TÜBİTAK

Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2017) 41: 96 – 109
© TÜBİTAK
doi:10.3906/mat-1503-84

Free storage basis conversion over finite fields

Ersan AKYILDIZ[1,2], **Ndangang Yampa HAROLD**[1], **Ahmet SINAK**[1,3,*]
[1]Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey
[2]Department of Mathematics, Middle East Technical University, Ankara, Turkey
[3]Department of Mathematics and Computer Sciences, Necmettin Erbakan University, Konya, Turkey

Received: 27.03.2015 • **Accepted/Published Online:** 14.03.2016 • **Final Version:** 16.01.2017

Abstract: Representation of a field element plays a crucial role in the efficiency of field arithmetic. If an efficient representation of a field element in one basis exists, then field arithmetic in the hardware and/or software implementations becomes easy. Otherwise, a basis conversion to an efficient one is searched for easier arithmetic. However, this conversion often brings a storage problem for transition matrices associated with these bases. In this paper, we study this problem for conversion between normal and polynomial bases in the extension field $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ where $q = p^n$. We construct transition matrices that are of a special form. This provides free storage basis conversion algorithms between normal and polynomial bases, which is crucial from the implementation point of view.

Key words: Finite field representation, conversion of field elements, transition matrix, normal basis, polynomial basis

1. Introduction

Efficient finite field arithmetic has a significant role in the implementation of cryptographic schemes [6, 8, 9]. Field elements have various representations depending on the choice of basis. The trivial representation of field elements is the polynomial basis representation. This representation has an efficient arithmetic for field operations: addition, subtraction, and constant multiplication. However, it is not efficient for multiplication or inversion. There have been several attempts to improve multiplication, inversion, and especially squaring. ln the literature, some efficient basis representations such as (optimal) normal basis, Dickson polynomial, Charlier polynomial, and Hermite polynomial representations have been proposed (see for instance [1, 2, 9, 11]). These representations play an important role in efficient arithmetic and are comparable with each other in view of arithmetic complexity. While squaring is not efficient in the polynomial basis representation of binary field elements, the normal basis is attractive for squaring since it can be performed with shift operation only, which is almost free in hardware implementations. The inversion in normal basis representation is also efficiently implemented by using Itoh and Tsujii algorithms in [5]. In order to multiply two elements in normal basis, a specialized version of normal basis with some conditions called optimal normal basis (ONB) of type I and II has been proposed in [10].

One may need a conversion algorithm having low complexity between basis representations. Conversion of binary field elements in various representations has been well studied. In the literature (see for instance [1, 2, 11]), there are conversion algorithms between polynomial basis representation to Hermite–Charlier–Dickson

*Correspondence: ahmet.sinak@metu.edu.tr—
2010 *AMS Mathematics Subject Classification:* 94A60, 11T06.

polynomial bases representations and vice versa with linear complexity. To the best of our knowledge, there is no efficient algorithm (in terms of space and time) to convert a field element from the polynomial basis representation to the normal basis representation (see for instance [6]). The natural method of performing conversion between two bases involves matrix multiplication. For large degree extensions, since the transition matrix is too big, there appears a storage complexity in addition to the time complexity. In this case, known conversion methods may not be used due to the memory problem. Hence, this deficiency leads to the motivation of some storage efficient conversion techniques between two bases in fields. Kaliski and Yin [7] have provided basis conversion techniques in the extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$, where q is a prime power and m is a positive integer. They have described the storage efficient conversion algorithms based on those techniques between polynomial basis and normal basis.

The motivation for the present work comes from [3], in which Gashkov et al. proposed a storage efficient basis conversion algorithm over a field of characteristic 7 in order to compute Tate pairing on hyperelliptic curves of genus 3. For any odd prime p, the storage efficient conversion algorithms between the polynomial and normal bases in the extension field $\mathbb{F}_{p^p}$ over $\mathbb{F}_p$ have been proposed in [13]. The irreducible trinomial $f(x) = x^p - x + 1$ over $\mathbb{F}_p$ was used to construct the extension field $\mathbb{F}_{p^p}$ over $\mathbb{F}_p$. In this paper, we generalize the method given in [13] to the extension field $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$, where p is an odd prime and $q = p^n$ with a positive integer n. We provide the storage efficient basis conversion algorithms in Algorithms 1 and 2 in the extension field $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$. These algorithms efficiently convert the representation of an element in polynomial basis to its representation in normal basis and vice versa without storage complexity.

The time complexity of an algorithm is approximately equal to the number of operations in the algorithm, and the space complexity of an algorithm is equal to the number of memory cells that the algorithm needs. Apart from the importance of the time complexity, its space complexity is also important. An efficient algorithm keeps the time complexity and space complexity as low as possible. Therefore, reducing the time complexity and/or space complexity of an algorithm is of vital importance from the implementation point of view.

This paper is organized as follows: Section 2 introduces basic definitions and gives conditions for the trinomial $f(x) = x^p - x - a \in \mathbb{F}_p[x]$ to be irreducible over $\mathbb{F}_q$. Section 3 constructs the transition matrix M and its inverse matrix M^{-1} without extra computation between normal and polynomial bases. Furthermore, we provide free storage basis conversion algorithms between normal and polynomial bases. Finally, we compute their complexities and compare them with previous results.

2. Preliminary

This section introduces basic definitions and results that will be used in the subsequent sections.

