Memoirs of the Faculty of Engineering, Okayama University, Vol. 45, pp. 54-59, January 2011

Squaring Algorithm Efficient for Cubic Extension Field Derived with Pseudo Gauss Period Normal Basis

Kenta NEKADO*
Graduate School of Natural Science and Technology, Okayama University
3–1–1, Tsushima–naka, Kita–ward, Okayama, Okayama 700–8530, Japan

Yusuke TAKAI*
Department of Communication Network Engineering, Faculty of Engineering, Okayama University
3–1–1, Tsushima–naka, Kita–ward, Okayama, Okayama 700–8530, Japan

Yasuyuki NOGAMI*
Graduate School of Natural Science and Technology, Okayama University
3–1–1, Tsushima–naka, Kita–ward, Okayama, Okayama 700–8530, Japan

Yoshitaka MORIKAWA*
Graduate School of Natural Science and Technology, Okayama University
3–1–1, Tsushima–naka, Kita–ward, Okayama, Okayama 700–8530, Japan

(Received January 5, 2011)

Recently, pairing–based cryptographies have attracted much attention. For fast pairing calculation, not only pairing algorithms but also arithmetic operations in extension field should be efficient. Especially for final exponentiation included in pairing calculation, squaring is more important than multiplication. This paper considers squaring algorithms efficient for cubic extension field which is often used for pairing implementaions.

1 INTRODUCTION

In this decade, pairing over elliptic curve has attracted much attention to realize epochal public–key cryptographic applications such as ID–based cryptography [1] and group signature [2]. As pairings enable to efficiently work for these appplications, several pairings such as Ate pairing [3], Xate pairing [4], and R–ate pairing [5] have been proposed. The above pairings consist of two steps, one is a calculation by Miller's algorithm, and the other is so–called final exponentiation. In order to provide fast these calculations, arithmetic operations in the definition field should be efficient. Especially, for final exponentiation, squaring is more important than the other operations such as multiplication.

As the definition field of the above pairings, most of researchers use optimal extension field (OEF) [6]. OEF can efficiently carry out arithmetic operations with some efficient algorithms such as Karatsuba multiplication algorithm [6]. Of course, OEF provides fast squaring by using efficient squaring algorithms such as complex squaring algorithm [7] and Chung–Hasan squaring (CH–SQR) algorithm [8]. However, OEF can not often be used as the definition field of pairings due to some mismatches of the conditions of the parameters such as characteristic p and extension degree m between OEF and pairing. Thus, construction method of extension field available regardless of these parameters is required.

For this requirement, the authors have proposed type–$\langle h, m\rangle$ all one polynomial field (AOPF) [9]. Type–$\langle h, m\rangle$ AOPF is constructed by Gauss period normal basis (GNB) [10] whose preparation needs a certain positive integer parameter h in addition to characteristic p and extension degree m. By changing h, type–$\langle h, m\rangle$ AOPF is flexibly available for almost all pairs of p and m. Additionally, the authors have proposed an efficient multiplication algorithm in type–$\langle h, m\rangle$ AOPF, namely type–$\langle h, m\rangle$ cyclic vector multiplication algorithm (CVMA). By using type–$\langle h, m\rangle$ CVMA, type–$\langle h, m\rangle$ AOPF can carry out multiplication and squaring almost as efficient as OEF; however, in the cases that $m = 2$ and 3, compared to the squaring algorithms efficient for OEF, type–$\langle h, m\rangle$ CVMA is not efficient for squaring. Thus, for type–$\langle h, m=2\rangle$ AOPF, Kato et al. and the authors have proposed efficient squaring algorithms [11, 12] having the equivalent efficiency of the squaring algorithm efficient for OEF. On the other hand, for type–$\langle h, m=3\rangle$ AOPF, such squaring algorithm have

*E-mail: { nekado, takai, nogami, morikawa }@trans.cne.okayama-u.ac.jp

not been proposed yet, although cubic extension field is often used for pairing implementations [4, 13]. Therefore, this paper introduces pseudo Gauss period normal basis (PGNB), and derives a squaring algorithm efficient for the type–$\langle h, m=3\rangle$ AOPF with PGNB.

Notation: $\mathbb{F}_p$, $\mathbb{F}_{p^m}$, $\mathbb{F}_{p^m}^*$, and $E(\mathbb{F}_{p^m})$ denote a prime field, an m–th extension field over $\mathbb{F}_p$, the multiplicative group in $\mathbb{F}_{p^m}$, and the elliptic curve defined over $\mathbb{F}_{p^m}$. For two integers m and n, $m \,|\, n$ means that m divides n. M_m, S_m, A_m, D_m, and L_m denote the calculation costs of a multiplication, a squaring, an addition (a subtraction), a doubling, an one–half multiplication in $\mathbb{F}_{p^m}$, respectively.

