Basis Translation Matrix between Two Isomorphic Extension Fields via Optimal Normal Basis

Yasuyuki Nogami, Ryo Namba, and Yoshitaka Morikawa

This paper proposes a method for generating a basis translation matrix between isomorphic extension fields. To generate a basis translation matrix, we need the equality correspondence of a basis between the isomorphic extension fields. Consider an extension field F_{p^m} where p is characteristic. As a brute force method, when p^m is small, we can check the equality correspondence by using the minimal polynomial of a basis element; however, when p^m is large, it becomes too difficult. The proposed methods are based on the fact that Type I and Type II optimal normal bases (ONBs) can be easily identified in each isomorphic extension field. The proposed methods efficiently use Type I and Type II ONBs and can generate a pair of basis translation matrices within 15 ms on Pentium 4 (3.6 GHz) when $m\log_2 p = 160$.

Keywords: Public key cryptography, extension field, optimal normal basis, basic translation.

I. Introduction

The algebraic public-key cryptography based on elliptic curves (EC) [1], efficient and compact subgroup trace representations (XTR) [2], and so on are constructed over large finite fields. To use this cryptography in practice, fundamental arithmetic operations such as multiplication and division in the definition field must be efficiently implemented. From the viewpoint of software implementation, it is said that extension fields are superior to prime fields [3].

Recently, several efficient extension fields with a special modular polynomial have been proposed, namely, the optimal extension field (OEF) [4], and the all-one polynomial field (AOPF) [5], [6], which adopt an irreducible binomial and an all-one polynomial as the modular polynomials, respectively. A customized programming library for such an extension field is usually provided so as to be fully efficient. It may consist of multiplication, inversion, Frobenius mapping, exponentiation, square root calculation, and so on. However, such a library does not carry out every operation as quickly as possible. If we can exploit the advantages of some libraries by switching extension fields (libraries), then the application will be faster. In this case, it is important to translate vector representations between two extension fields that are isomorphic to each other. This paper shows such an example.

Moreover, due to compatibility issues, the vector representation of the input element does not facilitate operations. If a more efficient representation exists and the translation is known, then it is possible to carry out the operations in the more efficient representation.

Consider two isomorphic extension fields whose characteristic and extension degree are p and m, respectively. As a brute force method, we can obtain a basis translation

Manuscript received June 18, 2007; revised Dec. 13, 2007.

Yasuyuki Nogami (phone: + 81 86 251 8128, email: nogami@cne.okayama-u.ac.jp), Ryo Namba (email: nanba@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.

matrix by checking the correspondence of the minimal polynomial of a certain *proper element* when p^m is small, where a proper element belongs to the concerned extension field but not to its proper subfield. On the other hand, when p^m is large, it is difficult to obtain such a correspondence between the isomorphic extension fields.

In [7], Paar outlined a brute force method to construct a basis translation matrix between isomorphic binary extension fields F_{2^m} and $\hat{F}_{(2^k)^l}$, where $m = kl$ and $\hat{F}_{(2^k)^l}$ is a binary tower field. This method is based on a root finding algorithm. Let $\hat{A}$ and $f(x)$ be a primitive element of $\hat{F}^*_{(2^k)^l}$ and the modular polynomial of F_{2^m}, respectively. The root finding algorithm tries to determine an index u such that $f(\hat{A}^u) = 0$. This method needs to consider a primitive element of $F^*_{2^m}$. When m is large, it is not easy to compute the index u if we do not have any explicit mathematical relations between $\hat{A}$ and a zero of $f(x)$. Its time complexity is exponential.

Sunar [8] improved Paar's idea by using equal degree factorization [9] instead of the root finding algorithm. This method does not need to prepare any primitive elements. In addition, the target extension field does not need to be a tower field. As described in [10], Sunar's method calculates a translation matrix in polynomial time as

$$O\left(m^{3.06}\left(\log m\right)^2 \cdot \log p\right). \qquad (1)$$

This method efficiently uses the fact that, over a binary extension field, a Frobenius mapping with respect to F_2 is equivalent to a squaring. Therefore, it is not efficient to directly apply this method to an odd characteristic extension field.

In this paper, we propose a method to obtain a basis translation matrix for F_{p^m} via Type I and Type II optimal normal bases (ONBs). This paper only deals with the case that Type I or Type II ONB exists. First, it is shown that the basis translation matrix can be easily obtained when Type I ONB exists in F_{p^m}. In this case, the following two conditions must be satisfied: m+1 is a prime number, and the characteristic p is a primitive element in F_{m+1}. Type I ONB consists of the conjugate zeros of the irreducible polynomial $(x^{m+1}-1)/(x-1)$ over F_p and is obviously identified in each isomorphic extension field. This method efficiently uses the fact that the multiplicative order of each zero of $(x^{m+1}-1)/(x-1)$ is $m+1$. In section III, the method for generating a basis translation matrix via Type I ONB is shown in detail. If the above conditions are not satisfied, we cannot use Type I ONB.

The worst case in which Type I ONB does not exist in F_{p^m} happens when $k=(p^m-1)/(p-1)$ is a large prime number. In this case, the smallest order of an arbitrary proper element in F_{p^m} is k. Considering zeros of $(x^k-1)/(x-1)$, there are k–1 proper elements of the order k. In this case, the multiplicative order cannot play an important role in obtaining the correspondence between the bases of isomorphic extension fields. In this paper, it is shown that some such difficult cases can be overcome when Type II ONB exists in F_{p^m}. In this case, the following conditions must be satisfied: 2m+1 is a prime number, and either p is a primitive element in F_{2m+1} or the order of p in F_{2m+1} is m and $2 \mid (m-1)$. In section IV, a method for generating a basis translation matrix via Type II ONB is proposed. This method uses the fact that the multiplicative order of each zero of $(x^{2m+1}-1)/(x-1)$ is 2m+1. In section V, some examples are shown. Then, the proposed methods are implemented on Pentium 4 (3.60 GHz), which demonstrates that we can obtain a translation matrix within 15 ms, even for a large extension field, as $\log_2 p \cong 32$ and m=5. The previous methods [7], [8] are based on finding a root of the modular polynomial; however, the proposed method is not.

