SCIS'99 The 1999 Symposium on
Cryptography and Information Security
Kobe, Japan, January 26-29, 1999
The Institute of Electronics,
Information and Communication Engineers

Elliptic Curve Algorithm on OEF with Frobenius Map

Tetsutaro Kobayashi * Kazumaro Aoki Fumitaka Hoshino
Kunio Kobayashi Hikaru Morita

Abstract— A new elliptic curve scalar multiplication algorithm is proposed. The throughput of the algorithm is about twice as fast as some conventional OEF-base algorithms. Furthermore, since this algorithm suits to conventional computational units such as 16, 32 and 64 bits, its base field $\mathbf{F}_{p^m}$ is expected to enhance elliptic curve operation efficiency more than $\mathbf{F}_p$ (p is a prime) and $\mathbf{F}_{2^n}$ for software implementation. This paper shows implemented results to clarify our techniques' efficiency.

Keywords: elliptic curve cryptosystem, scalar multiplication, OEF, finite field, Frobenius map, table reference method

1 Introduction

While a modular exponentiation was a prime subject to speed up the RSA scheme, a scalar multiplication of an elliptic curve point is focused to study to speed up the elliptic curve schemes such as EC DSA and EC ElGamal. Especially, elliptic curves over $\mathbf{F}_p$ (p is a prime) or $\mathbf{F}_{2^n}$ have been implemented by many companies and standardized by several organizations such as IEEE P1363 and ISO/IEC JTC1/SC27.

For the $\mathbf{F}_{2^n}$ type, many efficient computational algorithms have been proposed. Koblitz [2] introduced base-ϕ expansion method which uses a Frobenius map to multiply $\mathbf{F}_{2^n}$-rational points over the elliptic curve defined over $\mathbf{F}_2, \mathbf{F}_4, \mathbf{F}_8$ or $\mathbf{F}_{16}$. Müller [5] and Cheon et. al. [3] extended the base-ϕ expansion method to elliptic curves defined over $\mathbf{F}_{2^r}$, where r is less than m.

However, since the $\mathbf{F}_{2^n}$ type does not offer adequate speed on general purpose machines, it needs very high-capacity tables or special-purpose machines. If you select $|p|$ (the bit size of a prime p) to match to the operation unit of an individual computer, the scalar multiplication of $\mathbf{F}_{p^m}$ could be calculated faster than those of $\mathbf{F}_p$ or $\mathbf{F}_{2^n}$ where $|p^m|$ should be close to $|p|$ or $|2^n|(=n)$ under the condition of the same security level. Bailey and Paar newly proposed an elliptic curve scheme on Optimal Extension Fields (OEF), or an $\mathbf{F}_{p^m}$ type at Crypto'98 [1].

This paper extends the base-ϕ extension method from $\mathbf{F}_{2^n}$ to the general finite field $\mathbf{F}_{p^m}$ by using a table-reference method. Several table reference methods have been developed for schemes using fixed primitive points – base points – such as the DSA scheme [4]. Ours is the first to combine the Frobenius map and the table-reference method that does not need any pre-computation. When p equals two, this method is reduced to Koblitz's method. Different from Cheon's method, This method isn't limited to an elliptic curve defined on $\mathbf{F}_{2^r}$ The method works over OEF-type elliptic curves because the table reference method is effective even if p is large. If you select p close to 2^{16}, 2^{32} or 2^{64}, that are suitable operation units for computers, a performance of our method is about twice as fast as that of the ordinary OEF methods.

Section 2 describes the definition of OEF and the Frobenius map procedure. The new procedure of base-ϕ expansion method is proposed in Sect. 3. Section 4 shows how to construct the elliptic curve for the proposed method. Their efficiency is given in Sect. 5. Section 6 concludes this paper.