2 FUNDAMENTALS

This section briefly goes over optimal extension field (OEF) and all one polynomial field (AOPF).

2.1 Optimal Extension Field (OEF)

Bailey et al. have proposed OEF [6], which achieves efficient arithmetic operations by using some efficient algorithm such as Karatsuba multiplication method [6]. OEF is constructed by a polynomial basis as

$$\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}, \quad \alpha = \sqrt[m]{n}, \tag{1}$$

where n is the constant term of an m–th degree irreducible binomial over $\mathbb{F}_p$. In order to prepare the above polynomial basis, p and m need to satisfy the conditions such that each prime factor of m divides $p-1$, for example. Thus, OEF is not available for every pair of p and m.

2.2 All One Polynomial Field (AOPF)

The authors have proposed type–$\langle h, m\rangle$ AOPF [9], which efficiently carries out multiplication and squaring with cyclic vector multiplication algorithm (CVMA) [9] in almost the same of OEF. Additionally, type–$\langle h, m\rangle$ AOPF does not need any arithmetic operations for Frobenius mapping since type–$\langle h, m\rangle$ AOPF is constructed by Gauss period normal basis (GNB) [10]. In what follows, this paper briefly introduces GNB and CVMA.

2.2.1 Gauss Period Normal Basis (GNB)

Suppose a positive integer h which satisfies

Condition 1 (the h of GNB)

1) $r = hm + 1$ is a prime number not equal to p.
2) $\gcd(hm/e, m) = 1$, where e is the order of p in $\mathbb{F}_r$.

Then, let d be a primitive h–th root of unity in $\mathbb{F}_r^*$, the following multicative group is obtained,

$$\left\langle \left\{ \left\langle\!\left\langle p^i d^k \right\rangle\!\right\rangle : 0 \le i < m,\, 0 \le k < h \right\}, \cdot \right\rangle = \mathbb{F}_r^*, \tag{2}$$

where $\langle \mathbb{S}, \cdot \rangle$ means a multiplicative group with a non–empty set $\mathbb{S}$ and $\langle\!\langle t \rangle\!\rangle$ denotes $t \pmod r$ with an integer t and a prime number r. Let β be a primitive r–th root of unity in $\mathbb{F}_{p^e}^*$. In other words, it is a zero of the all one polynomial (AOP) over $\mathbb{F}_p$ as

$$f(t) = \frac{t^r - 1}{t - 1} = \sum_{i=0}^{r-1} t^i. \tag{3}$$

Then, GNB [10] is defined with the above h, d and β as follows.

$$\{\gamma, \gamma^p, \gamma^{p^2}, \cdots, \gamma^{p^{m-1}}\}, \quad \gamma = \sum_{k=0}^{h-1} \beta^{d^k} \in \mathbb{F}_{p^m}. \tag{4}$$

This paper especially calls it type–$\langle h, m\rangle$ GNB. This basis has the following properties.

Property 1 The summation of the elements in type–$\langle h, m\rangle$ GNB is given by

$$\sum_{i=0}^{m-1} \gamma^{p^i} = \sum_{i=0}^{m-1} \sum_{k=0}^{h-1} \beta^{p^i d^k} = \sum_{i=1}^{r-1} \beta^i = -1, \tag{5}$$

because p and d satisfy Eq. (2) and the β is a zero of the AOP as shown in Eq. (3).

Property 2 Type–$\langle h, m\rangle$ GNB can be prepared whenever $4p \nmid m(p-1)$ [10].

Since type–$\langle h, m\rangle$ GNB is prepared with a zero β of the AOP given by Eq. (3), the extension field constructed by this basis is called type–$\langle h, m\rangle$ AOPF. According to **Prop**. 2, type–$\langle h, m\rangle$ AOPF is available for every pair of p and m when $p > m$.

2.2.2 Cyclic Vector Multiplication Algorithm (CVMA)

Generally, as the parameter h of type–$\langle h, m\rangle$ GNB becomes larger, multiplication and squaring in type–$\langle h, m\rangle$ AOPF become more inefficient. In this section, in order to give an example of the relation between the parameter h and the efficiencies of multiplication and squaring, this paper briefly shows type–$\langle h, m\rangle$ CVMA [9] which is a multiplication (squaring) algorithm efficient for type–$\langle h, m\rangle$ AOPF.