Throughout this paper, p and m are a prime number and a positive integer, respectively; F_p and F_{p^m} denote a prime field and its m-th extension field, respectively; and $F^*_{p^m}$ is the multiplicative group in F_{p^m}. Since this paper considers two bases, $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ in F_{p^m}, the representations of the same element A with $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ are respectively distinguished with subscripts $A_{\boldsymbol{b}}$ and $A_{\hat{\boldsymbol{b}}}$. Without any additional explanation, the lower and upper case letters a and A show elements in the prime field and the extension field, respectively. Moreover, α denotes zeros of the modular polynomial, $\boldsymbol{v}^T$ means the transpose vector of $\boldsymbol{v}$, and $d \mid m$ means that m is divisible by d.

II. Preparation

Let us briefly go over bases in the extension field, the minimal polynomial, basis translation, Type I ONB, Type II ONB, all-one polynomial field (AOPF), and optimal extension field (OEF).

1. Bases in the Extension Field

In order to construct the arithmetic operations in F_{p^m}, we need an irreducible polynomial $f(x)$ of degree m over F_p. Let α be a zero of $f(x)$, that is a proper element[1] in F_{p^m}, then the following set forms a basis:

$$\boldsymbol{b} = \left\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\right\}, \qquad (2)$$

which is called a polynomial basis. An arbitrary element A in F_{p^m} is represented as

[1] In this paper, we call an element that belongs to F_{p^m} but not to its proper subfield a proper element in F_{p^m}.

$$A = a_1 + a_2\alpha + \cdots + a_m\alpha^{m-1}. \tag{3}$$

The vector representation of $A_{\boldsymbol{b}}$ is $[A_{\boldsymbol{b}}] = (a_1, \cdots, a_m)^T$. A multiplication and inversion in F_{p^m} are carried out using the relation $f(\alpha) = 0$; therefore, $f(x)$ is called the modular polynomial of F_{p^m}.

According to the polynomial basis (2), the following set also forms a basis:

$$\left\{\alpha, \alpha^2, \alpha^3, \cdots, \alpha^m\right\}, \tag{4}$$

which is called pseudo-polynomial basis in this paper.

When the following conjugates of α are linearly independent:

$$\left\{\alpha, a^p, \alpha^{p^2}, \cdots, \alpha^{p^{m-1}}\right\}, \tag{5}$$

it is called a normal basis, which is efficient for Frobenius mapping $A \to A^p$. Every basis introduced here consists of m linearly independent elements in F_{p^m}.

2. Minimal Polynomial

The minimal polynomial $M_\alpha(x)$ of a proper element $\alpha \in F_{p^m}$ with respect to F_p is the monic irreducible polynomial of degree m over F_p such that $M_\alpha(\alpha) = 0$. The minimal polynomial $M_\alpha(x)$ divides $x^{p^m} - x$.

3. Basis Translation

Let $N(m)$ be the number of monic irreducible polynomials of degree m over F_p. $N(m)$ is given by [11] as

$$N(m) = \frac{1}{m}\sum_{i|m} \mu(i) p^{m/i}, \tag{6}$$

where $\mu(n)$ is the Möbius function given as

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1, \\ (-1)^k & \text{if } n \text{ is the product of } k \text{ distinct primes,} \\ 0 & \text{if } n \text{ is divisible by } k \text{ distinct primes.} \end{cases} \tag{7}$$

The summation in (6) is carried out for each i that divides m. We can choose one among $N(m)$ irreducible polynomials as the modular polynomial of F_{p^m}. Extension fields F_{p^m} with the same characteristic p and extension degree m are isomorphic to each other [11]. Therefore, when p^m is large, there are many isomorphic extension fields F_{p^m} because $N(m)$ becomes large, as given by (6).

In order to envisage the basis translation, let us consider a quite simple example of the basis translation between $\boldsymbol{b} = \{1, \alpha\}$ and $\hat{\boldsymbol{b}} = \{1, \hat{\alpha}\}$, where α and $\hat{\alpha} \in F_{3^2}$ are zeros of irreducible polynomials $M_a(x) = x^2 + x + 2$ and $M_{\hat{a}}(x) = x^2 + 2x + 2$, respectively. To obtain a basis translation matrix, we need a certain representation of α with $\hat{\boldsymbol{b}}$, that is $\alpha_{\hat{\boldsymbol{b}}}$. Since the order of α is $3^2-1=8$, in other words, α is a primitive element in F_{3^2}, we can consider $\hat{\alpha}$, $\hat{\alpha}+2$, $2\hat{\alpha}$, and $2\hat{\alpha}+1$ in F_{p^m} as the candidates of $\alpha_{\hat{\boldsymbol{b}}}$. Their orders are also 8. Of course, $\alpha \neq \hat{\alpha}$ because their minimal polynomials $M_a(x)$ and $M_{\hat{a}}(x)$ are different from each other. We find that α corresponds to $\hat{\alpha}+2$ or $2\hat{\alpha}$ because

$$M_\alpha(x) = M_{\hat{\alpha}+2}(x) = M_{2\hat{\alpha}}(x) = x^2 + x + 2. \tag{8}$$

In other words,

$$M_\alpha(\hat{\alpha}+2) = M_\alpha(2\hat{\alpha}) = 0. \tag{9}$$

It is noted that $\hat{\alpha}+2$ and $2\hat{\alpha}$ are conjugates with respect to F_3. Thus, the basis translation matrices that translate the vector representation of an element $A_{\boldsymbol{b}} \in F_{3^2}$ to that of $A_{\hat{\boldsymbol{b}}}$ are given as

$$\mathbf{T} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \tag{10}$$

that satisfy

$$\boldsymbol{b} = \hat{\boldsymbol{b}}\mathbf{T}. \tag{11}$$

As mentioned above, when p^m is small, using this brute force method, we can generate a basis translation matrix between $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$. If we can find $B_{\hat{\boldsymbol{b}}}$ such that $M_a(B_{\hat{\boldsymbol{b}}}) = 0$, then $B_{\hat{\boldsymbol{b}}}$ corresponds to α. Then, we will obtain a basis translation matrix. On the other hand, when p^m is large and $N(m)$ is accordingly large, it is difficult to find $B_{\hat{\boldsymbol{b}}}$ such that $M_a(B_{\hat{\boldsymbol{b}}}) = 0$ among many elements in F_{p^m}.