* kotetsu@isl.ntt.co.jp
NTT Information and Communication Systems Laboratories
(1-1 Hikarinooka, Yokosuka-Shi, Kanagawa, 239-0847 Japan)

2 Preparation

2.1 Frobenius Map

In this section, we define the Frobenius map. Let $E/\mathbf{F}_p$ denote a non-supersingular elliptic curve defined over a finite field $\mathbf{F}_p$ where p is a prime or any power of a prime. $P = (x, y)$ is an $\mathbf{F}_{p^m}$-rational point on $E/\mathbf{F}_p$. The Frobenius map ϕ is defined as

$$\phi : (x, y) \rightarrow (x^p, y^p).$$

The Frobenius map is an endomorphism map over $E(\mathbf{F}_{p^m})$. It satisfies the equation

$$\phi^2 - t\phi + p = 0, \qquad -2\sqrt{p} \leq t \leq 2\sqrt{p}. \tag{1}$$

Since E is non-supersingular, the endomorphism ring of E is an order of the imaginary quadratic field $\mathbf{Q}(\sqrt{t^2 - 4p})$ [6]. The ring $\mathbf{Z}[\phi]$ is a subring of the endomorphism ring.

To compute the Frobenius map ϕ takes negligible time, provided that element $a \in \mathbf{F}_{p^m}$ is represented using a normal basis of $\mathbf{F}_{p^m}$ over $\mathbf{F}_p$.

2.2 Normal Basis and Polynomial Basis

The elements of the field $\mathbf{F}_{p^m}$ can be represented in some different ways: for example, "polynomial basis"

or "normal basis." In polynomial basis, element $a \in \mathbf{F}_{p^m}$ is represented as

$$a = a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0, \quad (2)$$

where $a_i \in \mathbf{F}_p$ and α is a defining element of $\mathbf{F}_{p^m}$ over $\mathbf{F}_p$.

In normal basis, $a \in \mathbf{F}_{p^m}$ is represented as

$$a = a_{m-1}\alpha^{p^{m-1}} + \cdots + a_1\alpha^p + a_0\alpha \quad (3)$$

where $a_i \in \mathbf{F}_p$ and α is a generator of normal basis.

Addition and subtraction in $\mathbf{F}_{p^m}$ are quite fast in both representation forms. When you choose polynomial basis, multiplication and squaring can be done by reasonable speed.

When you choose normal basis, the p-th power operation, which is equal to the Frobenius map, is quite fast. Though multiplication isn't fast in general normal basis, there are several techniques for fast multiplication for $\mathbf{F}_{2^m}$ such as the optimal normal basis [7]. Thus, fast algorithms for scalar multiplication using the Frobenius map [2] have been developed using $\mathbf{F}_2$ or its extension field represented by normal basis.

We, on the other hand, developed a fast Frobenius map algorithm for OEF [1] which has a special polynomial basis.

2.3 Frobenius Map for OEF

Let OEF be the finite field $\mathbf{F}_{p^m}$ that satisfies the following

- p is a prime less than but close to the word size of the processor,
- p is a pseudo-Mersenne prime ($p = 2^n \pm c$; $\log_2 c \leq n/2$) and
- An irreducible binomial $f(x) = x^m - \omega$ exists.

Although [1] showed that OEF has an efficient algorithm for multiplication and squaring, there was no discussion of the Frobenius map. In this section, we present an algorithm [9] to compute the Frobenius map in OEF.

We consider the following polynomial basis representation of element $a \in \mathbf{F}_{p^m}$:

$$a = a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0$$

where $a_i \in \mathbf{F}_p$, $\alpha \in \mathbf{F}_{p^m}$ is a root of $f(x)$. Since we choose $|p|$ to be less than the processor's word size, we can represent a using m registers.

The Frobenius map moves a to a^p;

$$\phi(a) = a^p = a_{m-1}\alpha^{(m-1)p} + \cdots + a_1\alpha^p + a_0. \quad (4)$$

Since α is a root of $f(x) = 0$, $\alpha^m = \omega$,

$$\alpha^{ip} = \alpha^{(ip \bmod m)}\omega^{\lfloor (ip)/m \rfloor}$$

where $\lfloor x \rfloor$ is the maximum integer not exceeding x.

Assuming $\gcd(m, p) = 1$, $(i_1 p \bmod m) = (i_2 p \bmod m)$ is equivalent to $i_1 = i_2$. Thus, the map $\pi(i) \triangleq ip \bmod m$ is bijective.