Let A, B and Y in type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^m}$ be

$$A = \sum_{l=0}^{m-1} a_l \gamma^{p^l},\ B = \sum_{l=0}^{m-1} b_l \gamma^{p^l},\ Y = AB = \sum_{l=0}^{m-1} y_l \gamma^{p^l}, \tag{6a}$$

$$a_l,\, b_l,\, y_l \in \mathbb{F}_p. \tag{6b}$$

Then, type–$\langle h, m\rangle$ CVMA calculates y_l in Eq. (6a) as

$$y_l = \begin{cases} -a_l b_l - c_l + h c_m & (\text{when } h \text{ is odd}), \\ -a_l b_l - c_l & (\text{when } h \text{ is even}), \end{cases} \tag{7a}$$

$$c_l = \sum_{0 \le i < j < m} \sum (a_i - a_j)(b_i - b_j) \sum_{k=0}^{h-1} \delta_l\big[\eta[i, j, k]\big], \tag{7b}$$

where δ_s denotes the unit impulse function as

$$\delta_s(t) = \begin{cases} 1 & (\text{when } s = t), \\ 0 & (\text{otherwise}), \end{cases} \tag{8}$$

and η means a function as

$$\eta[i,j,k] = \epsilon\big[\langle\!\langle p^i + p^j d^k \rangle\!\rangle\big], \tag{9a}$$
$$\epsilon\big[\langle\!\langle p^i d^k \rangle\!\rangle\big] = i, \quad \text{and} \quad \epsilon[0] = m. \tag{9b}$$

With type–$\langle h, m\rangle$ CVMA, the calculation amounts of a multiplication and a squaring are explicitly given as follows.

$$M_m = \frac{m(m+1)}{2}M_1 + \begin{cases} \left(\frac{m(m-1)(h+2)}{2} - 1 + m\right)A_1 + H_1 & \\ \qquad\qquad \text{(when } h \text{ is odd)}, & \text{(10a)} \\ \left(\frac{m(m-1)(h+2)}{2}\right)A_1 \quad \text{(when } h \text{ is even)}, & \end{cases}$$

$$S_m = \frac{m(m+1)}{2}S_1 + \begin{cases} \left(\frac{m(m-1)(h+1)}{2} - 1 + m\right)A_1 + H_1 & \\ \qquad\qquad \text{(when } h \text{ is odd)}, & \text{(10b)} \\ \left(\frac{m(m-1)(h+1)}{2}\right)A_1 \quad \text{(when } h \text{ is even)}, & \end{cases}$$

where H_1 denotes the calculation cost of a scalar–h multiplication in $\mathbb{F}_p$. As shown in Eq. (10a), type–$\langle h, m\rangle$ CVMA needs more additions in $\mathbb{F}_p$ as h becomes larger. Usually, A_1 is much smaller than M_1. However, if the number of additions in $\mathbb{F}_p$ is quite large, it will not be negligible. Thus, in order to carry out type–$\langle h, m\rangle$ CVMA more efficiently, we should adapt the minimal h among h's such that the conditions for type–$\langle h, m\rangle$ GNB are satisfied. Moreover, it is the most desirable when $h = 1$ or $h = 2$ because then type–$\langle h, m\rangle$ CVMA are the most efficient.

3 SQUARING ALGORITHMS EFFICIENT FOR $\mathbb{F}_{p^3}$

Chung and Hasan have derived some squaring algorithms efficient for OEF $\mathbb{F}_{p^3}$ from a certain approach [8]. Also in the case of type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^3}$, we can apply the derivation approach; however, it is a daunting challenge. Therefore, in order to derive a squaring algorithm efficient for type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^3}$, we must consider the different approach.

This section first runs over Chung–Hasan squaring (CH–SQR) algorithms and the derivation approach, and then shows the efficiency of the algorithms. After that, by using the different approach, the authors derive a squaring algorithm in type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^3}$ which has the efficiency equivalent to CH–SQR algorithms.

3.1 Squaring Algorithm Efficient for OEF $\mathbb{F}_{p^3}$

Let A and Y in OEF $\mathbb{F}_{p^3}$ be

$$A = \sum_{l=0}^{2} a_l \alpha^l, \quad Y = A^2 = \sum_{l=0}^{2} y_l \alpha^l, \quad a_l,\, y_l \in \mathbb{F}_p. \tag{11}$$

Schoolbook method [7] calculates y_l in Eq. (11) as

$$y_0 = a_0^2 + 2na_1a_2, \tag{12a}$$
$$y_1 = 2a_0a_1 + na_2^2, \tag{12b}$$
$$y_2 = a_1^2 + 2a_0a_2, \tag{12c}$$

where n is the constant number shown in Eq. (1). For Eq. (12), the following matrix is considered with the coefficients.

$$\left[\begin{array}{c|cccccc} & a_0^2 & 2a_0a_1 & a_1^2 & 2a_1a_2 & a_2^2 & 2a_2a_0 \\ \hline y_0 & 1 & 0 & 0 & n & 0 & 0 \\ y_1 & 0 & 1 & 0 & 0 & n & 0 \\ y_2 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]. \tag{13}$$

Let $U(y_l)$ denote the row vector of the above matrix. Chung and Hasan have derived efficient squaring algorithms by representing $U(y_0)$, $U(y_1)$ and $U(y_2)$ with 5 of the 24 vectors shown in **Table** 1 as follows [8].