Let us consider the worst case, in which it is too difficult to apply the brute force method in practice. For F_{p^m}, when $k = (p^m - 1)/(p-1)$ is a large prime number, the order of an arbitrary proper element in F_{p^m} is divisible by k. In other words, k is the smallest among the orders of proper elements. There are k–1 proper elements of order k in F_{p^m}. Therefore, we cannot efficiently filter the candidate elements by using the multiplicative order k as previously discussed.

4. Type I Optimal Normal Basis

Let β be a zero of the irreducible all-one polynomial (AOP)[2] $(x^{m+1} - 1)/(x-1)$ over F_{p^m}. Using β in F_{p^m}, the following set forms a normal basis:

[2] Since all the coefficients are 1, it is called an all-one polynomial. The irreducibility of an all-one polynomial can be easily checked by the conditions shown in theorem 1 [12].

$$\left\{\beta, \beta^{p}, \beta^{p^2}, \cdots, \beta^{p^{m-1}}\right\}. \tag{12a}$$

The normal basis (12a) is equivalent to the following pseudo-polynomial basis:

$$\left\{\beta, \beta^2, \beta^3, \cdots, \beta^m\right\}. \tag{12b}$$

Since a normal basis and pseudo-polynomial basis are efficient for an inversion with the Itoh-Tsujii inversion algorithm[3] [13] and a multiplication, respectively, this basis is efficient for implementing fast arithmetic operations in F_{p^m}. Therefore, (12a) is called a Type I ONB. Of course this is equivalent to (12b) [5].

Theorem 1. Type I ONB exists in F_{p^m} if and only if the following conditions are satisfied [14]:

<1.1> m+1 is a prime number,

<1.2> p is a primitive element in F_{m+1}.

According to the above theorem, Type I ONB exists in F_{p^m} only when m is even and larger than 1. It should be noted that the m zeros of the AOP of degree m have the same order m+1. Conversely, there are m elements of order m+1 in F_{p^m} since m+1 is a prime number. Therefore, we have the following lemma.

Lemma 1. When the conditions in theorem 1 are satisfied, m elements of order m+1, zeros of the AOP of degree m, form Type I ONB in F_{p^m}.

According to lemma 1, if the order of an element γ is m+1, then the set $\left\{\gamma, \gamma^2, \cdots, \gamma^m\right\}$ forms Type I ONB. In other words, we can distinguish Type I ONB in F_{p^m} by detecting the generator γ among all elements in F_{p^m}.

5. Type II Optimal Normal Basis

The following theorem shows the conditions for Type II ONB.

Theorem 2. Type II ONB exists in F_{p^m} if and only if the following <2.1> and either <2.2a> or <2.2b> are satisfied [14]:

<2.1> 2m+1 is a prime number,

<2. 2a> p is a primitive element in F_{2m+1},

<2. 2b> $2 \mid (m-1)$ and the order of p in F_{2m+1} is m.

Let β be a zero of $(x^{2m+1}-1)/(x-1)$. Since β satisfies $\beta^{2m+1}=1$, the set (13a) is equivalent to the set (13b) in sequence.

$$\left\{\beta, \beta^2, \cdots, \beta^m, \beta^{m+1}, \cdots, \beta^{2m-1}, \beta^{2m}\right\}, \tag{13a}$$

$$\left\{\beta, \beta^2, \cdots, \beta^m, \beta^{-m}, \cdots, \beta^{-2}, \beta^{-1}\right\}. \tag{13b}$$

If m and p satisfy the conditions shown in theorem 2, let τ_i be $\beta^i + \beta^{-i}$ and the set (14) forms a basis in F_{p^m}:

$$\left\{\tau_1, \tau_2, \tau_3, \cdots \tau_m\right\}. \tag{14a}$$

The set (14a) is equal to

$$\left\{\tau_1, \tau_1{}^{p}, \tau_1{}^{p^2}, \cdots \tau_1{}^{p^{m-1}}\right\}. \tag{14b}$$

The basis (14b) is called Type II ONB. Type II ONB does not require the extension degree m to be even. If m and p satisfy the conditions <2.1> and <2.2a> in theorem 2, the set (13a) forms a pseudo-polynomial basis in $F_{p^{2m}}$, that is, Type I ONB. Then, τ_i is the trace of β^i with respect to F_{p^m}, that is, a subfield of $F_{p^{2m}}$. On the other hand, if m and p satisfy <2.1> and <2.2b>, Type I ONB does not exist in $F_{p^{2m}}$. Both β^i and τ_i are proper elements in F_{p^m}. Therefore, we have the following lemma.

Lemma 2. When the conditions in theorem 2 are satisfied, 2m elements of order 2m+1 are zeros of the AOP of degree 2m, and they are (13a). Type II ONB in F_{p^m} is derived from the 2m elements (13a).

Therefore, we can distinguish Type II ONB in F_{p^m} by detecting an element β of order 2m+1 among elements in $F_{p^{2m}}$ or F_{p^m}.

