Cyclic Vector Multiplication Algorithm Based on a Special Class of Gauss Period Normal Basis

Hidehiro Kato, Yasuyuki Nogami, Tomoki Yoshida, and Yoshitaka Morikawa

This paper proposes a multiplication algorithm for F_{p^m}, which can be efficiently applied to many pairs of characteristic p and extension degree m except for the case that $8p$ divides $m(p-1)$. It uses a special class of type-$\langle k, m\rangle$ Gauss period normal bases. This algorithm has several advantages: it is easily parallelized; Frobenius mapping is easily carried out since its basis is a normal basis; its calculation cost is clearly given; and it is sufficiently practical and useful when parameters k and m are small.

Keywords: Extension field, public-key cryptosystem, fast implementation, optimal extension field, optimal normal basis.

I. Introduction

It is quite convenient to scalably change the security level of cryptographies according to the performance of the device or the importance of the information. If, for example, we would like to realize a public key cryptography whose key length is scalably changed, we need to prepare a certain definition field whose arithmetic operations are also scalably carried out. The target of this paper is public key cryptographies [1] and their applications [2] whose definition field is a certain extension field, F_{p^m}, where p and m are the characteristic and extension degree, respectively. As a recent cryptographic application, pairing-based cryptography [1] also needs arithmetic operations in such a large order extension field. For example, it uses a 160-bit prime number and 6 as characteristic and extension degree, respectively [1]. Using a special class of type-$\langle k,m\rangle$ Gauss period normal bases, for which $km+1$ must be a prime number, this paper proposes a multiplication algorithm which can be applied to many pairs of characteristic p and extension degree m except for the following case:

$$8p \mid m(p-1). \tag{1}$$

According to Dirichlet's theorem on arithmetic progressions [3], for an arbitrary positive integer m, there is an infinite number of k's such that $km+1$ becomes a prime number. In the case of (1), according to Gao [4], there exists a Gauss period normal basis in F_{p^m}. The authors also experimentally checked it for many pairs of p and m. In this paper, we only deal with the case that characteristic p is an odd prime number.

Constructing an efficient extension field F_{p^m}, such as an optimal extension field (OEF) [5] or Type I all-one polynomial field (Type I AOPF) [6] generally requires a certain irreducible polynomial of degree m over F_p. For example, since Type I

Manuscript received Feb. 02, 2007; revised Aug. 16, 2007.

This work was supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) from the Ministry of Internal Affairs and Communications of Japan.

Hidehiro Kato (email: kato@trans.cne.okayama-u.ac.jp), Yasuyuki Nogami (phone: + 81 86 251 8128, email: nogami@cne.okayama-u.ac.jp), Tomoki Yoshida (email: yoshida@trans.cne.okayama-u.ac.jp), and Yoshitaka Morikawa (email: morikawa@cne.okayama-u.ac.jp) are with the Department of Communication Network Engineering, Okayama University, Okayama, Japan.

AOPF adopts a certain normal basis in F_{p^m}, characteristic p and extension degree m must satisfy the conditions that m+1 is a prime number and p is a primitive element in F_{m+1}, for which the extension degree m must be even [6]. Moreover, in order to implement fast arithmetic operations, some algorithms, such as the Karatsuba method [7], the cyclic vector multiplication algorithm (CVMA) [6], and the Itoh-Tsujii inversion algorithm [8], will be additionally applied. If we cannot prepare such an extension field, then we will consider an irreducible trinomial as the modular polynomial of F_{p^m}. Of course, an irreducible trinomial does not exist for an arbitrary degree. Moreover, polynomial modulo operations and Frobenius mapping will become more time-consuming compared to those in OEF and Type I AOPF. The multiplication algorithm proposed in this paper is based on CVMA. Type I AOPF [6] and CVMA [6], [9] have the following four advantages: the calculation of CVMA is easily parallelized, Frobenius mapping in Type I AOPF is easily carried out since its basis is a normal basis, its calculation cost is clearly given, and it is sufficiently practical and useful when m is small. However, they have the disadvantage that the extension degree is restricted to be a certain even number, and correspondingly, the characteristic is restricted.

Using a special class of type-$\langle k,m\rangle$ Gauss period normal bases, this paper proposes a multiplication algorithm for F_{p^m} which is applied to many pairs of characteristic p and extension degree m except for the case of (1). This algorithm also has the the previously outlined advantages. Moreover, it can eliminate the disadvantage of extension degree restriction. The main idea is that if we can prepare Type I AOPF $F_{p^{km}}$ with a certain number k, we have the objective extension field F_{p^m} as its subfield. In this paper, we call this subfield F_{p^m} Type I extended AOPF (Type I-X AOPF). In order to consider Type I AOPF $F_{p^{km}}$, we need a positive integer k such that km+1 is a prime number and p is a primitive element in F_{km+1} [6]. We simulated many pairs of p and m. Such a positive integer k always existed except for the case of (1). Thus, we can always prepare Type I-X AOPF F_{p^m} with a normal basis that is given as a special class of Gauss period normal bases [1]. Of course, Frobenius mapping does not need any arithmetic operations. The CVMA in Type I AOPF $F_{p^{km}}$ can be directly applied for its subfield Type I-X AOPF F_{p^m}; however, the calculation cost becomes unnecessarily large corresponding to parameter k. Therefore, the proposed multiplication algorithm is given by modifying the CVMA of Type I AOPF $F_{p^{km}}$ for Type I-X AOPF F_{p^m}. After that, the calculation cost of the proposed algorithm is evaluated and experimental results are shown. These results demonstrate that the proposed algorithm is sufficiently practical and useful when parameters k and m are small.