Table 1: The vectors prepared by Chung and Hasan

i	V_i	$V_i \cdot W$ †
1	[1 0 0 0 0 0]	a_0^2
2	[0 0 0 0 1 0]	a_2^2
3	[1 1 1 1 1 1]	$(a_0 + a_1 + a_2)^2$
4	[1 −1 1 −1 1 1]	$(a_0 - a_1 + a_2)^2$
5	[16 8 4 2 1 4]	$(4a_0 + 2a_1 + a_2)^2$
6	[16 −8 4 −2 1 4]	$(4a_0 - 2a_1 + a_2)^2$
7	[1 2 4 8 16 4]	$(a_0 + 2a_1 + 4a_2)^2$
8	[1 −2 4 −8 16 4]	$(a_0 - 2a_1 + 4a_2)^2$
9	[0 1 0 0 0 0]	$2a_0a_1$
10	[0 0 0 1 0 0]	$2a_1a_2$
11	[4 3 2 1 0 2]	$2(a_0 + a_1 + a_2)(2a_0 + a_1)$
12	[−4 3 −2 1 0 2]	$2(a_0 - a_1 + a_2)(-2a_0 + a_1)$
13	[0 1 2 3 4 2]	$2(a_0 + a_1 + a_2)(a_1 + 2a_2)$
14	[0 1 −2 3 −4 −2]	$2(a_0 - a_1 + a_2)(a_1 - 2a_2)$
15	[1 0 −1 0 1 −1]	$(a_0 + a_1 - a_2)(a_0 - a_1 - a_2)$
16	[0 1 0 −1 0 0]	$2a_1(a_0 - a_2)$
17	[1 0 −1 −1 0 −1]	$(a_0 - a_1 - 2a_2)(a_0 + a_1)$
18	[0 1 1 0 −1 1]	$(2a_0 + a_1 - a_2)(a_1 + a_2)$
19	[1 0 −1 1 0 −1]	$(a_0 + a_1 - 2a_2)(a_0 - a_1)$
20	[0 1 −1 0 1 −1]	$(2a_0 - a_1 - a_2)(a_1 - a_2)$
21	[1 0 0 0 −1 0]	$(a_0 + a_2)(a_0 - a_2)$
22	[0 1 0 1 0 0]	$2a_1(a_0 + a_2)$
23	[0 4 0 1 0 0]	$2a_1(4a_0 + a_2)$
24	[0 1 0 4 0 0]	$2a_1(a_0 + 4a_2)$

† W denotes a vector as $[a_0^2\ 2a_0a_1\ a_1^2\ 2a_1a_2\ a_2^2\ 2a_2a_0]^T$.

The case of CH–SQR$_1$ algorithm:

$$U(y_0) = V_1 + nV_{10}, \tag{14a}$$
$$U(y_1) = nV_2 + V_9, \tag{14b}$$
$$U(y_2) = V_1 + V_2 - V_{15}. \tag{14c}$$

The case of CH–SQR$_2$ algorithm:

$$U(y_0) = V_1 + nV_{10}, \tag{15a}$$
$$U(y_1) = nV_2 + V_9, \tag{15b}$$
$$U(y_2) = V_1 + V_2 - V_4 + V_9 + V_{10}. \tag{15c}$$

The case of CH–SQR$_3$ algorithm:

$$U(y_0) = V_1 + nV_{10}, \tag{16a}$$
$$U(y_1) = nV_2 + V_3 + V_{10} - V_{25}, \tag{16b}$$
$$U(y_2) = -V_1 - V_2 + V_{25}. \tag{16c}$$

where $V_{25} = (V_3 + V_4)/2$.

Then, for each algorithms, the calculation amounts of a squaring in OEF $\mathbb{F}_{p^3}$ are given by **Table** 2.

3.2 Squaring Algorithm Efficient for AOPF $\mathbb{F}_{p^3}$

Let A and Y in type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^3}$ be

$$A=\sum_{l=0}^{2} a_l\gamma^{p^l}, \quad Y=A^2=\sum_{l=0}^{2} y_l\gamma^{p^l}, \quad a_l,\, y_l \in \mathbb{F}_p. \tag{17}$$

The following positive integers satisfy **Cond**. 1 (1).

$$h = 2, 4, 6, 10, 12, \ldots \tag{18}$$

For example, type–$\langle h=2, m=3\rangle$ CVMA calculates y_l in Eq. (17) as

$$y_0 = -(a_0-a_{q_1})^2 - (a_{q_1}-a_{q_2})^2 - a_0^2, \tag{19a}$$
$$y_{q_1} = -(a_0-a_{q_2})^2 - (a_{q_1}-a_{q_2})^2 - a_{q_1}^2, \tag{19b}$$
$$y_{q_2} = -(a_0-a_{q_1})^2 - (a_0-a_{q_2})^2 - a_{q_2}^2, \tag{19c}$$

where q_1 and q_2 are given by

$$[q_1, q_2] = \begin{cases} [1,2] & (\text{when } \langle\langle p\rangle\rangle = 2 \text{ or } 5), \\ [2,1] & (\text{when } \langle\langle p\rangle\rangle = 3 \text{ or } 4), \end{cases} \tag{20}$$