2.1. Finite field representations

For a prime p, the residue class ring $\mathbb{Z}_p$ forms a finite field that is identified with the Galois field $\mathbb{F}_p$ with p elements. To construct a finite extension field over $\mathbb{F}_p$, one needs an irreducible polynomial over $\mathbb{F}_p$. Let $g(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + x^n \in \mathbb{F}_p[x]$ be a monic irreducible polynomial over $\mathbb{F}_p$. Then the residue class ring

$$\mathbb{F}_p[x]/_{\langle f(x)\rangle} = \{c_0 + c_1x + c_2x^2 + \cdots + c_{n-1}x^{n-1} | \, c_i \in \mathbb{F}_p \text{ for } 0 \leq i \leq n-1\} \tag{1}$$

is a finite field with p^n elements, where $\langle g(x)\rangle$ is the principal ideal generated by g in $\mathbb{F}_p[x]$. A finite field in (1) can be denoted by $\mathbb{F}_q$, where $q = p^n$. Up to isomorphism, there is a unique finite field with q elements; however, $\mathbb{F}_q$ has various representations.

Throughout this paper, we consider a finite extension field $\mathbb{F}_{q^p}$ defined over the ground field $\mathbb{F}_q$, where $q = p^n$, $\gcd(p, n) = 1$ and p is an odd prime. Let $\alpha \in \mathbb{F}_{q^p}$ be a root of irreducible polynomial f of degree p over $\mathbb{F}_q$. Then a basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ of the form $\{1, \alpha, \alpha^2, \ldots, \alpha^{p-1}\}$ is called *a polynomial basis* and

$$\mathbb{F}_{q^p} = \{c_0 + c_1\alpha + c_2\alpha^2 + \cdots + c_{p-1}\alpha^{p-1} | c_i \in \mathbb{F}_q \text{ for } 0 \leq i \leq p-1\}$$

is called the *polynomial basis representation* of $\mathbb{F}_{q^p}$. Let $\beta \in \mathbb{F}_{q^p}$ be a root of irreducible normal polynomial f of degree p over $\mathbb{F}_q$. Then a basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ of the form $\{\beta, \beta^q, \beta^{q^2}, \ldots, \beta^{q^{p-1}}\}$ is called a *normal basis* of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ and

$$\mathbb{F}_{q^p} = \{c_0 + c_1\beta^q + c_2\beta^{q^2} + \cdots + c_{p-1}\beta^{q^{p-1}} | c_i \in \mathbb{F}_q \text{ for } 0 \leq i \leq p-1\}$$

gives the *normal basis representation* of $\mathbb{F}_{q^p}$. Note that an irreducible polynomial f of degree p over $\mathbb{F}_q$ is said to be *normal* if all the distinct p roots of f form a normal basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$.

Definition 2.1 *[8] Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ be an irreducible polynomial of degree n over $\mathbb{F}_q$ with $a_n \neq 0$. Then the reciprocal of f, denoted by f^*, is defined as*

$$f^*(x) = x^n f(\frac{1}{x}) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n.$$

Lemma 2.2 *[8] The reciprocal of a monic irreducible polynomial over $\mathbb{F}_q$ is also an irreducible polynomial over $\mathbb{F}_q$.*

An irreducible trinomial has a structure that makes it a good choice for representing the extension field. In some cases, the degree of the middle term is relatively small compared to the polynomial degree. The reduction operation is faster when an irreducible trinomial is used to construct the extension field. Therefore, choosing an irreducible trinomial can lead to a faster arithmetic operation in the field (see for instance [4]). The following theorem gives a necessary condition for a trinomial $f(x) = x^p - x - a \in \mathbb{F}_q[x]$ to be irreducible over $\mathbb{F}_q$.

Theorem 2.3 *[8] Let $a \in \mathbb{F}_q$ and p be the characteristic of $\mathbb{F}_q$. Then the trinomial $f(x) = x^p - x - a$ is irreducible in $\mathbb{F}_q[x]$ if and only if $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(a) \neq 0$, where $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}$ is the absolute trace function from $\mathbb{F}_q$ to $\mathbb{F}_p$.*

In particular, $f(x) = x^p - x + 1$ is an irreducible polynomial over $\mathbb{F}_q$, where $q = p^n$ if and only if $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(1) = n \neq 0$ if and only if $\gcd(p, n) = 1$. Since f is irreducible over $\mathbb{F}_q$ with $\gcd(p, n) = 1$, its reciprocal $f^*(x) = x^p - x^{p-1} + 1$ is also irreducible over $\mathbb{F}_q$ by Lemma 2.2. The following theorem gives the conditions for an irreducible polynomial over $\mathbb{F}_q$ to be normal over $\mathbb{F}_q$.

Theorem 2.4 *[12] Let f be a monic irreducible polynomial of degree n over $\mathbb{F}_q$ and α be a root of f. Let*

$$x^n - 1 = (\mu_1(x)\mu_2(x)\cdots\mu_r(x))^t,$$

where μ_i *is the distinct monic irreducible factors of* x^n-1 *for* $i \in \{1,2,\ldots,r\}$ *and* $t \in \mathbb{Z}^+$. *Suppose that* μ_i *has degree* d_i *for* $i \in \{1,2,\ldots,r\}$. *Then* f *is normal over* $\mathbb{F}_q$ *if and only if*

$$L_{\bar{\mu_i}}(\alpha) \neq 0,$$

where $\bar{\mu_i}(x) = \frac{x^n-1}{\mu_i(x)}$ *and* $L_{\bar{\mu_i}}(x)$ *is the linearized* q*-associate of* $\bar{\mu_i}(x)$ *for* $i \in \{1,2,\ldots,r\}$.

