 256-bit 정수 곱셈과 비슷한 성능을 보인다. 따라서 모듈러 감산의 고속화는 타원곡선 상수배 연산의 성능 향상에 많은 영향을 미친다[1-4].

본 논문은 NIST P-256에서 효율적인 모듈러 감산 방법을 제안한다. 새로운 유한체 원소 표현 방법을 제안하고, 제안하는 표현 방법 기반의 모듈러 감산 방법을 제시한다. 제안하는 방법을 Cortex-M3 75MHz에서 테스트한 결과 NIST 표준에서 제안하는 모듈러 감산 방법에 비하여 25%의 성능 향상을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 NIST

P-256의 기존 모듈러 감산 방법에 대하여 기술한다. 3장에서는 제안하는 원소 표현방법과 모듈러 감산 방법을 기술한다. 4장에서는 기존 결과와 비교하고 결론을 내린다.

II. 기존의 NIST P256의 유한체 연산

NIST에서 제안하는 256-bit 소수 p는

$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$$

이다. 32-bit 워드(word)를 기본 단위로 하는 환경을 고려하면, $W = 2^{32}$에 의하여 유한체 $GF(p)$의 원소 a, b는 일반적으로 다음과 같이 표현한다.

$$a = a_7 W^7 + a_6 W^6 + \cdots + a_2 W^2 + a_1 W^1 + a_0,$$
$$b = b_7 W^7 + b_6 W^6 + \cdots + b_2 W^2 + b_1 W^1 + b_0,$$
$$a_i, b_i \in \{0, 2^{32} - 1\}.$$

두 원소 a, b의 곱 c 는 다음과 같은 두 과정에 의하여 계산된다:

■ 정수 곱셈:

$$c = ab = \sum_{i=0}^{15} c_i W^i,\ c_i \in \{0, 2^{32} - 1\}$$

■ p에 의한 모듈러 감산 연산:

$$s \equiv c \bmod p \equiv \sum_{i=0}^{7} s_i \alpha^i,\ s_i \in \{0, 2^{32} - 1\}.$$

모듈러 곱셈 연산의 복잡도는 위의 두 과정에 의하여 결정된다.

2.1 NIST P256의 모듈러 감산

FIPS 186-4에서 제안하는 타원곡선암호의 유한체 소수는 구현 환경을 고려하여 32-bit 또는 64-bit 단위 연산에 효율적인 소수로 선택되었다. 제안하는 소수가 연산관점에서 가장 효율적인 소수라고 단정할 수는 없으나, 32 또는 64비트 단위의 항으로 구성되어 다음과 같이 비교적 빠르게 연산된다.

모듈러 감산은 Fig.1과 같으며 실제로

s1+2(s2+s3)+s4+s5−(s6+s7+s8+s9)

의 연산과 추가적인 모듈러 감산에 의해서 계산된다. 또한, 정수 뺄셈의 경우 연산 시 정수값의 크기를 비교하는 등 추가적인 연산을 필요로 한다.

Input: $c = (c15, c14, c13, \cdots, c2, c1, c0)$, $W = 2^{32}$
$p[i] = i \cdot p$
Output : $s = c \bmod p$

```
(1)s1=(c7, c6, c5, c4, c3, c2, c1, c0),
   s2=(c15, c14, c13, c12, c11,0,0,0),
   s3=(0, c15, c14, c13, c12,0,0,0),
   s4=(c15, c14,0,0,0, c10, c9, c8),
   s5=(c8, c13, c15, c14, c13, c11, c10, c9),
   s6=(c10, c8,0,0,0, c13, c12, c11),
   s7=(c11, c9,0,0, c15, c14, c13, c12),
   s8=(c12,0, c10, c9, c8, c15, c14, c13),
   s9=(c13,0, c11, c10, c9,0, c15, c14).
(2)t=0;
    for i from 0 to 7 do
       (t,x1[i])=s1[i]+2(s2[i]+s3[i])+s4[i]
                 +s5[i]+t
    t1 = (t, x1) - p[t]
     if( t1 >= p[1] )
       t1 = t1 - p[1]
(3)t=0;
    for i from 0 to 7 do
       (t,x1[i])=s6[i]+s7[i]+s8[i]+s9[i]+t
    t2 = (t, x1) - p[t]
     if( t2 >= p[1] )
       t2 = t2 - p[1]
(4)Return( (t1-t2) mod p256 ).
```

Fig. 1. Fast reduction modulo NIST P256

Fig. 1의

t1= tW^{256}+x1 mod p,

t2= tW^{256}+x1 mod p

에서 여러 번을 뺄셈이 수행된다. 따라서 고속화를 위하여 p[i]=i·p를 사전 연산하여

```
t1 = (t, x1) - p[t]
 if( t1 >= p[1] )
      t1 = t1 - p[1]
```

와 같이 연산한다. 기존의 방법은 (2)와 (3)에서 모듈러 감산이 필요하며 마지막 연산에서 모듈러 감산이 추가적으로 발생한다[5-8].