Throughout this paper, $\#_{SADD}$ and $\#_{SMUL}$ denote the number of additions and the number of multiplications, respectively. In this paper, a subtraction in F_p is counted up as an addition in F_p. The characteristic and extension degree are denoted by p and m, respectively, where p is a prime number, F_{p^m} denotes an m-th extension field over F_p, and $F^*_{p^m}$ denotes the multiplicative group in F_{p^m}. Without any additional explanation, lower and upper case letters show elements in the prime field and extension field, respectively, and a Greek character shows a zero of a modular polynomial.

II. Fundamentals

We briefly discuss extension fields, the Type I AOPF and the CVMA [6].

1. Extension Fields

Some extension fields that have fast arithmetic operations have been previously proposed, such as the OEF [5] and Type I AOPF [6]. To implement fast arithmetic operations, the parameters discussed in this subsection play important roles.

A. Modular Polynomial

In general, constructing an extension field F_{p^m} requires an irreducible polynomial of degree m over F_p. Using this irreducible polynomial as the modular polynomial, arithmetic operations such as multiplication are carried out. In particular, it is said that binomials, trinomials, and all-one polynomials[1)] are efficient for fast arithmetic operations.

In order to prepare a certain irreducible polynomial, although irreducible binomials, trinomials, and all-one polynomials can be easily obtained [5], [10], [11], several irreducibility tests are generally needed until an irreducible polynomial is obtained. The irreducibility test becomes more time-consuming as characteristic p and extension degree m become larger. In addition, an irreducible binomial, trinomial, and the all-one polynomial of degree m over F_p do not exist for an arbitrary pair of p and m. For example, an irreducible binomial of degree m over F_p exists if, and only if, each prime factor of m divides $p-1$ and $4\,|\,(p-1)$ when $4\,|\,m$. The well-known OEF adopts an irreducible binomial as the modular polynomial [5].

B. Basis

The extension field F_{p^m} can be considered as a vector space of degree m over F_p. We can pick up m linearly independent elements in F_{p^m} as a basis. Polynomial bases and normal bases are well-known [11]. For example, a normal basis is efficient for Frobenius mapping, $A \to A^p$, and a polynomial basis is efficient for vector multiplication. An optimal normal

1) A polynomial whose coefficients are all one is called an *all-one polynomial*. For example, $(x^{m+1}-1)/(x-1)$.

basis (ONB) has efficiencies of both the normal basis and polynomial basis [12]. A normal basis in F_{p^m} consists of m conjugate elements of a certain proper element in F_{p^m}; however, not every set of conjugate elements in F_{p^m} forms a normal basis [11]. Therefore, when we would like to use a normal basis, we generally need to check whether the conjugate elements form a normal basis. The well-known Type I and Type II ONBs can be easily obtained; however, these useful normal bases exist only when the extension degree m is a certain even number and a certain number, respectively [6], [9].

C. Algorithm

Among fundamental arithmetic operations in the extension field, multiplication and inversion are especially time-consuming. Therefore, for quick calculation, some algorithms, such as the Karatsuba method [7] and Itoh-Tsujii inversion algorithm [8], are applied. However, the Karatsuba method requires a polynomial basis, and the Itoh-Tsujii algorithm and Avanzi's exponentiation method require fast Frobenius mapping [13]. It is not easy to satisfy both requirements. The OEF can satisfy them [5]; however, it is also restricted by other conditions, such as characteristic p and extension degree m as discussed in section II.1.A.

2. Type I All-One Polynomial Field

The Type I AOPF is an extension field F_{p^m} whose extension degree must be a certain even number [6]. The Type I AOPF adopts the following modular polynomial and basis to implement fast arithmetic operations.

Modular polynomial: all-one polynomial

$$(x^{m+1}-1)/(x-1), \tag{2}$$

where it must be irreducible over F_p.
Basis: a pseudo-polynomial basis

$$\{\omega, \omega^2, \cdots, \omega^{m-1}, \omega^m\}, \tag{3}$$

where ω is a zero of the modular polynomial.

The pseudo polynomial basis (3) is equivalent to the following normal basis:

$$\{\omega, \omega^p, \omega^{p^2}, \cdots, \omega^{p^{m-1}}\}. \tag{4}$$

When p=2, it is specifically called a Type I ONB. It is efficient for fast arithmetic operations in an extension field. In the remainder of this paper, we call this normal basis (4) a Type I ONB. The following conditions must be satisfied by $(x^{m+1}-1)/(x-1)$ to be irreducible over F_p. First, m+1 must be a prime number, and secondly, p must be a primitive element in F_{m+1}. Accordingly, the extension degree m must be an even number; therefore, Type I AOPFs cannot have odd prime extension degrees. In Type I AOPF, we calculate a multiplication and inversion by a cyclic vector multiplication algorithm and the Itoh-Tsujii inversion algorithm,[2)] respectively [8].

3. Cyclic Vector Multiplication Algorithm

CVMA [6] uses the following two relations:

$$\omega^{m+1} = 1, \quad \omega + \omega^2 + \cdots + \omega^m = -1, \tag{5}$$

where ω is a zero of $(x^{m+1}-1)/(x-1)$. That is, the modular polynomial of Type I AOPF and the basis (3) consist of m conjugate elements of ω as shown in (4). Let us consider two vectors X and Y in F_{p^m}, which are represented by (3) as

$$X = (x_1, x_2, \cdots, x_m), \ Y = (y_1, y_2, \cdots, y_m), \tag{6}$$

where $x_i, y_i \in F_p$, and $m \ge i \ge 1$.
Suppose the product Z of X and Y as

$$Z = XY = (z_1, z_2, \cdots, z_m), \tag{7}$$

where $z_i \in F_p$ ($m \ge i \ge 1$). Noting that m is even because m+1 is a prime number larger than 2, according to CVMA [6], we calculate

$$q_i = \sum_{s=1}^{m/2} \left\{ \left(x_{\langle 2^{-1}i+s \rangle} - x_{\langle 2^{-1}i-s \rangle} \right) \cdot \left(y_{\langle 2^{-1}i+s \rangle} - y_{\langle 2^{-1}i-s \rangle} \right) \right\}, \tag{8a}$$