6. All-One Polynomial Field

We have proposed Type I and Type II AOPFs in previous works [5], [6]. Type I and Type II AOPFs adopt Type I and Type II ONBs, respectively. In both AOPFs, a cyclic vector multiplication algorithm (CVMA) is used to quickly carry out multiplication. Of course, in order to prepare Type I and Type II AOPFs, theorems 1 and 2 must be satisfied, respectively.

7. Optimal Extension Field

Bailey and others proposed an extension field called an optimal extension field (OEF) [4]. The OEF adopts an irreducible binomial $f(x)=x^m-c$ as the modular polynomial and is constructed based on theorem 3.

Theorem 3. If the following two conditions are satisfied, irreducible binomials x^m-c exist over F_q [11]:

<3.1> Each prime factor of m divides q–1,

<3.2> $q \equiv 1 \mod 4$ when $m \equiv 0 \mod 4$.

[3] For this inversion algorithm, Frobenius mapping should be efficiently carried out.

Table 1. Comparison of OEF with Type I and Type II AOPFs.

	Modular polynomial	Basis	Condition
Type I AOPF F_{p^m}	All one polynomial $(x^{m+1}$-1)/(x-1)	Type I ONB $\{\beta,\beta^p,\beta^{p^2},\cdots,\beta^{p^{m-1}}\}$ $=\{\beta,\beta^2,\beta^3,\cdots,\beta^m\}$	Thm.1
Type II AOPF F_{p^m}	All one polynomial $(x^{2m+1}$-1)/(x-1)	Type I ONB $\{\tau_1,\tau_1^p,\tau_1^{p^2},\cdots,\tau_1^{p^{m-1}}\}$ $=\{\tau_1,\tau_2,\tau_3,\cdots,\tau_m\}$ $\tau_i=\beta^i+\beta^{-i}$	Thm.2
OEF F_{q^m}	Irreducible binomial x^m-c, $c\in F_q$	Polynomial basis $\{1,\alpha,\alpha^2,\cdots,\alpha^{m-1}\}$	Thm.3

$^*\alpha$ and β are zeros of modular polynomials, respectively. q is a power of the characteristic p.

To explain our proposed method easily, we consider two isomorphic OEFs in section 5 because we can easily construct OEFs and examine the obtained basis translation matrix. In this paper, we mainly deal with an odd prime number as the characteristic p; therefore, p^m–1 is divisible by 2. Accordingly, we can construct OEF $F_{p^{2m}}$ with a certain modular polynomial x^2– λ over F_{p^m}, where λ is a quadratic non residue in F_{p^m}. In what follows, the method that constructs $F_{p^{2m}}$ over F_{p^m} by using an irreducible binomial as the modular polynomial is referred to as OEF extension.

III. Basis Translation via Type I ONB

Let characteristic p and extension degree m satisfy the conditions of theorem 1. In this case, Type I ONB exists in F_{p^m}. Let us consider a basis translation between two different bases $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ via Type I ONB. According to theorem 1 and lemma 1, there are m proper elements whose order is m +1 in F_{p^m}.

Let $B_{\boldsymbol{b}}$ be a non-zero element in F_{p^m}. Noting that m + 1 is a prime number, if $B_{\boldsymbol{b}}$ satisfies

$$B_{\boldsymbol{b}}^{(p^m-1)/(m+1)}\neq 1, \tag{15}$$

the order of $B_{\boldsymbol{b}}^{(p^m-1)/(m+1)}$ is m +1 as follows:

$$\left(B_{\boldsymbol{b}}^{(p^m-1)/(m+1)}\right)^{m+1}=B_{\boldsymbol{b}}^{p^m-1}=1. \tag{16}$$

Thus, lemma 1 ensures that $B_{\boldsymbol{b}}^{(p^m-1)/(m+1)}$ is a generator of Type I ONB, that is

$$\beta_{\boldsymbol{b}}=B_{\boldsymbol{b}}^{(p^m-1)/(m+1)}. \tag{17}$$

```
    Input:   A basis b and modular polynomial $f(x)$ of degree $m$
   Output:   A matrix M = $[b_{ij}]$ that represent the Type I ONB with
             b.
       1.    Let $B_b \in F_{p^m}$ be a random no-zero element.
       2.    $B_b \leftarrow B_b^{(p^m-1)/(m+1)}$, if $B_b$=1 then, go to 1.
       3.    $C_b \leftarrow B_b$.
       4.    for $i \leftarrow 1$ to $m$ do
       5.         $j \leftarrow 1$.
       6.         for $j \leftarrow 1$ to $m$ do
       7.              $b_{ji} \leftarrow b_j$.
       8.         end for
       9.         $B_b \leftarrow B_b C_b$.
      10.    end for
```

Fig. 1. Algorithm for generating a matrix **M** that represent the Type I ONB with ***b***.

The probability that a non-zero element satisfies (15) is almost $m/(m+1)$. Thus, we can easily determine such an element as $B_{\boldsymbol{b}}$ in F_{p^m}. Then, using (17), we have the representation of Type I ONB with $\boldsymbol{b}$,

$$\left\{\beta_{\boldsymbol{b}},\beta_{\boldsymbol{b}}^2,\cdots,\beta_{\boldsymbol{b}}^{m-1},\beta_{\boldsymbol{b}}^m\right\}=\boldsymbol{b}\mathbf{M}, \tag{18}$$

where

$$\mathbf{M}=\begin{pmatrix}[\beta_{\boldsymbol{b}}]\\ [\beta_{\boldsymbol{b}}^2]\\ \vdots\\ [\beta_{\boldsymbol{b}}^m]\end{pmatrix}^T. \tag{19}$$

As described in section II.3, an element in F_{p^m} is characterized by its order and its minimal polynomial. On the other hand, as shown in lemma 1, we can identify Type I ONB as the set of m elements of order m+1 in F_{p^m}. Figure 1 shows the algorithm for generating the matrix **M**. In Fig. 1, the coefficients of vector and matrix are denoted by lower-case letters with subscripts such as $[B_{\boldsymbol{b}}]=[b_i]$, $\mathbf{M}=[b_{ij}]$.