Eq. (19) is expanded as

$$y_0 = -2a_0^2+2a_0a_{q_1}-2a_{q_1}^2+2a_{q_1}a_{q_2}-a_{q_2}^2, \tag{21a}$$
$$y_{q_1} = -a_0^2-2a_{q_1}^2+2a_{q_1}a_{q_2}-2a_{q_2}^2+2a_{q_2}a_0, \tag{21b}$$
$$y_{q_2} = -2a_0^2+2a_0a_{q_1}-a_{q_1}^2-2a_{q_2}^2+2a_{q_2}a_0. \tag{21c}$$

For Eq. (21), the following matrix is considered with the coefficients.

$$\left[\begin{array}{c|cccccc} & a_0^2 & 2a_0a_{q_1} & a_{q_1}^2 & 2a_{q_1}a_{q_2} & a_{q_2}^2 & 2a_{q_2}a_0 \\ \hline y_0 & -2 & 1 & -2 & 1 & -1 & 0 \\ y_{q_1} & -1 & 0 & -2 & 1 & -2 & 1 \\ y_{q_2} & -2 & 1 & -1 & 0 & -2 & 1 \end{array}\right]. \tag{22}$$

In the case of OEF $\mathbb{F}_{p^3}$, there are a lot of non–zero elements in the coefficient matrix as Eq. (13). Thus, in order to make squaring more efficient, it is comparatively easy to choose 5 of the 24 vectors shown in **Table** 1. On the other hand, in the case of type–$\langle h, m=3\rangle$ AOPF, because there are few non–zero elements in the coefficient matrix as the above example, it is very difficult to find a pair of the 5 suitable vectors in the same way of OEF $\mathbb{F}_{p^3}$. Therefore, in what follows, let us consider the different approach.

3.2.1 Pseudo GNB (PGNB)

For type–$\langle h, m\rangle$ GNB shown in Eq. (4), let us consider to replace γ with 1 as

$$\{\gamma^p, \gamma^{p^2}, \cdots, \gamma^{p^{m-1}}, 1\}. \tag{23}$$

The above set also forms a basis because it is obvious that the elements in the set are independent of each other, according to Eq. (5). This paper especially calls this basis type–$\langle h, m\rangle$ PGNB.

Let B and Z in type–$\langle h, m\rangle$ AOPF $\mathbb{F}_{p^3}$ be represented with type–$\langle h, m\rangle$ PGNB as

$$B=\sum_{l=1}^{2} b_l\gamma^{p^l} + b_3\cdot 1, \quad Z=\sum_{l=1}^{2} z_l\gamma^{p^l} + z_3\cdot 1, \tag{24a}$$
$$b_l,\, z_l \in \mathbb{F}_p. \tag{24b}$$

When $A = B$ and $Y = Z$, the following equations are obtained from Eq. (5).

$$a_0 = -b_3, \quad a_1 = b_1 - b_3, \quad a_2 = b_2 - b_3, \tag{25a}$$
$$y_0 = -z_3, \quad y_1 = z_1 - z_3, \quad y_2 = z_2 - z_3. \tag{25b}$$

Additionally, from Eq. (25), the following equations are obtained.

$$b_1 = a_1 - a_0, \quad b_2 = a_2 - a_0, \quad b_3 = -a_0, \tag{26a}$$
$$z_1 = y_1 - y_0, \quad z_2 = y_2 - y_0, \quad z_3 = -y_0. \tag{26b}$$

This paper derives each squaring algorithm efficient for type–$\langle h = 2, m = 3\rangle$ and type–$\langle h = 4, m = 3\rangle$ AOPF with Eq. (25), (26).