where $m \ge i \ge 0$. Then, we have the coefficient z_i as

$$z_i = q_0 - q_i, \ m \ge i \ge 1, \tag{8b}$$

where the subscript $\langle\cdot\rangle$ means $\cdot$ mod (m + 1). When extension degree m = 4, CVMA calculates

$$q_0 = (x_1 - x_4)(y_1 - y_4) + (x_2 - x_3)(y_2 - y_3), \tag{9a}$$

$$z_1 = q_0 - \{(x_2 - x_4)(y_2 - y_4) + x_1 y_1\}, \tag{9b}$$

$$z_2 = q_0 - \{(x_3 - x_4)(y_3 - y_4) + x_2 y_2\}, \tag{9c}$$

$$z_3 = q_0 - \{(x_1 - x_2)(y_1 - y_2) + x_3 y_3\}, \tag{9d}$$

$$z_4 = q_0 - \{(x_1 - x_3)(y_1 - y_3) + x_4 y_4\}. \tag{9e}$$

2) Since AOPF adopts a normal basis, Frobenius mapping does not require any arithmetic operations.

From (8a) and (9), we find that the terms $x_l y_l$, $1 \le l \le m$ and $(x_i - x_j)/(y_i - y_j)$, $1 \le i < j \le m$ appear in the calculations of $q_{\langle l \rangle}$ and $q_{\langle i+j \rangle}$, respectively. It should be noted that CVMA in Type I AOPF adopts the pseudo-polynomial basis (3).

Compared to the Karatsuba-based multiplication [5], [7], the calculation cost for CVMA in Type I AOPF F_{p^m} can be clearly evaluated as given in [6] as

$$\#_{\text{SMUL}} = \frac{m(m+1)}{2}, \ \#_{\text{SADD}} = \frac{3m^2 - m - 2}{2}, \tag{10}$$

because CVMA is based on (8). In the Itoh-Tsujii inversion algorithm, Frobenius mapping $A \to A^p$ is required several times. If the extension field adopts a normal basis such as Type I AOPF, Frobenius mapping does not require any arithmetic operations [6]. The calculation cost of the Karatsuba-based multiplication is evaluated as $\#_{\textbf{SMUL}} = m^{\log_2 3}$ [7].

4. Problems in Previous Works

Most efficient extension fields F_{p^m}, such as OEF and Type I AOPF, restrict the modular polynomial by which the arithmetic operations can be quickly carried out. Accordingly, characteristic p and extension degree m are also restricted. Granger and others [14] proposed an efficient multiplication in an extension field; however, it works only when extension degree m is divisible by 6. Even if we have an efficient multiplication algorithm and software library, they are custom-designed for the objective extension field in general; therefore, it cannot be used for another extension field.[3] These restrictions narrow the efficiency and versatility of cryptographic applications.

Avanzi and others [13] introduced processor adequate finite fields (PAFFs) focused on the odd characteristic $p<2^w$, where w is some processor related word length. Arithmetic operations in extension field F_{p^m} can be implemented by using integer operations within the word length. Moreover, [13] focuses on exponentiations. Some cryptographic applications require several exponantiations over the extension field, and the exponents become quite large numbers, for which [13] recommends the use of p-adic representation and Frobenius mapping. The p-adic representation also contributes to keeping within the word length. Then, if we need little calculation for Frobenius mapping, the exponentiations can be quickly carried out. As introduced in [13], OEF is one of the most efficient PAFFs because Frobenius mapping is quickly carried out [5]. This paper picks up OEF as the competitor but does not restrict the characteristic within the word length.

3) For another extension field, we need another multiplication program in order to be similarly efficient.

III. Type I-X All-One Polynomial Field

As described in sections I and II, the well-known extension fields F_{p^m} OEF and AOPF are restricted by characteristic p, extension degree m, and the modular polynomial. In this section, using a special class of type-$\langle k,m \rangle$ Gauss period normal bases, we present a multiplication algorithm that can be applied to many pairs of characteristic p and extension degree m except for the case of (1), in which 8p divides $m(p$-1). It is efficient when k and m are small.

1. Main Idea

Let the objective extension field be the m-th extension field F_{p^m} over the prime field F_p. If we can prepare the extension field $F_{p^{km}}$ as a Type I AOPF with a certain number k, as shown in Fig. 1, we obtain the objective F_{p^m} as its subfield. In addition, we can use CVMA.

Here, we will briefly discuss the type-$\langle k,m \rangle$ Gauss period normal basis (GNB) defined as in [15] as follows.

Definition 1. Let km+1 be a prime number not equal to p. Suppose that gcd($km/e,m$) = 1, where e is the order of p modulo km+1. Then, for any primitive k-th root θ of the unity in F_{km+1},

$$\gamma = \sum_{i=0}^{k-1} \beta^{\theta^i} \tag{11}$$

generates a normal basis $\{\gamma, \gamma^p, \cdots, \gamma^{p^{m-1}}\}$ in F_{p^m}, where β is a zero of $(x^{km+1}-1)/(x-1)$. We call this normal basis type-<k, m> Gauss period normal basis.

The following Type I eXtended normal basis (Type I-X NB) is a special class of Gauss period normal bases.

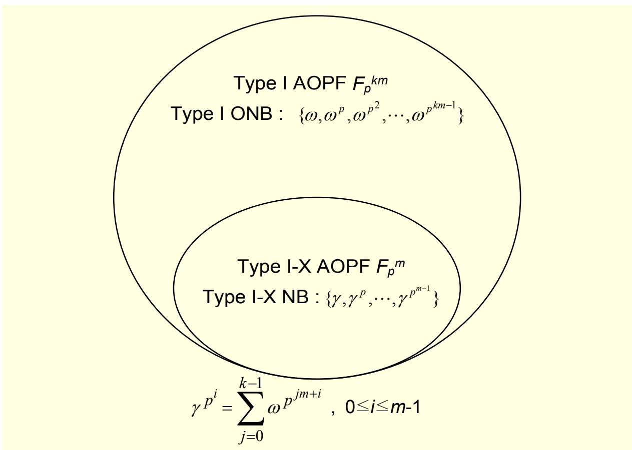


Fig. 1. Image of the main idea.