In the same way, we can obtain the matrix $\hat{\mathbf{M}}$ for $\hat{\boldsymbol{b}}$ such that

$$\left\{\beta_{\hat{\boldsymbol{b}}},\beta_{\hat{\boldsymbol{b}}}^2,\cdots,\beta_{\hat{\boldsymbol{b}}}^{m-1},\beta_{\hat{\boldsymbol{b}}}^m\right\}=\hat{\boldsymbol{b}}\hat{\mathbf{M}}. \tag{20}$$

Therefore, we have

$$\boldsymbol{b}\mathbf{M}=\left\{\beta,\beta^2,\cdots,\beta^m\right\}=\hat{\boldsymbol{b}}\hat{\mathbf{M}}. \tag{21}$$

Thus, the matrices $\mathbf{T}=\mathbf{M}\hat{\mathbf{M}}^{-1}$ and $\mathbf{T}^{-1}=\hat{\mathbf{M}}\mathbf{M}^{-1}$ are the basis translation matrices between $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$.

$$\boldsymbol{b}\underset{\mathbf{T}^{-1}=\hat{\mathbf{M}}\mathbf{M}^{-1}.}{\overset{\mathbf{T}=\mathbf{M}\hat{\mathbf{M}}^{-1}}{\rightleftarrows}}\hat{\boldsymbol{b}} \tag{22}$$

Using **T** and $\mathbf{T}^{-1}$, a basis translation requires m^2 F_p-multiplications.

As shown above, if the conditions of theorem 1 are satisfied, Type I ONB exists in F_{pm}. Accordingly, we can obviously identify the basis elements from lemma 1. Otherwise, we cannot use Type I ONB. Moreover, in the worst case described in section II.3, it is too difficult to identify a certain proper element only from the multiplicative order of element. Of course, as shown in section II.3, it is too difficult to check the minimal polynomial correspondence between isomorphic extension fields. In the next section, it is shown that some such difficult cases are overcome by using Type II ONB.

IV. Basis Translation via Type II ONB

To generate basis translation matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$ between $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ via Type II ONB in F_{p^m}, consider a zero β of $(x^{2m+1}-1)/(x-1)$. In this section, suppose the conditions in theorem 2 are satisfied and let e be

$$e=\begin{cases}\dfrac{p^{2m}-1}{2m+1} & \text{if } <2.2\text{a}> \text{ is satisfied,}\\[2ex] \dfrac{p^{m}-1}{2m+1} & \text{if } <2.2\text{b}> \text{ is satisfied.}\end{cases} \tag{23}$$

As shown in theorem 2, when the condition <2.2b> is satisfied, β is a proper element in F_{p^m}, and $2m+1$ divides p^m–1. Let $B_{\boldsymbol{b}}\in F_{p^m}$ satisfy

$$B_{\boldsymbol{b}}^{e}\neq 1\ . \tag{24}$$

Then, we consider the following elements as in the part of section III after lemma 2:

$$\left\{\,B_{\boldsymbol{b}}^{e}\,,\cdots,B_{\boldsymbol{b}}^{me},\,B_{\boldsymbol{b}}^{(m+1)e},\cdots,B_{\boldsymbol{b}}^{(2m-1)e},B_{\boldsymbol{b}}^{2me}\right\}. \tag{25a}$$

These elements are equivalent to the following elements in sequence:

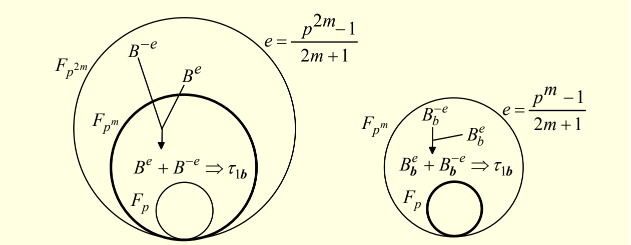
$$\left\{B_{\boldsymbol{b}}^{e},B_{\boldsymbol{b}}^{2e},\cdots,B_{\boldsymbol{b}}^{me},B_{\boldsymbol{b}}^{-me},\cdots,B_{\boldsymbol{b}}^{-2e},B_{\boldsymbol{b}}^{-e}\right\}. \tag{25b}$$

For τ_1 defined in section II.5, lemma 2 ensures that

$$\tau_{1\boldsymbol{b}}=B_{\boldsymbol{b}}^{e}+B_{\boldsymbol{b}}^{-e}\,. \tag{26}$$

Figure 2(b) show the correspondence between τ_{1b} and $B_{\boldsymbol{b}}^{e}+B_{\boldsymbol{b}}^{-e}$. Since $B_{\boldsymbol{b}}$ is a proper element in F_{p^m}, we obtain the following Type II ONB generator τ_{1b} in F_{p^m} using (26). Therefore, a matrix $\mathbf{M}$ is given as

$$\left\{\tau_{1\boldsymbol{b}},\tau_{2\boldsymbol{b}},\cdots,\tau_{m\boldsymbol{b}}\right\}=\boldsymbol{b}\mathbf{M}, \tag{27}$$


$F_{p^{2m}}$
B^{-e}
B^{e}
$e=\dfrac{p^{2m}-1}{2m+1}$
F_{p^m}
$B^{e}+B^{-e}\Rightarrow\tau_{1\boldsymbol{b}}$
F_p
F_{p^m}
B_b^{-e}
B_b^{e}
$e=\dfrac{p^{m}-1}{2m+1}$
$B_{\boldsymbol{b}}^{e}+B_{\boldsymbol{b}}^{-e}\Rightarrow\tau_{1\boldsymbol{b}}$
F_p

(a) Case of <2.2a>

(b) Case of <2.2b>

Fig. 2. Basis translation with Type II ONB.

where

$$\mathbf{M}=\begin{pmatrix}\left[\tau_{1\boldsymbol{b}}\right]\\ \left[\tau_{2\boldsymbol{b}}\right]\\ \vdots\\ \left[\tau_{m\boldsymbol{b}}\right]\end{pmatrix}^{T}. \tag{28}$$

On the other hand, in the case that <2.2a> is satisfied, $\beta=B^e$ is a proper element in $F_{p^{2m}}$ as shown in Fig. 2. Therefore, we need to construct extension field $F_{p^{2m}}$ in order to prepare the set (25a). For the preparation, the second extension field $F_{p^{2m}}$ is constructed over F_{p^m} by OEF extension. We adopt the OEF procedure for the second extension for two reasons. First, it is easy to get the modular binomial of degree 2. Secondly, it is always possible to construct the second extension field $F_{p^{2m}}$ when p^m–1 is divisible by 2.