We rewrite Equation (4) using $\pi(i)$ as follows:

$$a^p = a'_{m-1}\alpha^{m-1} + \cdots + a'_1\alpha + a'_0,$$
$$\text{where } a'_{\pi(i)} \triangleq a_i\omega^{\lfloor \frac{ip}{m} \rfloor}.$$

Since p, m and ω is independent of an element a, we can pre-compute $\omega_i = \omega^{\lfloor \frac{ip}{m} \rfloor}$ before computing the Frobenius map. According, the complete procedure to compute the Frobenius map to an element on OEF is as follows;

[Frobenius Map Procedure for OEF]

Input: $[a_0, \ldots, a_{m-1}](= a)$
Output: $[a'_0, \ldots, a'_{m-1}](= \phi(a))$
Step 1: compute $b_i = a_i\omega_i$, for $i = 1$ to $m-1$.
Step 2: compute $a'_{\pi(i)} = b_i$, for $i = 1$ to $m-1$.
Step 3: $a'_0 = a_0$.

This procedure needs only $m-1$ multiplication on $\mathbf{F}_p$. This takes negligible time compared to multiplication on $\mathbf{F}_{p^m}$, which needs m^2 multiplication [1] on $\mathbf{F}_p$.

3 Base-ϕ Scalar Multiplication

3.1 Procedure

In this section, we describe the base-ϕ scalar multiplication method. The following procedure computes $Q = kP$ for inputs P and k, where P is an $\mathbf{F}_{p^m}$-rational point on E and $0 < k < N_m$ and N_m denotes the number of $\mathbf{F}_{p^m}$-points on E. Recall t is a trace of E.

[Base-ϕ Scalar Multiplication Procedure]

Step 1: Expansion of k
- **Step 1-1:** $i \leftarrow 0, x \leftarrow k, y \leftarrow 0, u_j \leftarrow 0$ for $\forall j$
- **Step 1-2:** if ($x = 0$ and $y = 0$) then goto **Step 2:**
- **Step 1-3:** $u_i \leftarrow x \bmod p$.
- **Step 1-4:** $v \leftarrow (x - u_i)/p, x \leftarrow tv + y, y \leftarrow -v$
- **Step 1-5:** $i \leftarrow i + 1$ goto **Step 1-2:**

Step 2: Reduction
- **Step 2-1:** $d_i \leftarrow u_i + u_{i+m} + u_{i+2m}$, for $0 \leq i < m$.
- **Step 2-2:** $c_i \leftarrow d_i - z$ for $0 \leq i \leq m-1$, where z is an integer that minimize $\sum_i w_H(c_i)$.

Step 3: Table Reference Multiplication
- **Step 3-1:** $P_i \leftarrow \phi^i P$ for $0 \leq i < m$
- **Step 3-2:** $Q \leftarrow \mathcal{O}, j \leftarrow |p| + 1$
- **Step 3-3:** $Q \leftarrow 2Q$
- **Step 3-4:** for ($i = 0$ to m) {
  if ($c_{ij} = 1$) then $Q \leftarrow Q + P_i$
  }
- **Step 3-5:** $j \leftarrow j - 1$
- **Step 3-6:** if ($j \geq 0$) then goto **Step 3-3.**

[1] This is by straightforward method.

We use $w_H(x)$ by Hamming weight of x expressed in binary digit.

First, the procedure finds u_i such that $k = \sum_{i=0}^{l} u_i\phi^i$ in **Step 1** by using $\phi^2 - t\phi + p = 0$, where t is a trace of E. This part of the procedure is nearly equal to the procedure in [5] and the integer l is discussed in Sect. 3.2.

Next, it reduces the series of base-ϕ expansion $\{u_0, \ldots, u_{2m+3}\}$ into $\{c_0, \ldots, c_{m-1}\}$ in **Step 2**. Detailed explanation is given in Sect. 3.3.

Finally, it calculates kP using $\{c_0, \ldots, c_{m-1}\}$ in **Step 3**. Detailed discussion about this part is in [8].