2. Definition of Type I-X AOPF

We consider an extension field defined as follows.

Modular polynomial: all-one polynomial

$$(x^{km+1}-1)/(x-1), \tag{12}$$

where it must be irreducible over F_p.
Basis: a normal basis

$$\{\gamma, \gamma^p, \cdots, \gamma^{p^{m-1}}\}, \tag{13a}$$

where γ is defined by

$$\gamma^{p^i} = \sum_{u=0}^{k-1} \omega^{p^{i+um}}, \quad 0 \le i \le m-1, \tag{13b}$$

where ω is a zero of the modular polynomial.

The relation of ω and γ is shown in Fig. 1. In the remainder of this paper, we call the extension field defined here a Type I-X AOPF. In addition, we call the normal basis (13a) Type I-X NB. It is a special class of type-$\langle k,m \rangle$ Gauss period normal bases. Many studies about GNB have been carried out [16], [17]. Gao [4] discussed the normal basis and the self dual normal basis in detail. Nöcker [16] discussed how to efficiently implement arithmetic operations in an extension field with GNB.

To prepare Type I-X NB (13a) as a special class of Gauss period normal bases, the modular polynomial $(x^{km+1}-1)/(x-1)$ needs to be irreducible over F_p. In other words, this paper only considers the case for which the following two conditions are satisfied: 1) $km + 1$ is a prime number, and 2) p is a primitive element in F_{km+1}. Of course, parameter k is closely related to the calculation cost; therefore, it is preferable for k to be the smallest among many k's that satisfies conditions 1 and 2. We consider these conditions because property 1 is shown based on the primitivity of p in F_{km+1}. Accordingly, the proposed algorithm shown in Fig. 2 efficiently uses the primitivity.

Input: $X = \sum_{i=0}^{m-1} x_i \gamma^{p^i}$, $Y = \sum_{i=0}^{m-1} y_i \gamma^{p^i}$.

Output: $Z = XY = \sum_{i=0}^{m-1} z_i \gamma^{p^i}$

Preparation:

1. Determine k such that Type I AOPF $F_{p^{km}}$ exists.
2. For $0 \le i \le m$, $q[i] \leftarrow 0$.
3. For $0 \le t \le m-1$ and $0 \le h \le k-1$, $g[\langle p^{t+hm} \rangle] \leftarrow t + 1$.
4. $g[0] \leftarrow 0$.

Procedure:

For $0 \le i \le m-1$, $q[i + 1] \leftarrow x_i y_i$.
For $0 \le i < j \le m-1$, {
  $M \leftarrow (x_i - x_j)(y_i - y_j)$,
  For $0 \le h \le k-1$, {
    $q[g[\langle p^i + p^{j+hm} \rangle]] \leftarrow q[g[\langle p^i + p^{j+hm} \rangle]] + M$.
  }
}
For $0 \le i \le m-1$, $z_i \leftarrow kq[0]-q[i + 1]$.

(End of algorithm)

Fig. 2. Modified CVMA for Type I-X AOPF F_{p^m}.

■ *Vector Representation of an Element in* F_{p^m}

Consider an arbitrary element X in Type I-X AOPF F_{p^m} represented with Type I-X NB in F_{p^m} as

$$X = \sum_{i=0}^{m-1} x_i \gamma^{p^i} = (x_0, x_1, x_2, \cdots, x_{m-1}), \; x_i \in F_p . \tag{14}$$

Noting that $\gamma = \sum_{j=0}^{k-1} \omega^{p^{jm}}$, we also represent X in Type I-X AOPF F_{p^m} with Type I ONB in $F_{p^{km}}$ as

$$\begin{aligned} X &= \sum_{i=0}^{m-1} x_i \left(\omega + \omega^{p^m} + \cdots + \omega^{p^{(k-1)m}}\right)^{p^i} \\ &= (x_0, x_1, x_2, \cdots, x_{m-1}, \\ &\quad\; x_0, x_1, x_2, \cdots, x_{m-1}, \\ &\quad\; \vdots \\ &\quad\; x_0, x_1, x_2, \cdots, x_{m-1}). \end{aligned} \tag{15}$$

Here, we use both vector representations of (14) and (15).

3. Remarks

According to Gao [4] and the following remarks, we can construct the extension field F_{p^m} as Type I-X AOPF for many pairs of characteristic p and extension degree m except for the case of (1).

Remark 1. For an arbitrary extension degree m, there is an infinite number of k's such that km+1 becomes a prime number. It is well-known as the Dirichlet's theorem on arithmetic progressions [3].

Remark 2. From many experimental results, except for the case of (1), there exist positive integer k's such that 1) km+1 is a prime number and 2) p is a primitive element in F_{km+1} [4).]

Remark 3. When (1) is satisfied, there is no positive integer k that satisfies 1) km+1 is a prime number and 2) p is a primitive element in F_{km+1}.