Let the modular binomial be x^2–λ, where λ is a quadratic non-residue in F_{p^m} such that $\lambda^{(p^m-1)/2}\neq 1$. Then, the square root $\sqrt{\lambda}$ belongs to $F_{p^{2m}}$ as a proper element. In the OEF procedure, a multiplication over $F_{p^{2m}}$ is carried out by three multiplications and four additions over F_{p^m} with the Karatsuba method [4]. Moreover, since $\tau_{1\boldsymbol{b}}$ is the trace of B^e with respect to F_{p^m}, $\tau_{1\boldsymbol{b}}$ is obtained as follows:

$$B^{e}\ =b_{1\boldsymbol{b}}+b_{2\boldsymbol{b}}\sqrt{\lambda},\qquad b_{1\boldsymbol{b}},\,b_{2\boldsymbol{b}}\in F_{p^m}\,, \tag{29a}$$

$$B^{-e}=b_{1\boldsymbol{b}}-b_{2\boldsymbol{b}}\sqrt{\lambda}. \tag{29b}$$

Therefore, we can write

$$B^{e}+B^{-e}=2b_{1\boldsymbol{b}}=\tau_{1\boldsymbol{b}}. \tag{30}$$

Thus, we obtain the set (25a) in $F_{p^{2m}}$. Then, $\tau_{1\boldsymbol{b}}$ is determined from (30). Then, the basis translation matrix is given as in (27) and (28).

V. Examples

1. Example of Basis Translation

Let us consider an easy example of a basis translation between two bases $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ in F_{13^4}. Consider the following Type I ONB in F_{13^4} as

$$\left\{\beta,\beta^2,\beta^3,\beta^4\right\}=\left\{\beta,\beta^{13^3},\beta^{13},\beta^{13^2}\right\}, \tag{31}$$

where β is a zero of the modular polynomial of AOPF of degree 4. Thus its order is equal to $m+1=5$. Let us consider a polynomial basis $\boldsymbol{b}=\{1,\alpha,\alpha^2,\alpha^3\}$ given by α. Let its minimal polynomial be $M_\alpha(x)=x^4-2$. Since α satisfies $\alpha^4=2$, the order of α is 48 as $\alpha^{48}=2^{12}=1$. In this case, we have

$$e=(p^m-1)/(m+1)=(13^4-1)/5=5712. \tag{32}$$

Of course, α does not satisfy (15) because 48 divides 5,712 and $1+\alpha$ satisfies (15). Therefore, we have the following four linearly independent relations:

$$\begin{pmatrix}\beta_b\\ \beta_b^2\\ \beta_b^3\\ \beta_b^4\end{pmatrix}^T=\begin{pmatrix}(1+\alpha)^{5712}\\ \left\{(1+\alpha)^{5712}\right\}^2\\ \left\{(1+\alpha)^{5712}\right\}^3\\ \left\{(1+\alpha)^{5712}\right\}^4\end{pmatrix}^T=\begin{pmatrix}1\\ \alpha\\ \alpha^2\\ \alpha^3\end{pmatrix}^T\mathbf{M}, \tag{33}$$

where $\mathbf{M}$ and $\mathbf{M}^{-1}$ are given by

$$\mathbf{M}=\begin{pmatrix}3&3&3&3\\11&3&10&2\\9&4&4&9\\8&12&1&5\end{pmatrix}, \tag{34a}$$

$$\mathbf{M}^{-1}=\begin{pmatrix}12&8&4&11\\12&12&9&3\\12&1&9&10\\12&5&4&2\end{pmatrix}. \tag{34b}$$

In the same way, let us consider another polynomial basis $\hat{\boldsymbol{b}}=\{1,\hat{\alpha},\hat{\alpha}^2,\hat{\alpha}^3\}$ given by $\hat{\alpha}$. Let its minimal polynomial be $M_{\hat{\alpha}}(x)=x^4-6$. We can calculate the matrix $\hat{\mathbf{M}}$ and its inverse matrix $\hat{\mathbf{M}}^{-1}$ as

$$\begin{pmatrix}\beta_{\hat{b}}\\ \beta_{\hat{b}}^2\\ \beta_{\hat{b}}^3\\ \beta_{\hat{b}}^4\end{pmatrix}^T=\begin{pmatrix}(1+\hat{\alpha})^{5712}\\ \left\{(1+\hat{\alpha})^{5712}\right\}^2\\ \left\{(1+\hat{\alpha})^{5712}\right\}^3\\ \left\{(1+\hat{\alpha})^{5712}\right\}^4\end{pmatrix}^T=\begin{pmatrix}1\\ \hat{\alpha}\\ \hat{\alpha}^2\\ \hat{\alpha}^3\end{pmatrix}^T\hat{\mathbf{M}}, \tag{35}$$

$$\hat{\mathbf{M}}=\begin{pmatrix}3&3&3&3\\5&12&1&8\\1&12&12&1\\5&1&12&8\end{pmatrix}, \tag{36a}$$

$$\hat{\mathbf{M}}^{-1}=\begin{pmatrix}12&2&10&2\\12&3&3&10\\12&10&3&3\\12&11&10&11\end{pmatrix}. \tag{36b}$$