2.2. Our method

Let $\alpha \in \mathbb{F}_{q^p}$ be a root of the irreducible trinomial $f(x) = x^p - x + 1$ over $\mathbb{F}_q$. Then $\mathbb{F}_{q^p}$ has the polynomial basis $\bar{\alpha} = \{\alpha^{p-1}, \ldots, \alpha^2, \alpha, 1\}$ over $\mathbb{F}_q$. Note that the elements of polynomial basis are used in reverse order so that the inverse of transition matrix can be easily computed (see in Section 3.2). By Theorem 2.4, $f(x) = x^p - x + 1$ is not normal over $\mathbb{F}_q$ but its reciprocal $f^*(x) = x^p - x^{p-1} + 1$, which is irreducible over $\mathbb{F}_q$, is normal over $\mathbb{F}_q$. Since $\beta = \alpha^{-1} \in \mathbb{F}_{q^p}$ is a root of f^*, its conjugates β^{q^i} are the distinct roots of f^* for $i \in \{0,1,\ldots,p-1\}$. Then the row vector $\bar{\beta} = \{\beta, \beta^q, \ldots, \beta^{q^{p-1}}\}$ is a normal basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$. In this case, each β^{q^i} is expressed as a linear combination of α^i for $i \in \{0,1,\ldots,p-1\}$, which gives us the transition matrix M from polynomial basis to normal basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$. Then we simply obtain the inverse of the transition matrix from normal basis to polynomial basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$. Therefore, we provide free storage basis conversion algorithms between polynomial basis $\bar{\alpha}$ and normal basis $\bar{\beta}$.

3. Free storage basis conversion in finite fields

Basis conversion involves computing the representation of a field element from one basis to another basis. In the present section, we describe our basis conversion method between polynomial basis and normal basis. Note that all computations of our method are performed in the prime field $\mathbb{F}_p$. Section 3.1 gives the relation between the polynomial basis elements and normal basis elements, which produce the transition matrix M. The special form of M provides a free storage basis conversion algorithm from polynomial basis to normal basis. In Section 3.2, the transition matrix M^{-1} is easily constructed by simple permutation operations from M. Similarly, the special form of M^{-1} provides a free storage basis conversion algorithm from normal basis to polynomial basis. Finally, Sections 3.3 and 3.4 give the complexities of these algorithms and comparison with the previous result, respectively.

3.1. Conversion from polynomial basis to normal basis

Construction of transition matrix from polynomial basis to normal basis: The following lemma serves as a tool to construct the transition matrix from polynomial basis to normal basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$.

Lemma 3.1 *Let* $\alpha \in \mathbb{F}_{q^p}$ *be a root of the irreducible polynomial* $f(x) = x^p - x + 1$ *over* $\mathbb{F}_q$, *where* $q = p^n$. *Then we can see that* $\alpha^{p^i} = \alpha - i$ *for* $i \in \mathbb{N}$.

Proof. We use induction on i to show that $\alpha^{p^i} = \alpha - i$ for $i \in \mathbb{N}$. For $i = 1$, $\alpha^p = \alpha - 1$ since α is a root of f in $\mathbb{F}_{q^p}$. By the freshman's dream, since

$$\alpha^{p^2} = (\alpha^p)^p = (\alpha - 1)^p = \alpha^p - 1 = \alpha - 2,$$

then $\alpha^{p^i} = \alpha - i$ is also true for $i = 2$. Assume that the result $\alpha^{p^i} = \alpha - i$ is true for $i = k$. By the freshman's dream and the above assumption,

$$\alpha^{p^{k+1}} = (\alpha^{p^k})^p = (\alpha - k)^p = \alpha^p - k = \alpha - (k+1).$$

This proves that $\alpha^{p^i} = \alpha - i$ is true for $i = k+1$. Thus, by induction, the result holds for $i \in \mathbb{N}$. □

The next theorem gives the transition matrix M from polynomial basis to normal basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$.

Theorem 3.2 *Let $\mathbb{F}_{q^p}$ be a finite extension field of $\mathbb{F}_q$, where $q = p^n$ and $\gcd(p, n) = 1$ for an odd prime p. Let $\alpha \in \mathbb{F}_{q^p}$ be a root of $f(x) = x^p - x + 1 \in \mathbb{F}_q[x]$ and $\beta = \alpha^{-1}$. Then the matrix*

$$M = \begin{pmatrix} -1 & 0 & 0 & \cdots & 0 & 1 \\ -1 & -n & -(n)^2 & \cdots & -(n)^{p-2} & 0 \\ -1 & -2n & -(2n)^2 & \cdots & -(2n)^{p-2} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -(p-2)n & -((p-2)n)^2 & \cdots & -((p-2)n)^{p-2} & 0 \\ -1 & -(p-1)n & -((p-1)n)^2 & \cdots & -((p-1)n)^{p-2} & 0 \end{pmatrix} \in \mathbb{F}_p^{p\times p} \tag{2}$$

is the transition matrix from the polynomial basis $\bar{\alpha} = \{\alpha^{p-1}, \ldots, \alpha^2, \alpha, 1\}$ to the normal basis $\bar{\beta} = \{\beta, \beta^q, \beta^{q^2}, \ldots, \beta^{q^{p-1}}\}$ of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$, where $\mathbb{F}_p^{p\times p}$ represents the set of $p \times p$ matrices over $\mathbb{F}_p$.

Before giving the proof, we introduce the following lemma to write β^{q^i} as a linear combination of α^i for $i \in \{0, 1, \ldots, p-1\}$.