Proof (Remark3). Let km+1 be a prime number and $8p$ divide $m(p$–1). Consider the primitivity of the element p in F_{km+1}. Using Legendre symbol (a/b) and the well-known quadratic reciprocity law [18], we have

$$\left(\frac{p}{km+1}\right) = (-1)^{\frac{km(p-1)}{4}} \left(\frac{km+1}{p}\right) = \left(\frac{1}{p}\right), \tag{16}$$

4) We examined many pairs of p and m; however, there were no counter examples. Therefore, it will be available for almost every pair of p and m except for the case of (1). There are $\phi(km)$ primitive elements in F^*_{km+1}, where $\phi(\cdot)$ is the Euler's function.

where it is noted that $8p$ divides $m(p-1)$. Consequently, p is not a primitive element in F_{km+1}. □

4. Original CVMA for Type I-X AOPF

If CVMA in Type I AOPF $F_{p^{km}}$ is applied in the multiplication of elements in Type I-X AOPF F_{p^m} as it is, the calculation cost becomes unnecessarily large. For example, from (10), the number of F_p-multiplications required for a multiplication in $F_{p^{km}}$ becomes

$$\#_{\mathbf{SMUL}} = km(km+1)/2. \quad (17)$$

The appropriate cost for Type I-X AOPF F_{p^m} will be

$$\#_{\mathbf{SMUL}} = m(m+1)/2. \quad (18)$$

Next, we modify the original CVMA in Type I AOPF $F_{p^{km}}$ to be efficiently applied in its subfield Type I-X AOPF F_{p^m}. We consider a multiplication of two elements in Type I-X AOPF F_{p^m} by modifying the CVMA in Type I AOPF $F_{p^{km}}$.

■ *Modification of CVMA for Type I-X AOPF*

$$X = \sum_{i=0}^{m-1} x_i \gamma^{p^i} = \sum_{i=0}^{m-1}\sum_{u=0}^{k-1} x_i \omega^{p^{i+um}} = \sum_{i=0}^{m-1}\sum_{u=0}^{k-1} x_i \omega^{\langle p^{i+um}\rangle}, \quad (19a)$$

$$Y = \sum_{j=0}^{m-1} y_j \gamma^{p^j} = \sum_{j=0}^{m-1}\sum_{v=0}^{k-1} y_j \omega^{p^{j+vm}} = \sum_{j=0}^{m-1}\sum_{v=0}^{k-1} y_j \omega^{\langle p^{j+vm}\rangle}, \quad (19b)$$

$$\begin{aligned}
XY &= \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} x_i y_j \gamma^{p^i+p^j} \\
&= \sum\sum_{0\le i<j\le m-1}\left(x_i y_j + x_j y_i\right)\gamma^{p^i+p^j} + \sum_{i=0}^{m-1} x_i y_i \gamma^{2p^i} \\
&= -\sum\sum_{0\le i<j\le m-1}\left\{\left(x_i - x_j\right)\left(y_i - y_j\right) - x_i y_i - x_j y_j\right\}\gamma^{p^i+p^j} \\
&\quad + \sum_{i=0}^{m-1} x_i y_i \gamma^{2p^i} \\
&= -\sum\sum_{0\le i<j\le m-1}\left(x_i - x_j\right)\left(y_i - y_j\right)\gamma^{p^i+p^j} \\
&\quad + \sum\sum_{0\le i<j\le m-1}\left(x_i y_i - x_j y_j\right)\gamma^{p^i+p^j} + \sum_{i=0}^{m-1} x_i y_i \gamma^{2p^i},
\end{aligned} \quad (20)$$

$$\begin{aligned}
&\sum_{0\le i<}\sum_{j\le m-1}(x_i y_i + x_j y_j)\gamma^{p^i+p^j} \\
&= \sum_{0\le i<}\sum_{j\le m-1} x_i y_i \gamma^{p^i+p^j} + \sum_{0\le i<}\sum_{j\le m-1} x_j y_j \gamma^{p^i+p^j} = \sum_{i=0}^{m-1}\left(x_i y_i \gamma^{p^i}\sum_{0\le j\le m-1, i\ne j}\gamma^{p^j}\right),
\end{aligned} \quad (21)$$

$$\begin{aligned}
XY &= -\sum\sum_{0\le i<j\le m-1}\left(x_i - x_j\right)\left(y_i - y_j\right)\gamma^{p^i+p^j} \\
&\quad + \sum_{i=0}^{m-1}\left(x_i y_i \gamma^{p^i}\sum_{0\le j\le m-1, i\ne j}\gamma^{p^j}\right) + \sum_{i=0}^{m-1} x_i y_i \gamma^{p^i+p^j} && (22\text{a}) \\
&= -\sum\sum_{0\le i<j\le m-1}\left(x_i - x_j\right)\left(y_i - y_j\right)\gamma^{p^i+p^j} \\
&\quad + \sum_{i=0}^{m-1} x_i y_i\left(\gamma^{p^i}\sum_{0\le j\le m-1, i\ne j}\gamma^{p^j} + \gamma^{p^i}\right) && (22\text{b}) \\
&= -\sum_{0\le i<}\sum_{j\le m-1}(x_i - x_j)(y_i - y_j)\gamma^{p^i+p^j} + \sum_{i=0}^{m-1} x_i y_i \gamma^{p^i}\sum_{j=0}^{m-1}\gamma^{p^j} && (22\text{c}) \\
&= -\left(\sum\sum_{0\le i<j\le m-1}\left(x_i - x_j\right)\left(y_i - y_j\right)\gamma^{p^i+p^j} + \sum_{i=0}^{m-1} x_i y_i \gamma^{p^i}\right). && (22\text{d})
\end{aligned}$$

Consider the multiplication of two elements X and Y in Type I-X AOPF F_{p^m} shown in (19), where $\langle\cdot\rangle$ denotes $\cdot$ mod $km+1$. This multiplication is calculated as (20). Using (21), we have (22). From (22c) to (22d), we use the following relation:

$$\sum_{j=0}^{m-1}\gamma^{p^j} = \sum_{j=0}^{m-1}\sum_{v=0}^{k-1}\omega^{p^{j+vm}} = -1. \quad (23)$$

Here, we use the following property.