3.2.2 Derivation with PGNB When $h = 2$

According to Eq. (25), Eq. (21) is given by

$$z_{q_1} = -b_{q_2}^2+2b_3b_{q_1}, \tag{27a}$$
$$z_{q_2} = b_{q_1}^2-2b_{q_1}b_{q_2}-b_{q_2}^2+2b_{q_2}b_3, \tag{27b}$$
$$z_3 = 2b_{q_1}^2-2b_{q_1}b_{q_2}+b_{q_2}^2+b_3^2. \tag{27c}$$

For Eq. (27), the following matrix is considered with the coefficients.

$$\left[\begin{array}{c|cccccc} & b_{q_1}^2 & 2b_{q_1}a_{q_2} & b_{q_2}^2 & 2b_{q_2}b_3 & b_3^2 & 2b_3b_{q_1} \\ \hline z_{q_1} & 0 & 0 & -1 & 0 & 0 & 2 \\ z_{q_2} & 1 & -1 & -1 & 1 & 0 & 0 \\ z_3 & 2 & -1 & 1 & 0 & 1 & 0 \end{array}\right]. \tag{28}$$

The above matrix has more non–zero elements than that of Eq. (22). Thus, for example, we can easily consider the deformation of Eq. (27) as

$$z_{q_1} = -b_{q_2}^2+2b_{q_1}b_3, \tag{29a}$$
$$z_{q_2} = (b_{q_1}-b_{q_2})^2-b_{q_2}^2+2b_{q_2}b_3, \tag{29b}$$
$$z_3 = \frac{1}{2}(2b_{q_1}-b_{q_2}+2b_3)(2b_{q_1}-b_{q_2}+b_3) + \frac{1}{2}b_{q_2}^2+3b_{q_1}b_3-\frac{3}{2}b_{q_2}b_3. \tag{29c}$$

According to Eq. (26), Eq. (29) is given by

$$y_0 = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2}) -\frac{1}{2}(a_0-a_{q_2})^2-3a_0(a_0-a_{q_1})-\frac{3}{2}a_0(a_0-a_{q_2}), \quad (30a)$$

$$y_{q_1} = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2}) -\frac{3}{2}(a_0-a_{q_2})^2-a_0(a_0-a_{q_1})+\frac{3}{2}a_0(a_0-a_{q_2}), \quad (30b)$$

$$y_{q_2} = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2})-(a_0-a_{q_1})^2 -\frac{5}{2}(a_0-a_{q_2})^2-3a_0(a_0-a_{q_1})+\frac{7}{2}a_0(a_0-a_{q_2}). \quad (30c)$$

Moreover, Eq. (30) can be deformed as

$$y_0 = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2}) -2(a_0-a_{q_2})^2+3a_0(a_{q_1}-a_{q_2})+\frac{3}{2}a_{q_2}(a_0-a_{q_2}), \quad (31a)$$

$$y_{q_1} = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2}) -(a_0-a_{q_2})^2+a_0(a_{q_1}-a_{q_2})+\frac{1}{2}a_{q_2}(a_0-a_{q_2}), \quad (31b)$$

$$y_{q_2} = -\frac{1}{2}(a_0+2a_{q_1}-a_{q_2})(2a_{q_1}-a_{q_2})+(a_{q_1}-a_{q_2})^2 -2(a_0-a_{q_2})^2+3a_0(a_{q_1}-a_{q_2})+\frac{1}{2}a_{q_2}(a_0-a_{q_2}). \quad (31c)$$

When Eq. (31) is calculated with the algorithm as **Alg**. 1, the calculation amount of a squaring in type–$\langle h=2, m=3\rangle$ AOPF is given as **Table** 3.

Algorithm 1: The squaring algorithm efficient for type–$\langle h=2, m=3\rangle$ AOPF

Input: $A=\sum_{i=0}^{2} a_i\gamma^{p^i}$, $a_i \in \mathbb{F}_p$.

Output: $Y=A^2=\sum_{i=0}^{2} y_i\gamma^{p^i}$, $y_i \in \mathbb{F}_p$.

1 $s_0 = a_0 - a_{q_2}$, $s_1 = a_{q_1} - a_{q_2}$.
2 $s_2 = s_1 + a_{q_1}$, $s_3 = s_2 + a_0$.
3 $t_0 = s_2 s_3/2$, $t_1 = s_0 a_{q_2}$, $t_2 = s_1 a_0$.
4 $t_3 = s_0^2$, $t_4 = s_1^2$.
5 $t_5 = -t_0 + t_2 - t_3$, $t_6 = 2t_2$, $t_7 = -t_3 + t_5 + t_6$.
6 $t_8 = t_1/2$, $t_9 = t_1 + t_8$.
7 $y_0 = t_7 + t_9$, $y_{q_1} = t_5 + t_8$, $y_{q_2} = t_4 + t_5 + t_8$

3.2.3 Derivation with PGNB When $h = 4$

Type–$\langle h=4, m=3\rangle$ CVMA calculates y_l in Eq. (17) as

$$y_0 = -2(a_0-a_{q_1})^2-(a_0-a_{q_2})^2-(a_{q_1}-a_{q_2})^2-a_0^2, \quad (32a)$$

$$y_{q_1} = -(a_0-a_{q_1})^2-(a_0-a_{q_2})^2-2(a_{q_1}-a_{q_2})^2-a_{q_1}^2, \quad (32b)$$

$$y_{q_2} = -(a_0-a_{q_1})^2-2(a_0-a_{q_2})^2-(a_{q_1}-a_{q_2})^2-a_{q_2}^2, \quad (32c)$$

where q_1 and q_2 are given by

$$[q_1, q_2] = \begin{cases} [1,2] & (\text{when } \langle\langle p\rangle\rangle = 4,\ 6,\ 7, \text{ or } 9), \\ [2,1] & (\text{when } \langle\langle p\rangle\rangle = 2,\ 3,\ 10, \text{ or } 11), \end{cases} \quad (33)$$