Lemma 3.3 *Let $\alpha \in \mathbb{F}_{q^p}$ be a root of the irreducible polynomial $f(x) = x^p - x + 1$ over $\mathbb{F}_q$ and $\beta = \alpha^{-1}$, where $q = p^n$. Then β is a root of the irreducible normal polynomial $f^*(x) = x^p - x^{p-1} + 1$ over $\mathbb{F}_q$. Moreover, $\beta^{q^i} = 1 - (\alpha - in)^{p-1}$ for $i \in \{0, 1, \ldots, p-1\}$.*

Proof. We first show that $\beta = \alpha^{-1}$ is a root of f^*. Since $\beta \neq 0$ and α is a root of f,

$$f^*(\beta) = \beta^p f(\frac{1}{\beta}) = \beta^p f(\alpha) = 0.$$

Hence, β is a root of f^*. By Lemma 3.1, the second assertion can be shown as follows:

$$\begin{aligned}
\beta &= \alpha^{-1} = 1 - \alpha^{p-1}, \\
\beta^q &= \left(1 - \alpha^{p-1}\right)^q = 1 - \alpha^{q(p-1)} = 1 - (\alpha - n)^{p-1}, \\
\beta^{q^2} &= \left(1 - (\alpha - n)^{p-1}\right)^q = 1 - (\alpha - n)^{q(p-1)} = 1 - (\alpha^q - n^q)^{p-1} = 1 - (\alpha - 2n)^{p-1}, \\
\beta^{q^3} &= \left(1 - (\alpha - 2n)^{p-1}\right)^q = 1 - (\alpha - 2n)^{q(p-1)} = 1 - (\alpha^q - (2n)^q)^{p-1} = 1 - (\alpha - 3n)^{p-1}, \\
&\vdots \\
\beta^{q^{p-1}} &= \left(1 - (\alpha - (p-2)n)^{p-1}\right)^q = 1 - (\alpha^q - ((p-2)n)^q)^{p-1} = 1 - (\alpha - (p-1)n)^{p-1}.
\end{aligned}$$

Thus, the proof is complete. □

We can also give the following lemma without proof (see for instance [13]).

Lemma 3.4 *Let p be a prime number and j be an integer with $0 \leq j \leq p-1$. Then the binomial coefficient $\binom{p-1}{j} \equiv (-1)^j \mod p$.*

Now we can prove Theorem 3.2 by expressing β^{q^i} as a linear combination of α^i for $i \in \{0, 1, \ldots, p-1\}$.

Proof of Theorem 3.2. By Lemma 3.3, each normal basis element can be written as $\beta^{q^i} = 1 - (\alpha - ni)^{p-1}$ for $i \in \{0, 1, \ldots, p-1\}$. For $i = 0$, we can write $\beta = 1 - \alpha^{p-1}$. For $i \in \{1, 2, \ldots, p-1\}$, using binomial expansion, we have

$$\beta^{q^i} = 1 - (\alpha - ni)^{p-1} = 1 - \sum_{j=0}^{p-1} \binom{p-1}{j} \alpha^{p-1-j} (-ni)^j$$

and by Lemma 3.4,

$$\beta^{q^i} \equiv 1 - \sum_{j=0}^{p-1} (-1)^j \alpha^{p-1-j} (-ni)^j \mod p.$$

Then since $(in)^{p-1} \equiv 1 \mod p$, we get

$$\beta^{q^i} \equiv -\sum_{j=0}^{p-2} \alpha^{p-1-j} (in)^j \mod p$$

for $i \in \{1, \ldots, p-1\}$. Thus we have the following:

$$\begin{array}{lll}
\text{For } i = 1, & \beta^q & = -\alpha^{p-1} - n\alpha^{p-2} - n^2\alpha^{p-3} - \cdots - n^{p-2}\alpha, \\
\text{For } i = 2, & \beta^{q^2} & = -\alpha^{p-1} - (2n)\alpha^{p-2} - (2n)^2\alpha^{p-3} - \cdots - (2n)^{p-2}\alpha, \\
\vdots & & \vdots \\
\text{For } i = p-1, & \beta^{q^{p-1}} & = -\alpha^{p-1} - (p-1)n\alpha^{p-2} - ((p-1)n)^2\alpha^{p-3} - \cdots - ((p-1)n)^{p-2}\alpha.
\end{array}$$

In view of the relation between the powers of α and β, we obtain the transition matrix M in (2) from $\bar{\alpha}$ to $\bar{\beta}$ of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ and this transition system is given as follows:

$$\begin{pmatrix} \beta \\ \beta^q \\ \vdots \\ \beta^{q^{p-2}} \\ \beta^{q^{p-1}} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & \cdots & 0 & 1 \\ -(n)^0 & -(n)^1 & -(n)^2 & \cdots & -(n)^{p-2} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -((p-2)n)^0 & -((p-2)n)^1 & -((p-2)n)^2 & \cdots & -((p-2)n)^{p-2} & 0 \\ -((p-1)n)^0 & -((p-1)n)^1 & -((p-1)n)^2 & \cdots & -((p-1)n)^{p-2} & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha^{p-1} \\ \alpha^{p-2} \\ \vdots \\ \alpha \\ 1 \end{pmatrix} \tag{3}$$

or, equivalently, it is denoted by $\bar{\beta} = M \cdot \bar{\alpha}$, where $M \in \mathbb{F}_p^{p \times p}$. Note that all the computations in M are performed in $\mathbb{F}_p$. □

Free storage basis conversion algorithm from polynomial basis to normal basis: There exists transition matrix M in (2) from polynomial basis $\bar{\alpha}$ to normal basis $\bar{\beta}$. Now we give a free storage basis conversion algorithm in Algorithm 1 to compute the normal basis representation of an element in $\mathbb{F}_{q^p}$ from its polynomial basis representation. Note that the special form of M provides Algorithm 1, which requires no storage complexity.