Property 1.

$$\gamma^{p^i+p^j} = \sum_{h=0}^{k-1}\gamma^{p^{g(i,j,h)}},\ 0 \le i < j \le m-1. \quad (24\text{a})$$

$\gamma^{p^{g(i,j,h)}}$ is given as

$$\gamma^{p^{g(i,j,h)}} = \begin{cases} -k\left(\gamma^{p^{m-1}} + \gamma^{p^{m-2}} + \cdots + \gamma^{p} + \gamma\right), \text{when} \left\langle p^i + p^{j+hm}\right\rangle = 0, \\ \gamma^{p^t} \text{ such that } \left\langle p^t\right\rangle = \left\langle p^i + p^{j+hm}\right\rangle, \text{otherwise}. \end{cases} \quad (24\text{b})$$

See appendix for its proof. Based on (22d) and the above property, we propose a multiplication algorithm in Type I-X AOPF F_{p^m} as shown in Fig. 2. In the algorithm, lines 1 and 5 correspond to the calculation of (22d). Multiplying the scalar k at line 8 corresponds to the former condition of (24b). In property 1 and the proposed algorithm shown in Fig. 2, the primitivity of p in F_{km+1} is efficiently used.

Unlike the algorithms in [16] and [17], our proposed algorithm is quite simple; therefore, the calculation cost is clearly given. Moreover, it is adaptable enough for changing characteristic p and extension degree m. In particular, when the extension degree m is small, it is quite efficient.

IV. Cost Evaluation and Comparison

1. Cost Evaluation

As shown in Fig. 2, the proposed algorithm requires the calculation of the indexes such as $\langle p^i+p^{j+hm}\rangle$; however, the indexes can be computed prior to calculation when the extension degree m is small. Then, we can directly write the program with the calculated indexes. Thus, the calculation cost for these indexes is not taken into account in this paper.

According to section III.4 and Fig. 2, the proposed algorithm requires the following calculation cost:

$$\#_{\mathrm{SMUL}} = \frac{m(m+1)}{2} + 1, \tag{25a}$$

$$\#_{\mathrm{SADD}} = \frac{m(m-1)(k+2)}{2} + m. \tag{25b}$$

The "+1" in (25a) corresponds to $kq[0]$ at line 8 in the proposed algorithm. When k=1, this multiplication is not needed. In addition, when k is small, $kq[0]$ can be calculated with (k–1) additions. For example, $3q[0]=q[0]+q[0]+q[0]$. In Table 1, $\#_{\mathrm{SADD}}$ is evaluated with such additions.

Compared to (10), $\#_{\mathrm{SMUL}}$ is almost the same, and $\#_{\mathrm{SADD}}$ is about k times larger. As previously discussed, it is preferable for parameter k to be small. When k=1, it is a Type I AOPF [6], and when k=2, it is a Type II AOPF [9].

2. Comparison

We checked the smallest parameter k for 10,000 160-bit prime numbers as characteristic p, in which the extension degree m was fixed at 6. From the experimental result, the average of the smallest k was 3.73. Moreover, for about 70% of the prime numbers, k was equal to or less than 3. Therefore, we consider $k \le 3$. For example, let us consider the case in which the modular polynomial is an irreducible trinomial:

$$x^6 + ax + b, \quad a, b \in F_p. \tag{26}$$

Using the Karatsuba method [5], the calculation cost for a multiplication with the modular polynomial (26) becomes

$$\#_{\mathrm{SMUL}} = 28, \quad \#_{\mathrm{SADD}} = 69. \tag{27}$$

On the other hand, when k=3, that in Type I-X AOPF F_{p^6} needs

$$\#_{\mathrm{SMUL}} = 21, \quad \#_{\mathrm{SADD}} = 83. \tag{28}$$

Table 1 shows the comparison of the calculation cost required for a multiplication in F_{p^m}. Since it is necessary in Type I-X AOPF F_{p^m} for $km+1$ to be a prime number, as shown in the table, there is no data when k and m are both odd numbers. From this comparison, we find that Type I-X AOPF achieves an efficiency almost as high as that of OEF. Moreover, Type I-X AOPF F_{p^m} can be constructed for many pairs of characteristic p and extension degree m except for the case of (1).

Table 2 shows the average computation time for a multiplication in Type I-X AOPF F_{p^m}. We used Pentium 4 (3.6 GHz), C++ programming language, and NTL [19]. As characteristic p, we used a 160-bit prime number such that there were irreducible trinomials, and OEF and Type II OEF could be constructed. From Table 2, it can be concluded that Type I-X AOPF is practical enough for small extension degrees and small k.

Table 1. Comparison of the calculation cost for a multiplication in F_{p^m}.

Extension field F_{p^m}		Extension degree m: 3	4	5	6
General type OEF		(8,15)	(12,27)	(19,42)	(23,64)
Type II OEF		(6,17)	(9,30)	(15,46)	(15,46)
Irreducible trinomial		(10,17)	(15,30)	(23,46)	(28,69)
Type I-X AOPF	k=1	-	(10,22)	-	(21,51)
	k=2	(6,16)	-	(15,46)	(21,67)
	k=3	-	(10,36)	-	(21,83)

Remark : From the left hand side in the parenthesis, the numbers show #SMUL and #SADD, respectively.

Table 2. Computation time for a multiplication in F_{p^m} (μs).

Extension field F_{p^m}		Extension degree m: 3	4	5	6
General type OEF		6.37	9.77	15.6	20.2
Type II OEF		6.01	9.16	14.9	19.2
Irreducible trinomial		7.50	11.5	17.9	22.8
Type I-X AOPF	k=1	-	9.65	-	20.7
	k=2	5.94	-	15.4	21.6
	k=3	-	10.6	-	22.6

Remark : The authors used Pentium 4 (3.6 GHz), C++ programming language, and NTL [19]. The characteristic p was a 160-bit prime.

3. Application

In general, it is said that the security level of public key cryptography increases as the size of the definition field increases[5]. If we can easily and seamlessly change the size of the definition field, we can realize variable key-length public key cryptography. Type I-X AOPF is useful for this. For example, fix characteristic p to a certain 32-bit prime

5) Of course, if there are any other conditions from the viewpoint of security, the definition field should be selected such that those conditions are satisfied.