3.2 Loop Number of Step 1

Theorem 1 *[5] Let $p \geq 4$ and let $k \in \mathbf{Z}[\phi]$. If we set $l = \lceil 2\log_p ||k|| \rceil + 3$, then there exist rational integers $-p/2 \leq u_i \leq p/2$, $0 \leq i \leq l$, such that*

$$k = \sum_{i=0}^{l} u_i\phi^i \tag{5}$$

where $||k|| := \sqrt{k\bar{k}}$ and $\bar{k}$ is the complex conjugate of k and $\lceil x \rceil$ is the minimum integer greater than or equal to x.

Since the proof of Theorem 1 in [5] does not assume p to be a small power of two, the loop in **Step 1** ends at most $i \leq 2\lceil \log_p ||k|| \rceil + 3$ for general p.

3.3 Reduction of the Result of Expansion

This section explains the background of the procedure in **Step 2**. "Type I Reduction" is related to **Step 2-1** and "Type II Reduction" is related to **Step 2-2**.

3.3.1 Type I Reduction

If k is randomly chosen from $0 < k < N_m$, we can assume $k \simeq p^m$ and $l = 2\lceil \log_p k \rceil + 3 \simeq 2m + 3$. However, the series of base-ϕ expansion $\{u_0, \ldots, u_{2m+3}\}$ can be easily reduced to $\{d_0, \ldots, d_{m-1}\}$ by using following equation.

$$\phi^m = 1 \qquad \text{in } \mathrm{End}_E. \tag{6}$$

This is because $x^{p^m} = x$ for $\forall x \in \mathbf{F}_{p^m}$.

$$\begin{aligned} \sum_{i=0}^{2\lceil \log_p k \rceil + 3} u_i\phi^i &= \sum_{i=0}^{m-1} (u_i + u_{i+m} + u_{i+2m})\phi^i \\ &= \sum_{i=0}^{m-1} d_i\phi^i. \end{aligned}$$

3.3.2 Type II Reduction

We can accelerate **Step 3** by decreasing the density of '1's in the bit expression of d_i by using Equation (7) after the Type I Reduction.

$$\sum_{i=0}^{m-1} \phi^i = 0 \tag{7}$$

Equation (7) is derived from Equation (6) and $\phi \neq 1$.

The theoretical required time for scalar multiplication is shown in Table 1. "Type I base-ϕ" denotes the proposed procedure using d_i instead of c_i at **Step 3** and "Type II base-ϕ" denotes the full proposed procedure.

Table 1: Required time for scalar multiplication

algorithm	EC-Addition	EC-Doubling
binary method	$\frac{1}{2}m\|p\|$	$m\|p\|$
signed binary method	$\frac{3}{8}m\|p\|$	$m\|p\|$
Type I expansion	$\frac{1}{2}m\|p\|$	$\|p\|$
Type II expansion	$\frac{1}{4}m\|p\|$	$\|p\|$

4 Elliptic Curve Generation

In this section, we discuss how to generate elliptic curves for the base-ϕ expansion method.

Let p be prime, where $p > 3$ and let E be the elliptic curve

$$Y^2 = X^3 + aX + b \tag{8}$$

over $\mathbf{F}_p$ ($a, b \in \mathbf{F}_p$). We should define elliptic curve E over $\mathbf{F}_p$ to use base-ϕ expansion. In such a case, we can easily compute N_m by using Theorem 2.

Theorem 2 (Weil Conjecture [6, pp.132-137]) *Suppose E is an elliptic curve over $\mathbf{F}_p$ and $t := p + 1 - N_1$. The number of $\mathbf{F}_{p^m}$-points on E*

$$N_m = p^m + 1 - (\alpha^m + \beta^m),$$

where α, β are the roots of $x^2 - tx + p$.

From the view point of cryptography, E is "good" elliptic curve if N_m has a large prime factor. Since $N_n \simeq p^n$ and $N_n | N_m$ if $n|m$, we have the best chance of getting a large prime factor of N_m when m is prime. We can generate elliptic curve $E/\mathbf{F}_p$ with N_m that has a large prime factor by using the following procedure.

[Elliptic Curve Generation Procedure for Base-ϕ Expansion]

Step 1: Generate $E/\mathbf{F}_p$ randomly and find its order $N_1 = p + 1 - t$.
Step 2: Find N_m using the Weil conjecture.
Step 3: If N_m doesn't have a large enough prime factor, go to **Step 1**.