Let $m \in \mathbb{F}_{q^p}$. Then m is represented uniquely as a linear combination of the polynomial basis elements, $m = a_1\alpha^{p-1} + a_2\alpha^{p-2} + \cdots + a_{p-1}\alpha + a_p$, where $a_i \in \mathbb{F}_q$ for $i \in \{1, 2, \ldots, p\}$. Thus, the row vector

$$m_{\bar{\alpha}} = (a_1, a_2, \ldots, a_p)$$

is called the *polynomial basis representation* of m with $\bar{\alpha}$. Similarly, m can be represented uniquely as a linear combination of the normal basis elements, $m = b_1\beta + b_2\beta^q + \cdots + b_p\beta^{q^{p-1}}$, where $b_j \in \mathbb{F}_q$ for $j \in \{1, 2, \ldots, p\}$. Then the row vector

$$m_{\bar{\beta}} = (b_1, b_2, \ldots, b_p)$$

is called the *normal basis representation* of m with $\bar{\beta}$. Let $\bar{\alpha}[i]$ denotes the i-th component a_i of $m_{\bar{\alpha}}$ and $\bar{\beta}[j]$ denotes the j-th component b_j of $m_{\bar{\beta}}$ for $i, j \in \{1, 2, \ldots, p\}$. Suppose that $m_{\bar{\alpha}} = (\bar{\alpha}[1], \bar{\alpha}[2], \ldots, \bar{\alpha}[p])$ is an input of Algorithm 1. The conversion from polynomial basis representation of m to normal basis representation of m is described in Algorithm 1, which requires no storage complexity. Note that all the computations in Algorithm 1 are performed in $\mathbb{F}_p$.

Algorithm 1 Polynomial basis to normal basis conversion

Input: $m_{\bar{\alpha}} = (\bar{\alpha}[1], \bar{\alpha}[2], \ldots, \bar{\alpha}[p])$
Output: $m_{\bar{\beta}} = (\bar{\beta}[1], \bar{\beta}[2], \ldots, \bar{\beta}[p])$

$z \leftarrow \frac{p-1}{2}$
$\bar{\beta}[1] \leftarrow \bar{\alpha}[p] - \bar{\alpha}[1]$
for $i = 1$ **to** z **do**
    $y_1 \leftarrow 0, y_2 \leftarrow 0, x \leftarrow 1, x_1 \leftarrow 0, x_2 \leftarrow 0, m \leftarrow 1$
    for $j = 1$ **to** z **do**
        $y_1 \leftarrow y_1 - x \cdot \bar{\alpha}[j]$
        $y_2 \leftarrow y_2 - x \cdot \bar{\alpha}[z + j]$
        $x \leftarrow i \cdot n \cdot x$
        $x_1 \leftarrow x_1 - m \cdot \bar{\alpha}[j]$
        $x_2 \leftarrow x_2 - m \cdot \bar{\alpha}[z + j]$
        $m \leftarrow -i \cdot n \cdot m$
    end for
    $\bar{\beta}[i + 1] \leftarrow y_1 + x \cdot y_2$
    $\bar{\beta}[p - i + 1] \leftarrow x_1 + m \cdot x_2$
end for
return $m_{\bar{\beta}}$

The following example shows the conversion from polynomial basis representation of m to normal basis representation of m.

Example 3.5 *Let $q = 49$ with $p = 7$ and $n = 2$. Let $\alpha \in \mathbb{F}_{49^7}$ be a root of the irreducible $f(x) = x^7 - x + 1$ over $\mathbb{F}_{49}$ and $\beta \in \mathbb{F}_{49^7}$ be a root of the normal irreducible polynomial $f^*(x) = x^7 - x^6 + 1$ over $\mathbb{F}_{49}$ where $\beta = \alpha^{-1}$. Then $\bar{\alpha} = \{\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha, 1\}$ is a polynomial basis and $\bar{\beta} = \{\beta, \beta^{49}, \beta^{49^2}, \beta^{49^3}, \beta^{49^4}, \beta^{49^5}, \beta^{49^6}\}$ is a normal*

basis of the extension field $\mathbb{F}_{49^7}$ *over* $\mathbb{F}_{49}$. *As Algorithm 1 runs for an input* $m_{\bar{\alpha}} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7)$, *one can obtain the following results:*

$$\begin{aligned}
\bar{\beta}[1] &= -\bar{\alpha}[1] + \bar{\alpha}[7], \\
\bar{\beta}[2] &= -\bar{\alpha}[1] - 2\bar{\alpha}[2] - 4\bar{\alpha}[3] - \bar{\alpha}[4] - 2\bar{\alpha}[5] - 4\bar{\alpha}[6], \\
\bar{\beta}[3] &= -\bar{\alpha}[1] - 4\bar{\alpha}[2] - 2\bar{\alpha}[3] - \bar{\alpha}[4] - 4\bar{\alpha}[5] - 2\bar{\alpha}[6], \\
\bar{\beta}[4] &= -\bar{\alpha}[1] - 6\bar{\alpha}[2] - \bar{\alpha}[3] - 6\bar{\alpha}[4] - \bar{\alpha}[5] - 6\bar{\alpha}[6], \\
\bar{\beta}[5] &= -\bar{\alpha}[1] + 6\bar{\alpha}[2] - \bar{\alpha}[3] + 6\bar{\alpha}[4] - \bar{\alpha}[5] + 6\bar{\alpha}[6], \\
\bar{\beta}[6] &= -\bar{\alpha}[1] + 4\bar{\alpha}[2] - 2\bar{\alpha}[3] + \bar{\alpha}[4] - 4\bar{\alpha}[5] + 2\bar{\alpha}[6], \\
\bar{\beta}[7] &= -\bar{\alpha}[1] + 2\bar{\alpha}[2] - 4\bar{\alpha}[3] + \bar{\alpha}[4] - 2\bar{\alpha}[5] + 4\bar{\alpha}[6].
\end{aligned}$$