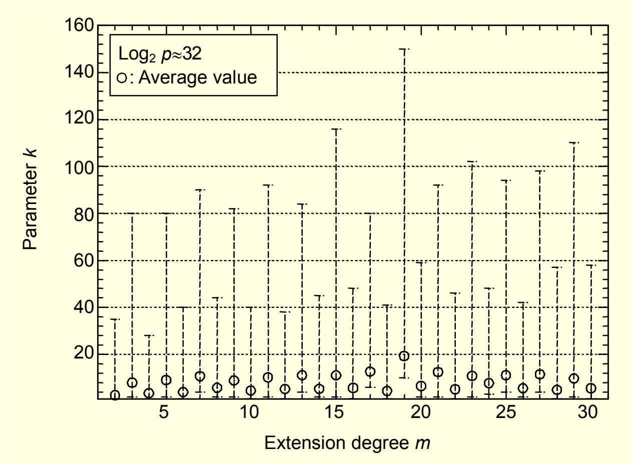


Fig. 3. Maximum, minimum, and average of the smallest k's for Type I-X NB in F_{p^m} when $\log_2 p \approx 32$.

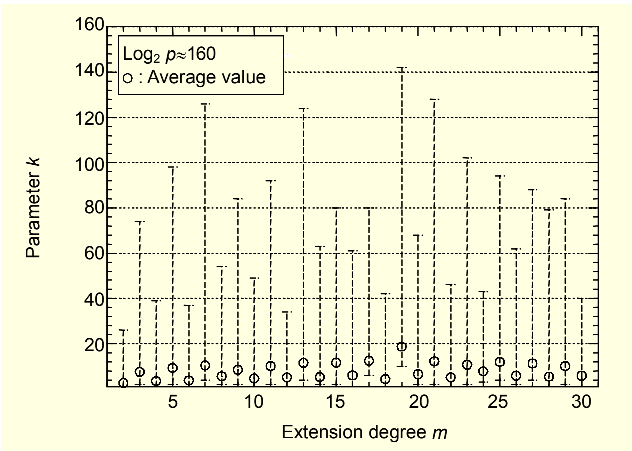


Fig. 4. Maximum, minimum, and average of the smallest k's for Type I-X NB in F_{p^m} when $\log_2$ p ≈160.

number, then change the extension degree m. As previously shown, it is easy to change the extension degree m for the proposed algorithm in Fig. 2, though parameter k will be correspondingly changed. By changing m, the size of the definition field F_{p^m} is changed. Accordingly the key-length is changed. When the size of characteristic p is 32 bits, we can change the key-length by every 32 bits. Some conventional methods such as OEF and irreducible trinomial-based extension fields cannot be easily treated in this way. When we use extension fields for elliptic curve cryptography, we must pay attention to several attacks, such as FR reduction [20] and the Weil descent attack [21].

On the other hand, for the proposed method, it is preferable for k and m to be small. It is especially preferable for parameter k to be small because $\#_{SADD}$ depends on k as shown in (25b). For 1,000 32-bit prime numbers as characteristic p, the authors measured the maximum, minimum, and average of parameter

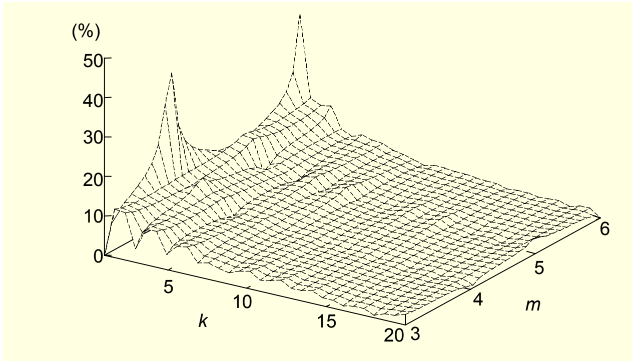


Fig. 5. Distribution of the smallest k's for Type I-X NB in F_{p^m} when $m = 3, 4, 5, 6$ and $\log_2 p \approx 160$.

k's such that Type I-X NB exists in F_{p^m}. Figure 3 shows the result. Figure 4 shows the result for 1,000 160-bit prime numbers, and Fig. 5 shows the distribution of the smallest k's for 10,000 160-bit prime numbers. As discussed in section I, pairing-based cryptography uses a 160-bit prime number and 6 as characteristic and extension degree [1], respectively. When $\log_2 p \approx 160$ and m=6, the maximum, minimum, and average were 37, 1, and 3.73, respectively. Moreover, for about 70% of the prime numbers, k was equal to or less than 3. Thus, the proposed method is useful for scalably changing m; however, it should be noted that the calculation cost increases as k and m increase.

V. Conclusion

This paper proposed a multiplication algorithm for F_{p^m} which was efficiently applied to many pairs of characteristic p and extension degree m except for the case in which $8p$ divided $m(p-1)$. This algorithm uses a special class of type-$\langle k, m\rangle$ Gauss period normal bases and has several advantages. It is easily parallelized, Frobenius mapping is easily carried out since its basis is a normal basis, its calculation cost is clearly given, and it is sufficiently practical and useful when k and m are small. As a future work, we would like to develop a multiplication algorithm that can support all kinds of Gauss period normal bases.