Therefore, we have

$$\mathbf{T}^{-1}=\hat{\mathbf{M}}\mathbf{M}^{-1}=\begin{pmatrix}1&0&0&0\\0&4&0&0\\0&0&3&0\\0&0&0&12\end{pmatrix}, \tag{37a}$$

$$\mathbf{T}=\mathbf{M}\hat{\mathbf{M}}^{-1}=\begin{pmatrix}1&0&0&0\\0&10&0&0\\0&0&9&0\\0&0&0&12\end{pmatrix}. \tag{37b}$$

By the way, $(2+\hat{\alpha})^{5712}$ also satisfies the condition (15). When we adopt $2+\hat{\alpha}$ instead of $1+\hat{\alpha}$, we have another pair of basis translation matrices.

For the above modular polynomials $M_a(x)$ and $M_{\hat{\alpha}}(x)$, we find the following relations:

$$M_\alpha(4x)=(4x)^4-2=9(x^4-6)=9M_{\hat{\alpha}}(x), \tag{38a}$$

$$M_\alpha(7x)=(7x)^4-2=9(x^4-6)=9M_{\hat{\alpha}}(x). \tag{38b}$$

Therefore, we can easily find the correspondence:

$$\alpha=4\hat{\alpha}\ \text{ or }\ \alpha=7\hat{\alpha}. \tag{39}$$

This relation can be also deduced from the matrices (37a) and (37b) because (37) have non-zero elements only in the main diagonal. If $f(x)$ and $\hat{f}(x)$ are not binomials, it is not easy to find such a relation. For example, when the polynomial bases $\boldsymbol{b}=\{1,\alpha,\alpha^2,\alpha^3\}$ and $\hat{\boldsymbol{b}}=\{1,\hat{\alpha},\hat{\alpha}^2,\hat{\alpha}^3\}$ are given by $M_a(x)=x^4+9x^3+11x^2+10x+12$ and $M_{\hat{a}}(x)=x^4+x^3+7x^2+7x+6$, respectively, the basis translation matrices between $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ become

$$\mathbf{T}^{-1}=\hat{\mathbf{M}}\mathbf{M}^{-1}=\begin{pmatrix}1&2&2&3\\0&0&2&3\\0&2&5&2\\0&4&12&3\end{pmatrix}, \tag{40a}$$

$$\mathbf{T}=\mathbf{M}\hat{\mathbf{M}}^{-1}=\begin{pmatrix}1&5&6&3\\0&10&10&5\\0&5&9&2\\0&10&7&3\end{pmatrix}. \tag{40b}$$

2. Example of Efficiency

In this section, using the following parameters p and m, we consider F_{p^m} with Type II OEF and Type II AOPF and demonstrate the efficiency of the basis translation.

$$p = 772248443358014011050875517891 / \\ 830459722926704731 \ \ (160\text{-bit}). \qquad (41a)$$

$$m = 9. \qquad (41b)$$

We consider Type II OEF and Type II AOPF because, as shown in Table 2, Type II OEF carries out multiplication and squaring faster and Type II AOPF carries out Frobenius mapping faster. The authors simulated an exponentiation with the well–known binary method and Avanzi's recent work, Frobenius–abusing exponentiation [15].

The binary method and Frobenius–abusing exponentiation respectively need the following calculation costs,

$$\{(m \log p)/2\}M + (m \log p - 1)S, \qquad (42a)$$

$$\{1 + (m \log p)/2\}\, M + (\log p - 1)S + \{(m-1)\log p\}\, \varphi, \quad (42b)$$

where M, S, and φ denote the calculation costs of multiplication, squaring, and Frobenius mapping in F_{p^m}, respectively. The simulation results tabulated in Table 2 almost follow the above cost evaluation. According to the simulation results, since the back and forth translations need 35.0 µs only, the efficiencies of both Type II OEF and Type II AOPF can be fully used by the basis translation. Specifically, we mainly use Type II OEF, and for an exponentiation with Frobenius-abusing exponentiation, we can apply Type II AOPF by the basis translation.

Table 2. Calculation times of arithmetic operations with Type II OEF and APOF.

(µs)

	Type II OEF	Type II AOPF
Multiplication	26.7	29.0
Squaring	20.4	22.4
Frobenius mapping	5.02	0.80
Binary method	5.06×10^4	5.30×10^4
Frobenius-abusing exponentiation [15]	3.07×10^4	2.63×10^4
Proposed method	3.69×10^5	
Sunar's method	6.8×10^6	
Basis translation ×2	35.0	

Simulated on Pentium 4, 3.6 GHz, NTL [16]

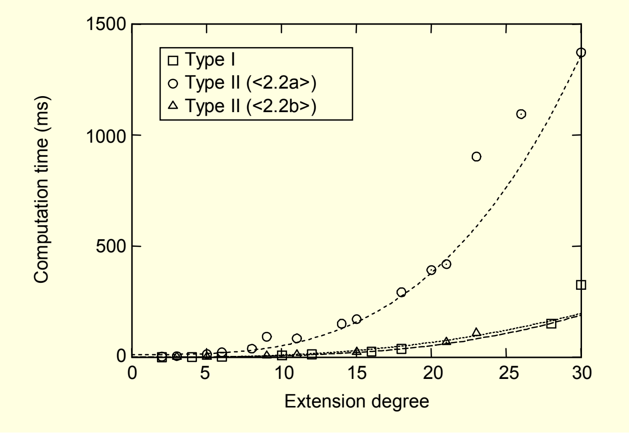


Fig. 3. Computation time to generate a pair of translation matrices $\mathbf{M}$ and $\mathbf{M}^{-1}$ for $\log p \cong 32$.