Therefore, one gets the normal basis representation $m_{\bar{\beta}} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ *in terms of the polynomial basis representation of* m. *In fact, this gives us the following transition matrix, which corresponds to* M *in Theorem 3.2 when* $n = 2$ *and* $p = 7$*:*

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -2 & -4 & -1 & -2 & -4 & 0 \\ -1 & -4 & -2 & -1 & -4 & -2 & 0 \\ -1 & -6 & -1 & -6 & -1 & -6 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & 0 \\ -1 & -3 & -2 & -6 & -4 & -5 & 0 \\ -1 & -5 & -4 & -6 & -2 & -3 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}. \tag{4}$$

3.2. Conversion from normal basis to polynomial basis

Construction of transition matrix from normal basis to polynomial basis: To do conversion from the normal basis $\bar{\beta}$ to the polynomial basis $\bar{\alpha}$, one needs the inverse of the transition matrix M. Now we find the inverse of M efficiently by permuting the rows of M. The following lemma is useful to find its inverse (see for instance [13]).

Lemma 3.6 *Let* k *be a positive integer and* p *be a prime number. Then we get*

$$\sum_{m=0}^{p-2} k^m \equiv \begin{cases} -1 & \mod p \text{ if } k \equiv 1 \mod p, \\ 0 & \mod p \text{ otherwise.} \end{cases}$$

Theorem 3.7 *The inverse of the transition matrix* M *in (2) is the following matrix*

$$M^{-1} = \begin{pmatrix} 0 & n^{p-1} & (2n)^{p-1} & \dots & ((p-1)n)^{p-1} \\ 0 & n^{p-2} & (2n)^{p-2} & \dots & ((p-1)n)^{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & n & 2n & \dots & (p-1)n \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \in \mathbb{F}_p^{p \times p}, \tag{5}$$

which is the transition matrix from the normal basis $\bar{\beta} = \{\beta, \beta^q, \beta^{q^2}, \dots, \beta^{q^{p-1}}\}$ *to the polynomial basis* $\bar{\alpha} = \{\alpha^{p-1}, \dots, \alpha^2, \alpha, 1\}$ *of* $\mathbb{F}_{q^p}$ *over* $\mathbb{F}_q$.

Proof. The transition matrix $M \in \mathbb{F}_p^{p\times p}$ contains the following invertible submatrix

$$Q = \begin{pmatrix} -1 & -n & -(n)^2 & \cdots & -(n)^{p-2} \\ -1 & -2n & -(2n)^2 & \cdots & -(2n)^{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -(p-2)n & -((p-2)n)^2 & \cdots & -((p-2)n)^{p-2} \\ -1 & -(p-1)n & -((p-1)n)^2 & \cdots & -((p-1)n)^{p-2} \end{pmatrix} \in \mathbb{F}_p^{(p-1)\times(p-1)},$$

which is the *Vandermonde matrix*. The i-th row R_i of the matrix Q consists of the entries $-(ni)^k$ for $k \in \{0,1,\dots,p-2\}$. Therefore, for $i,j \in \{1,2,\dots,p-1\}$, the rows of Q can be represented as

$$R_i = -((ni)^0,(ni)^1,(ni)^2,\dots,(ni)^{p-2}),$$

$$R_j = -((nj)^0,(nj)^1,(nj)^2,\dots,(nj)^{p-2}).$$

By Lemma 3.6, the multiplication of these two rows can be obtained as follows:

$$R_i \cdot R_j = (n^2ij)^0 + (n^2ij)^1 + \cdots + (n^2ij)^{p-2} \equiv \begin{cases} -1 & \mod p \text{ if } n^2ij \equiv 1 \mod p, \\ 0 & \mod p \text{ otherwise.} \end{cases}$$

The above property allows us to find the inverse matrix Q^{-1} only by performing permutation on the rows of Q such that the i-th column of Q^{-1} is equal to the negative of the transpose of the j-th row of Q, where $ij \equiv n^{-2} \mod p$. Therefore, the i-th column C_i of Q^{-1} can be written as

$$C_i = -R_j^T \tag{6}$$

where $i \equiv j^{-1}n^{-2} \mod p$, R_j represents the j-th row of Q, and R_j^T denotes the transpose of R_j. This can be expressed as follows. Using (6), C_i can be written as

$$\begin{aligned} C_i &= ((ni^{-1}n^{-2})^0,(ni^{-1}n^{-2})^1,(ni^{-1}n^{-2})^2,\dots,(ni^{-1}n^{-2})^{p-2})^T \\ &= ((n^{-1}i^{-1})^0,(n^{-1}i^{-1})^1,(n^{-1}i^{-1})^2,\dots,(n^{-1}i^{-1})^{p-2})^T \end{aligned}$$

where all computations are performed modulo p. Then the following result

$$(n^{-1}i^{-1})^{p-1} = (n^{-1}i^{-1})^{p-2}(n^{-1}i^{-1})^1 \equiv 1 \mod p$$

gives that $in = (n^{-1}i^{-1})^{p-2}$ in modulo p. In the same way, we have

$$\begin{aligned} (in)^2 &= (n^{-1}i^{-1})^{p-3}, \\ (in)^3 &= (n^{-1}i^{-1})^{p-4}, \\ (in)^4 &= (n^{-1}i^{-1})^{p-5}, \\ &\vdots \\ (in)^{p-3} &= (n^{-1}i^{-1})^2, \\ (in)^{p-2} &= (n^{-1}i^{-1})^1, \\ (in)^{p-1} &= (n^{-1}i^{-1})^0. \end{aligned}$$