Appendix. Proof of Property 1

Let $0 \le i < j \le m-1$. According to the relation between γ and ω, we have

$$\gamma^{p^i+p^j} = \sum_{u=0}^{k-1}\omega^{p^{i+um}} \cdot \sum_{v=0}^{k-1}\omega^{p^{j+vm}} = \sum_{u=0}^{k-1}\sum_{v=0}^{k-1}\omega^{p^{i+um}+p^{j+vm}}. \quad (A1)$$

If we set $h=v-u$, we have

$$\gamma^{p^i+p^j} = \sum_{u=0}^{k-1}\sum_{h=0}^{k-1}\left(\omega^{p^i+p^{j+hm}}\right)^{p^{um}}. \tag{A2}$$

When $\left\langle p^i + p^{j+hm}\right\rangle = 0$,

$$\sum_{u=0}^{k-1}\left(\omega^{p^i+p^{j+hm}}\right)^{p^{um}} = \sum_{u=0}^{k-1}\omega^0 = k. \tag{A3}$$

Since $\gamma^{p^{m-1}} + \gamma^{p^{m-2}} + \cdots + \gamma^p + \gamma = -1$, we have

$$k = -k\left(\gamma^{p^{m-1}} + \gamma^{p^{m-2}} + \cdots + \gamma^p + \gamma\right). \tag{A4}$$

On the other hand, when $\left\langle p^i + p^{j+hm}\right\rangle \neq 0$, since p is a primitive element in F_{km+1}, we can uniquely determine $0 \leq t \leq m-1$, which satisfies the following relation:

$$\sum_{u=0}^{k-1}\left(\omega^{p^i+p^{j+hm}}\right)^{p^{um}} = \sum_{u=0}^{k-1}\left(\omega^{p^t}\right)^{p^{um}} \tag{A5a}$$

$$= \sum_{u=0}^{k-1}\omega^{p^{t+um}}. \tag{A5b}$$

From (13b),

$$\sum_{u=0}^{k-1}\omega^{p^{t+um}} = \gamma^{p^t}. \tag{A6}$$

It is noted that the parameter t is uniquely determined from i, j, and h. Consequently, we have property 1.

References

[1] H. Cohen and G. Frey, *Handbook of Elliptic and Hyperelliptic Curve Cryptography, Discrete Mathematics and Its Applications,* Chapman & Hall CRC, 2005, pp. 280-285, p. 458.

[2] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," *Proc. Crypto*, LNCS 2442, 2002, pp. 354-368.

[3] T. Apostol, *Introduction to Analytic Number Theory*, Springer-Verlag, 1976.

[4] S. Gao, "Normal Bases over Finite Fields," Doctoral thesis, Waterloo, Ontario, Canada, 1993.

[5] D. Bailey and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms," *Proc. Crypto.*, LNCS 1462, 1998, pp. 472-485.

[6] Y. Nogami, A. Saito, and Y. Morikawa, "Finite Extension Field with Modulus of All-One Polynomial and Representation of Its Elements for Fast Arithmetic Operations," *IEICE Trans.*, vol. E86-A, no. 9, 2003, pp.2376-2387.

[7] D. Knuth, *The Art of Computer Programming*, *vol. 2, Semi-numerical Algorithms*, Addison-Wesley, 1981.

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

[9] Y. Nogami, S. Shinonaga, and Y. Morikawa, "Fast Implementation of Extension Fields with Type II ONB and Cyclic Vector Multiplication Algorithm," *IEICE Trans. Fundamentals*, vol. E88-A, no. 5, 2005, pp. 1200-1208.

[10] T. Sugimura and Y. Suetsugu, "Consideration on Cyclotomic Polynomials," *Trans. IEICE*, vol. J73-A, 1990, pp. 1929-1935.

[11] R. Lidl and H. Niederreiter, *Finite Fields*, *Encyclopedia of Mathematics and Its Applications*, Cambridge University Press, 1984.

[12] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, LNS 265, Cambridge Univ. Press, 1999.

[13] R. Avanzi and P. Mihailescu, "Generic Efficient Arithmetic Algorithms for PAFFs (Processor Adequate Finite Fields) and Related Algebraic Structures," *Proc. SAC*, LNCS 3006, Springer-Verlag, LNCS, 2003, pp. 320-334.

[14] R. Granger, D. Page, and N.P. Smart, "High Security Pairing-Based Cryptography Revisited," available at http://eprint.iacr.org/2006/059.pdf, 2006.

[15] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[16] M. Nöcker, "Data Structures for Parallel Exponentiation in Finite Fields," available at http://deposit.ddb.de/, 2001.

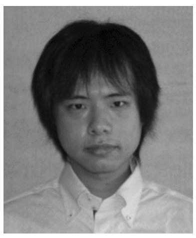
[17] S. Gao, J. Gathen, D. Panario, and V. Shoup, "Algorithms for Exponentiation in Finite Fields," *J. Symb. Comput*. vol. 29, no. 6, 2000, pp. 879-889.

[18] E. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.

[19] A Library for doing Number Theory, http://www.shoup.net/ntl/

[20] G. Frey, M. Muller, and H.G. Ruck, "The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems," *IEEE Trans. Inform. Theory*, vol. 45 no. 5, 1999, pp. 1717-1719.

[21] S.D. Galbraith and N.P. Smart, "A Cryptographic Application of Weil Descent," *Proc. of Cryptography and Coding*, Springer LNCS 1746, 1999, pp. 191-200.



Hidehiro Kato graduated from Okayama University in 2005 and obtained the MS degree in 2006. He is now a doctoral candidate of the Graduate School of Natural Science and Technology, Okayama University. He is studying finite field theory, especially the implementation of fast arithmetic operations in a finite field. He is a member of IEICE.

Yasuyuki Nogami graduated from Shinshu University in 1994 and received the PhD degree in 1999 from Shinshu University. He is now a research associate of Okayama University. His main fields of research are finite field theory and its applications. He is a member of IEICE and IEEE.

Tomoki Yoshida graduated from the Department of Communication Network Engineering, the Faculty of Engineering, Okayama University, in 2006. He is now with the Graduate School of Natural Science and Technology, Okayama University, where he is studying finite field theory.

Yoshitaka Morikawa graduated from the Department of Electronic Engineering, Osaka University in 1969, and obtained the MS degree in 1971. He then joined Matsushita Electric, where he engaged in research on data transmission. In 1972, he became a research associate at Okayama University, and subsequently became an associate professor in 1985. He is now a professor of the Department of Communication Network Engineering. He has been engaged in research on image information processing. He holds a DEng degree.