Eq. (32) is expanded as

$$y_0 = -4a_0^2+4a_0a_{q_1}-3a_{q_1}^2+2a_{q_1}a_{q_2}-2a_{q_2}^2+2a_{q_2}a_0, \quad (34a)$$

$$y_{q_1} = -2a_0^2+2a_0a_{q_1}-4a_{q_1}^2+4a_{q_1}a_{q_2}-3a_{q_2}^2+2a_{q_2}a_0, \quad (34b)$$

$$y_{q_2} = -3a_0^2+2a_0a_{q_1}-2a_{q_1}^2+2a_{q_1}a_{q_2}-4a_{q_2}^2+4a_{q_2}a_0. \quad (34c)$$

For Eq. (34), the following matrix is considered with the coefficients.

	a_0^2	$2a_0a_{q_1}$	$a_{q_1}^2$	$2a_{q_1}a_{q_2}$	$a_{q_2}^2$	$2a_{q_2}a_0$
y_0	−4	2	−3	1	−2	1
y_{q_1}	−2	1	−4	2	−3	1
y_{q_2}	−3	1	−2	1	−4	2

(35)

According to Eq. (25), Eq. (34) is given by

$$z_{q_1} = -b_{q_1}^2+2b_{q_1}b_{q_2}-b_{q_2}^2+2b_3b_{q_1}, \quad (36a)$$

$$z_{q_2} = b_{q_1}^2-2b_{q_2}^2+2b_{q_2}b_3, \quad (36b)$$

$$z_3 = 3b_{q_1}^2-2b_{q_1}b_{q_2}+2b_{q_2}^2+b_3^2. \quad (36c)$$

For Eq. (36), the following matrix is considered with the coefficients.

	$b_{q_1}^2$	$2b_{q_1}a_{q_2}$	$b_{q_2}^2$	$2b_{q_2}b_3$	b_3^2	$2b_3b_{q_1}$
z_{q_1}	−1	1	−1	0	0	2
z_{q_2}	1	0	−2	1	0	0
z_3	3	−1	2	0	1	0

(37)

The above matrix also has more non–zero elements than that of Eq. (35). Thus, for example, we can easily consider the deformation of Eq. (36) as

$$z_{q_1} = -(b_{q_1}-b_{q_2})^2+2b_{q_1}b_3, \quad (38a)$$

$$z_{q_2} = b_{q_1}^2-2b_{q_2}(b_{q_2}-b_3), \quad (38b)$$

$$z_3 = (b_{q_1}-b_{q_2})^2+2b_{q_1}^2+b_{q_2}(b_{q_2}-b_3)+b_3(b_{q_2}+b_3). \quad (38c)$$

According to Eq. (26), Eq. (38) is given by

$$y_0 = -2(a_0-a_{q_1})^2-(a_{q_1}-a_{q_2}) -a_0(2a_0-a_{q_2})+a_{q_2}(2a_0-a_{q_2}), \quad (39a)$$

$$y_{q_1} = -2(a_0-a_{q_1})^2-2(a_{q_1}-a_{q_2})+a_0(2a_0-a_{q_1}) -a_0(2a_0-a_{q_2})+a_{q_2}(2a_0-a_{q_2}), \quad (39b)$$

$$y_{q_2} = -(a_0-a_{q_1})^2-(a_{q_1}-a_{q_2}) -a_0(2a_0-a_{q_2})+3a_{q_2}(2a_0-a_{q_2}). \quad (39c)$$

When Eq. (39) is calculated with the algorithm as **Alg**. 2, the calculation amount of a squaring in type–$\langle h=4, m=3\rangle$ AOPF is given as **Table** 2.

Algorithm 2: The squaring algorithm efficient for type–$\langle h=4, m=3\rangle$ AOPF

Input: $A=\sum_{i=0}^{2} a_i\gamma^{p^i}$, $a_i \in \mathbb{F}_p$.

Output: $Y=A^2=\sum_{i=0}^{2} y_i\gamma^{p^i}$, $y_i \in \mathbb{F}_p$.