III. 제안하는 정수 표현법과 모듈러 감산 방법

3.1 제안하는 정수 표현법

본 논문은 기존과 달리 새로운 정수 표현법을 제

안한다. 기존의 경우 32-bit 환경에 다음과 같이 256-bit 정수를 표현한다. a를 P256 소수체의 원소라하면 다음과 같다.

$$a=\sum_{i=0}^{i=7}a_i2^{32}=a_7 2^{224}+a_6 2^{192}+\cdots+a_1 2^{32}+a_0,$$
$$a_i\in\{0,2^{32}-1\}.$$

위와 같이 표현된 정수를 유한체에서 곱셈 연산하는 경우 $p=2^{256}-2^{224}+2^{192}+2^{96}-1$을 이용하여 모듈러 감산을 수행한다.

본 논문에서 제안하는 표현법은 다음과 같다.

$$a=\sum_{i=0}^{i=7}a_i2^{-224+32i}=a_7+a_6 2^{-32}+\cdots+a_0 2^{-224},$$
$$a_i\in 0,2^{32}-1.$$

제안하는 표현방법을 사용하여 표현한 정수를 유한체에서 연산하는 경우 모듈러 감산이 추가적으로 발생한다. 이 경우 기존 방법과 달리 다음을 사용하여 모듈러 감산을 수행한다.

$$p\cdot 2^{-256}=1-2^{-32}+2^{-64}+2^{-160}-2^{-256}.$$

3.2 제안하는 모듈로 감산 방법

제안하는 원소 표현법은 기존과 다르나 덧셈, 뺄셈, 곱셈 연산은 기존과 동일한 방법을 사용하면 된다.(소프트웨어의 경우 기존 코드를 그대로 사용하여도 된다.) 기존의 표현방법은 모듈러 곱셈 결과를

$$c=ab=\sum_{i=0}^{15}c_i2^{32\cdot i}=t_1\cdot 2^{256}+t_0$$

라 하면 모듈러 감산시 $t_1\cdot 2^{256}$을

$$t_1 2^{256}\equiv t_1(2^{224}-2^{192}-2^{96}+1)\ \mathrm{mod}p$$

와 같이 계산한다. 그러나 제안하는 원소 표현법의 곱셈 결과는

$$c=ab=\sum_{i=0}^{15}c_i2^{-448+32i}=t_1\cdot 2^{256}+t_0,$$
$$=c_{15}2^{32}+\sum_{i=7}^{14}c_i2^{-448+32i}+\sum_{i=0}^{6}c_i2^{-448+32i}$$

이다. 따라서 모듈러 감산은

$$c_{15}2^{32},\quad \sum_{i=0}^{6}c_i2^{-448+32i}$$

에서 수행된다. 이를 알고리즘으로 기술하면 Fig. 2와 같다.

Input: $c=(c15,c14,c13,\cdots,c2,c1,c0)$, $W=2^{32}$
$p[i]=i\cdot p\cdot 2^{-256}$
Output : $c=s\ \mathrm{mod}p\cdot 2^{-256}$

```
(1)t=0
   x1[0]= c[0]
   x1[1]= c[1]
   x1[2]= c[2]
   (t,x1[3])= c[0]+c[3]
   (t,x1[4])= c[1]+c[4]+t
   (t,x1[5])= c[2]+c[5]+t
   (t,x1[6])= 2c[0]+c[3]+c[6]+c[15]+t
   x1[7]= t
   t1= x1·2^32- x1 - c[15]·2^-128
(2)t=0
  (t,x1[0])= t1[0]+c[4]+c[7]+c[15]+2·c[1]
  (t,x1[1])= t1[1]+c[5]+c[8]+2·c[2]+t
  (t,x1[2])= t1[2]+c[6]+c[9]+3·c[0]
             +2·c[3]+t
  (t,x1[3])= t1[3]+c[1]+c[4]+c[10]+t
  (t,x1[4])= t1[4]+c[2]+c[5]+c[11]+t
  (t,x1[5])= t1[5]+c[3]+c[6]+c[12]
             +2·c[0]+t
  (t,x1[6])= t1[6]+c[13]+t
  (t,x1[7])= t1[7]+c[14]+t
   t= t1[8] + t
(3)c = (t, x1) - p[t]
   if( c >= p[1] )
      c = c - p[1]
(4)Return( c ).
```

Fig. 2. Proposed fast reduction modulo NIST P256

Fig. 2. (2), (3)의 연산에서 mod p를 제외한 0이 아닌 워드는 66개이다. 반면에 제안하는 원소 표현법에 의한 모듈러 감산 (1), (2)에서는 0이 아닌 워드가 51개이다. 또한 제안하는 방법에서는 계산 과정에서 음수가 발생하지 않는다. 즉, (1)의 t1은 입력과 상관없이 언제나 양수이다. 따라서 정수의 대소 비교 및 음수를 양수로 변환하는 과정이 필요없다. 또한 알고리즘 1의 경우 (2)와 (3)에서 mod p 연산을 수행하나 제안하는 방법에서는 (3)에서 한번만 수행하면 된다.

Table 1. Performance results of modular reduction or multiplication modulo the NIST prime P256(Cortex-M3 75MHz)

method	clock cycle
multiplication without modular reduction	2073
modular reduction[6]	1946
proposed modular reduction	1471

IV. 비교 및 결론

본 논문에서는 새로운 모듈러 곱셈 방법을 제시하였다. 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] S. Gueron, V. Krasnov, "Fast prime field elliptic-curve cryptography with 256-bit primes," J. Cryptographic Engin., vol. 5, pp. 141-151, Jun, 2015

[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