For example, let $p = 2^{16} - 129$, $m = 11$ and $M := \frac{N_m}{N_1}$ ($N_1 \simeq 2^{16}, N_m \simeq 2^{176}$). We can find some parameters such that M becomes prime. One example is $a = -3, b = 179, |M| = 160$.

5 Total Efficiency

We show the total efficiency of a base-ϕ scalar multiplication method.

Table 2 shows the current results of our elliptic curve implementation. We implemented 16-bit and 32-bit cases to a 200 MHz Intel Pentium Pro PC to examine the impact of the base-ϕ scalar multiplication method. We use the Chudnovsky Jacobian coordinates [10] to represent a rational-point on an elliptic curve, and the parameters used in the implementation are following:

[32-bit OEF]
$p = 2^{31} - 1, m = 7,$
minimal polynomial $f(x) = x^7 - 3,$
$E : y^2 = x^3 - 3x - 212$

[16-bit OEF]
$p = 2^{16} - 129, m = 11,$
minimal polynomial $f(x) = x^{11} - 3,$
$E : y^2 = x^3 - 3x + 179$

The results clarify that the proposed base-ϕ expansion method speeds up the scalar multiplication about twice as fast as the traditional signed binary method in the 16-bit and 32-bit OEF cases.

Table 2: Scalar Multiplication Speeds (msec)

p	$\|M\|$	Signed Binary	Type I	Type II
32-bit	186-bit	7.87	4.64	4.11
16-bit	160-bit	79.9	39.4	37.7

6 Conclusions

This paper generalized the base-ϕ scalar multiplication method to suit finite fields with large characteristic (such as OEF). We also proposed a new technique for accelerate the base-ϕ method, where ϕ is an element of End_E.

This paper showed implemented results to clarify our techniques' efficiency. In the 16-bit and 32-bit OEF cases, the base-ϕ expansion method speeds up the calculation speed to twice as fast as traditional techniques.

Acknowledgments

We are very grateful to Eisaku Teranishi of NTT Advanced Technology Corporation to implement a part of our algorithm.

References

[1] D. V. Bailey and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms," Advances in Cryptology – CRYPTO '98, Lecture Notes in Computer Science 1462, pp.472-485, Springer, 1998.

[2] N. Koblitz, "CM-Curves with Good Cryptographic Properties," Advances in Cryptology – CRYPTO'91, Lecture Notes in Computer Science 576, pp.279-287, Springer-Verlag, 1992.

[3] J. H. Cheon, S. Park and D. Kim, "Two Efficient Algorithms for Arithmetic of Elliptic Curves Using Frobenius Map," Public Key Cryptography: Proceedings of the First international workshop, PKC '98, Lecture Notes in Computer Science 1431, pp.195-202, Springer, 1998.

[4] E. F. Brickell, D. M. Gordon, K. S. McCurley and D. B. Wilson, "Fast Exponentiation with Precomputation," Advances in Cryptology – EUROCRYPT'92, Lecture Notes in Computer Science 658, pp.200-207, Springer, 1993.

[5] V. Müller, "Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two," Journal of Cryptology(1998) 11, pp.219-234.

[6] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York, 1986.

[7] R. Mullin, I. Onyszchuk, S. Vanstone, and R. Wilson, "Optimal Normal Basis in $GF(p^n)$," Discrete Applied Mathematics, 22:149-161, 1988.

[8] T. Kobayashi and H. Morita, "Table Reference Method for Complex Multiplication," 1998 Engineering Sciences Society Conference of IEICE, SA-5-3, Oct, 1998.

[9] T. Kobayashi, K. Kobayashi and H. Morita, "Elliptic Curve Computation for OEF using Frobenius Map," ISEC98-33, pp.41-46, Sep, 1998.

[10] H. Cohen, A. Miyaji and T. Ono, "Efficient Elliptic Curve Exponentiation Using Mixed Coordinates," Advances in Cryptology - ASIACRYPT'98, Lecture Notes in Computer Science 1514, pp.51-65, Springer-Verlag (1998)