Then the i-th column of Q^{-1} can be given as $C_i = \left((in)^{p-1}, (in)^{p-2}, \ldots, (in)^2, (in)^1\right)^T$. Therefore, we get

$$Q^{-1} = \begin{pmatrix} n^{p-1} & (2n)^{p-1} & (3n)^{p-1} & \ldots & ((p-1)n)^{p-1} \\ n^{p-2} & (2n)^{p-2} & (3n)^{p-2} & \ldots & ((p-1)n)^{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n^2 & (2n)^2 & (3n)^2 & \ldots & ((p-1)n)^2 \\ n & 2n & 3n & \ldots & (p-1)n \end{pmatrix} \in \mathbb{F}_p^{(p-1)\times(p-1)}. \tag{7}$$

We can obtain the inverse matrix M^{-1} in (5) by the following three steps:

- the entries of the first column of M^{-1} are all 0 except the last one,
- the last row of M^{-1} consists of 1's,
- the rest of the M^{-1} is Q^{-1} in (7).

Thus, the transition matrix M^{-1} from normal basis to polynomial basis of $\mathbb{F}_{q^p}$ over $\mathbb{F}_q$ is obtained and this transition system is given as follows:

$$\begin{pmatrix} \alpha^{p-1} \\ \alpha^{p-2} \\ \vdots \\ \alpha \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & n^{p-1} & (2n)^{p-1} & \ldots & ((p-1)n)^{p-1} \\ 0 & n^{p-2} & (2n)^{p-2} & \ldots & ((p-1)n)^{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & n & 2n & \ldots & (p-1)n \\ 1 & 1 & 1 & \ldots & 1 \end{pmatrix} \cdot \begin{pmatrix} \beta \\ \beta^q \\ \vdots \\ \beta^{q^{p-2}} \\ \beta^{q^{p-1}} \end{pmatrix} \tag{8}$$

or, equivalently, it is denoted by $\bar{\alpha} = M^{-1} \cdot \bar{\beta}$ where $M^{-1} \in \mathbb{F}_p^{p\times p}$. □

The complexity of construction M^{-1} from M is given as follows: To obtain the inverse transition matrix $M^{-1} \in \mathbb{F}_p^{p\times p}$, it is enough to compute $i \equiv j^{-1}n^{-2} \mod p$, where $i, j \in \{1, 2, \ldots, p-1\}$ in the computation point of view. We can use the *extended Euclidean algorithm* to compute $i \equiv j^{-1}n^{-2} \mod p$ with $O(\log^3 p)$ operations under big-O notation. Since there exist $p-1$ columns of Q^{-1}, the complexity of finding i's is $O(p\log^3 p)$. Therefore, the computational complexity of M^{-1} is $O(p\log^3 p)$.
The following example illustrates how to find the inverse of M efficiently.

Example 3.8 *Let $q = 25$ with $p = 5$ and $n = 2$. Let $\alpha \in \mathbb{F}_{25^5}$ be a root of the irreducible polynomial $f(x) = x^5 - x + 1 \in \mathbb{F}_{25}[x]$ and $\beta = \alpha^{-1} \in \mathbb{F}_{25^5}$ be a root of irreducible polynomial $f^*(x) = x^5 - x^4 + 1$ over $\mathbb{F}_{25}$. Then the transition matrix M from the polynomial basis $\{\alpha^4, \alpha^3, \alpha^2, \alpha, 1\}$ to the normal basis $\{\beta, \beta^{25}, \beta^{25^2}, \beta^{25^3}, \beta^{25^4}\}$ of $\mathbb{F}_{25^5}$ over $\mathbb{F}_{25}$ is given by*

$$M = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ -1 & -2 & -4 & -3 & 0 \\ -1 & -4 & -1 & -4 & 0 \\ -1 & -1 & -1 & -1 & 0 \\ -1 & -3 & -4 & -2 & 0 \end{pmatrix} \in \mathbb{F}_5^{5\times 5}.$$

Then the 4×4 *invertible submatrix* Q *is*

$$Q = \begin{pmatrix} -1 & -2 & -4 & -3 \\ -1 & -4 & -1 & -4 \\ -1 & -1 & -1 & -1 \\ -1 & -3 & -4 & -2 \end{pmatrix} \in \mathbb{F}_5^{4\times 4}.$$

Let C_i *be the* i*-th column of* Q^{-1} *and* R_j *be the* j*-th row of* Q*. Then using the relation* $ij \equiv 2^{-2} \equiv 4 \mod 5$ *for* $i, j \in \{1, \ldots, 4\}$*, one can get*

$$C_4 = -R_1^T, \quad C_2 = -R_2^T, \quad C_3 = -R_3^T \quad \text{and} \quad C_1 = -R_4^T.$$

Then we have the inverse matrix

$$Q^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 1 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix} \in \mathbb{F}_5^{4\times 4}, \text{ which gives } M^{-1} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 3 & 4 & 1 & 2 \\ 0 & 4 & 1 & 1 & 4 \\ 0 & 2 & 4 & 1 & 3 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_5^{5\times 5}.$$

Free storage basis conversion algorithm from normal basis to polynomial basis: There exists inverse transition matrix M^{-1} as in (8) from the normal basis $\bar{\beta}$ to the polynomial basis $\bar{\alpha}$. In this section, we give a free storage basis conversion algorithm in Algorithm 2 to compute the polynomial basis representation of an element in $\mathbb{F}_{q^p}$ from its normal basis representation. The special form of M^{-1} provides Algorithm 2, which requires no storage complexity. Suppose that $m_{\bar{\beta}} = (\bar{\beta}[1], \bar{\beta}[2], \ldots, \bar{\beta}[p])$ is an input of Algorithm 2. The conversion from normal basis representation of m to polynomial basis representation of m is described in Algorithm 2. Note that all the computations in Algorithm 2 are performed in $\mathbb{F}_p$.