VI. Simulation

The proposed methods with Type I ONB and Type II ONB (<2.2b>, <2.2a>) were implemented on Pentium 4 (3.60 GHz) with NTL [16] over C++ programming language. Figure 3 shows the experimental results. The graph shows the average computation time per 100 times generating a pair of two matrices such as $\mathbf{M}$ and $\mathbf{M}^{-1}$, where $\mathbf{M}$ is introduced in section III. The characteristic p and modular polynomial $M_a(x)$ were randomly selected such that $\log_2 p \cong 32$.

The exponentiation in (15) and (24) are major computations in the proposed methods. In general, a multiplication in F_{p^m} requires $m^{1.58}$ multiplications over F_p with Karatsuba method and the exponentiation by the well-known binary method requires about $m \log_2 p$ multiplications over F_{p^m}. Thus, the proposed methods need $O(m^{2.58} \log_2 p)$ F_p-multiplications.

The exponentiation (24) in the case of Type II <2.2a> is about six times slower than that of Type I and Type II <2.2b> since we need to prepare $F_{p^{2m}}$ over F_{p^m}. This is because the number of multiplications for the exponentiation in (24) becomes about double because $\log_2 p^{2m} = 2m \log_2 p$. In addition, a multiplication over the extension field $F_{p^{2m}}$ is carried out by three multiplications over F_{p^m} with the OEF technique. Consequently, the calculation cost becomes about six times larger.

The basis translation with Type II ONB also overcomes some of the worst cases in which we cannot use Type I ONB and $(p^m-1)/(p-1)$ becomes a large prime number. For example, when $p=3856381327$ and $m=5$, $k=(p^m-1)/(p-1)$ becomes a large prime number such that $\log_2 k \cong 160$, using Type II ONB, we can generate a pair of basis translation matrices within 15 ms. Thus, the proposed method generates translation matrices systematically without checking the minimal polynomial.

VII. Conclusion

In this paper, we proposed a method to obtain a basis translation matrix with Type I and Type II ONBs. The proposed methods were based on the fact that a zero of the all-one polynomial can be identified in each isomorphic extension field with its order. First, a method for generating a basis translation matrix with Type I ONB was shown. Then, for the case that we could not use Type I ONB, a method for generating a basis translation matrix with Type II ONB was shown. From experimental results, it was shown that the proposed methods could generate a pair of basis translation matrices within 15 ms on Pentium 4 (3.60 GHz) when $m\log_2 p \cong 160$. This paper discussed only Type I and II ONBs; however, the proposed method can be extended for an arbitrary pair of the characteristic and extension degree with the Gauss period normal bases.

References

[1] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curve in Cryptography*, Cambridge University Press LNS 265, 1999.

[2] A. Lenstra and E. Verheul, “The XTR Public Key System,” *Advances in Cryptology: Proc.Crypto*., Springer, 2000, pp. 1-19.

[3] K. Aoki, K. Okeya, T. Kobayashi, Y. Sakai, K. Takashima T. Tsurumaru, G. Yamamoto, H. Yoshida, and D. Watanabe, “Optimization of Prime Field Multiplication Using Redundant Representation,” *SCIS*, 2004, pp. 3A3–5.

[4] D. Bailey and C. Paar, “Optimal Extension Fields for Fast Arithmetic on Public-Key Algorithms,” *Proc. Crypto*. vol. 1462, 1998, pp. 472-485.

[5] 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,” *Trans. IEICE*, vol. E86–A, no. 9, 2003, pp. 2376-2387.

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

[7] C. Paar, “Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields,” PhD thesis, Inst. for Experimental Math., Univ. of Essen, 1994.

[8] B. Sunar, “An Efficient Basis Conversion Algorithm for Composite Fields with Given Representations,” *IEEE Trans. on Computer*, vol. 54, no. 8, 2005, pp. 992-997.

[9] D.G. Cantor and H. Zassenhaus, “A New Algorithm for Factoring Polynomials over Finite Fields,” *Math. of Computation*, vol. 36, 1981, pp. 587-592.

[10] H.W. Lenstra Jr., “Finding Isomorphisms between Finite Fields,” *Math. of Computation*, vol. 56, 1991, pp. 329-347.

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

[12] T. Sugimura and Y. Suetsugu, “Consideration on Irreducible Cyclotomic Polynomials,” *Trans. IEICE*, vol. J73–A, no. 12, 1990, pp. 1929-1935.

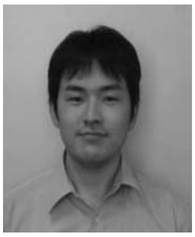
[13] T. Itoh and S. Tsujii, “A Fast Algorithm for Computing Multiplicative Inverses in *GF*(2^m) Using Normal Bases,” *Information and Computation*, vol. 78, no. 3, 1988, pp. 171-177.

[14] A. Menezes, *Applications of Finite Fields*, Kluwer Academic Publishers, Boston, MA, 1993.

[15] R.M. Avanzi and P. Mihaāilescu , “Generic Efficient Arithmetic Algorithms for PAFFs (Processor Adequate Finite Fields),” *LNCS*, vol. 3006, 2004, pp. 321-334.

[16] V. Shoup, A Library for Doing Number Theory, Available from http://www.shoup.net/ntl/

Yasuyuki Nogami graduated from the Department of Electrical and Electronic Engineering, Shinshu University in 1994 and received the PhD degree in 1999 from Shinshu University. He is now a research associate of Okayama University. The main fields of his research are finite field theory and its applications. He is a member of IEEE.



Ryo Namba graduated from the Department of Communication Network Engineering, Okayama University in 2006. He is a master’s course student with the Department of Natural Science and Technology of the Graduate School of Okayama University. His research interests are finite field theory and its applications.

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, an associate professor in 1985. He is now a professor of the Department of Communication Network Engineering, where he has been engaged in research on image information processing. He holds a D.\ Eng.\ degree. He is a member of IEEE, IEICE, and ITE.