1 $s_0 = a_0 - a_{q_1}$, $s_1 = a_0 - a_{q_2}$, $s_2 = a_{q_1} - a_{q_2}$.
2 $t_0 = s_0^2$, $t_1 = s_2^2$.
3 $t_2 = s_0 a_0$, $t_3 = s_1 a_0$, $t_4 = s_1 a_{q_2}$.
4 $t_5 = -t_0 - t_1 - t_3 + t_4$, $t_6 = 2t_2$, $t_7 = 2t_4$.
5 $y_0 = -t_0 + t_5$, $y_{q_1} = y_0 - t_1 - t_6$, $y_{q_2} = t_5 - t_7$.

Table 2: The calculation amounts of a squaring in OEF $\mathbb{F}_{p^3}$

algorithm	Schoolbook	Karatsuba multiplication	CH–SQR$_1$	CH–SQR$_2$	CH–SQR$_3$
calculation amount	$(3,3,3,3,0,2)$ †	$(0,6,13,0,0,2)$ †	$(3,2,9,2,0,2)$ †	$(2,3,8,2,0,2)$ †	$(1,4,10,1,1,2)$ †

† (a,b,c,d,e,f) means $aM_1 + bS_1 + cA_1 + dD_1 + eL_1 + fN_1$,
where N_1 denotes the calculation cost of a scalar–n multiplication in $\mathbb{F}_p$.

Table 3: The calculation amounts of a squaring in type–$\langle h,m\rangle$ AOPF $\mathbb{F}_{p^3}$

the parameter h	$h=2$		$h=4$	
algorithm	type–$\langle h,m\rangle$ CVMA	the proposed method	type–$\langle h,m\rangle$ CVMA	the proposed method
calculation amount	$(0,6,9,0,0)$ †	$(3,2,13,1,2)$ †	$(0,6,15,0,0)$ †	$(3,2,10,2,0)$ †

† (a,b,c,d,e) means $aM_1 + bS_1 + cA_1 + dD_1 + eL_1$.

4 CONCLUSIONS

This paper introduced type–$\langle h,m\rangle$ pseudo Gauss period normal basis (PGNB), and derives each squaring algorithm efficient for type–$\langle h=2, m=3\rangle$ and type–$\langle h=4, m=3\rangle$ AOPF with PGNB. For an arbitary characteristic p, type–$\langle h=2, m=3\rangle$ or type–$\langle h=4, m=3\rangle$ AOPF is available by about 88.9% [9]. Thus, for an arbitary characteristic p, the proposed methods shown in **Sec**. 3.2.2 and 3.2.3 can be used by about 88.9%.

REFERENCES

[1] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairings," SCIS 2000, pp. 26–28, 2000.

[2] D. Boneh, X. Boyan, and H. Shacham, "Short group signatures,", Proc. of Crypto2004, LNCS 3152, pp. 41–55, 2004.

[3] H. Hess, N. P. Smart and F. Vercauteren, "The eta pairing re–visited," IEEE Trans. Inf. Theory, Vol. 52, pp. 4595–4602, 2006.

[4] Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa, "Integer Variable χ–based Ate Pairing," Pairing 2008, LNCS vol. 5209, pp. 178–191, 2008.

[5] E. Lee, H. S. Lee, and C. M. Park, "Efficient and Generalized Pairing Computation on Abelian Varieties," ePrint, No. 040, 2008.

[6] D. Bailey and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public–Key Algorithms," Proc. of Asiacrypt2000, LNCS 1976, pp. 248–258, 2000.

[7] R. Dahab, A. J. Devegili, C. Ó hÉigeartaigh, and M. Scott, "Multiplication and Squaring on Pairing–Friendly Fields," ePrint, No. 471, 2006.

[8] J. Chung, and M. A. Hasan, "Asymmetric Squaring Formulae," Technical Report CACR 2006–24, Univ. of Waterloo, 2006.

[9] K. Nekado, Y. Nogami, H. Kato, and Y. Morikawa, "Cyclic Vector Multiplication Algorithm and Existence Probability of Gauss Period Normal Basis," IEICE Trans., Vol. E94–A, No. 1, Jan. 2011, to apear.

[10] S. Gao, "Abelian groups, Gauss periods, and Normal bases," Finite Fields and Their Applications, Vol. 7, pp. 149–164, 2001.

[11] H. Kato, Y. Nogami, and Y. Morikawa, "Fast Squaring in TypeI All One Polynomial Field," ITC–CSCC 2008, pp. 273–275, 2008.

[12] K. Nekado, H. Kato, Y. Nogami, and Y. Morikawa, "Efficient Squaring Algorithm for Xate Pairing with Freeman Curve," Memoirs Faculty Eng. Okayama Univ., Vol. 44, No. 9, pp. 69–72, 2009.

[13] Y. Sakemi, K. Nishii, T. Izuta, Y. Nogami, and Y. Morikawa, "Accelerating Cross Twisted Ate Pairing with Ordinary Pairing Friendly Curve of Composite Order That Has Two Large Prime Factors," TwC 2010, CD–ROM TwC–1–4, 2010.