Algorithm 2 Normal basis to polynomial basis conversion

Input: $m_{\bar{\beta}} = (\bar{\beta}[1], \bar{\beta}[2], \ldots, \bar{\beta}[p])$
Output: $m_{\bar{\alpha}} = (\bar{\alpha}[1], \bar{\alpha}[2], \ldots, \bar{\alpha}[p])$

$z \leftarrow \frac{p-1}{2}$
$\bar{\alpha}[p] \leftarrow \bar{\beta}[p]$
for i =1 **to** z **do**
    $\bar{\alpha}[p] \leftarrow \bar{\alpha}[p] + \bar{\beta}[i] + \bar{\beta}[p-i]$
    $x \leftarrow 1, m \leftarrow 1$
    for j=1 **to** p-1 **do**
        $x \leftarrow i \cdot n \cdot x$
        $m \leftarrow -i \cdot n \cdot m$
        if $i = 1$ **then** $y \leftarrow 0$
        else $y \leftarrow \bar{\alpha}[p-j]$
        end if
        $\bar{\alpha}[p-j] \leftarrow y + m \cdot \bar{\beta}[p-i+1] + x \cdot \bar{\beta}[i+1]$
    end for
end for
return $m_{\bar{\alpha}}$

The following example shows the conversion from normal basis representation of m to polynomial basis representation of m.

Example 3.9 *We consider the irreducible polynomial* $f(x) = x^7 - x + 1$ *over* $\mathbb{F}_{49}$ *in Example 3.5. As Algorithm 2 runs for an input* $m_{\bar{\beta}} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$, *it gives the following results:*

$$\begin{aligned}
\bar{\alpha}[1] &= \bar{\beta}[2] + \bar{\beta}[3] + \bar{\beta}[4] + \bar{\beta}[5] + \bar{\beta}[6] + \bar{\beta}[7],\\
\bar{\alpha}[2] &= 4\bar{\beta}[2] + 2\bar{\beta}[3] + 6\bar{\beta}[4] - 6\bar{\beta}[5] - 2\bar{\beta}[6] - 4\bar{\beta}[7],\\
\bar{\alpha}[3] &= 2\bar{\beta}[2] + 4\bar{\beta}[3] + \bar{\beta}[4] + \bar{\beta}[5] + 4\bar{\beta}[6] + 2\bar{\beta}[7],\\
\bar{\alpha}[4] &= \bar{\beta}[2] + \bar{\beta}[3] + 6\bar{\beta}[4] - 6\bar{\beta}[5] - \bar{\beta}[6] - \bar{\beta}[7],\\
\bar{\alpha}[5] &= 4\bar{\beta}[2] + 2\bar{\beta}[3] + \bar{\beta}[4] + \bar{\alpha}[5] + 2\bar{\beta}[6] + 4\bar{\beta}[7],\\
\bar{\alpha}[6] &= 2\bar{\beta}[2] + 4\bar{\beta}[3] + 6\bar{\beta}[4] - 6\bar{\beta}[5] - 4\bar{\beta}[6] - 2\bar{\beta}[7],\\
\bar{\alpha}[7] &= \bar{\beta}[1] + \bar{\beta}[2] + \bar{\beta}[3] + \bar{\beta}[4] + \bar{\beta}[5] + \bar{\beta}[6] + \bar{\beta}[7].
\end{aligned}$$

Therefore, we get the polynomial basis representation $m_{\bar{\alpha}} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ *in terms of the normal basis representation of* m. *In fact, this gives us the following transition matrix* M^{-1}, *which is the inverse of* M *in (4) in Example 3.5:*

$$\begin{pmatrix} a_1\\ a_2\\ a_3\\ a_4\\ a_5\\ a_6\\ a_7 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 4 & 2 & 6 & 1 & 5 & 3\\ 0 & 2 & 4 & 1 & 1 & 4 & 2\\ 0 & 1 & 1 & 6 & 1 & 6 & 6\\ 0 & 4 & 2 & 1 & 1 & 2 & 4\\ 0 & 2 & 4 & 6 & 1 & 3 & 5\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7 \end{pmatrix}. \tag{9}$$

It can be easily verified whether transition matrix M^{-1} *in (9) from* $\bar{\beta}$ *to* $\bar{\alpha}$ *is the inverse of the transition matrix* M *in (4) from* $\bar{\alpha}$ *to* $\bar{\beta}$.

3.3. Complexities of proposed algorithms

This section gives the time complexities of Algorithms 1 and 2 in terms of the required number of field operations over $\mathbb{F}_p$. Let $\mathbb{F}_{p^n}$ be an extension field of degree n over $\mathbb{F}_p$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_p$. Then $k \in \mathbb{F}_{p^n}$ can be written uniquely $k = a_1\alpha_1 + a_2\alpha_2 + \cdots + a_n\alpha_n$ as a linear combination of the basis elements where $a_i \in \mathbb{F}_p$ for $i \in \{1, 2, \ldots, n\}$. Therefore, there exist n components of the representation of $k \in \mathbb{F}_{p^n}$ over $\mathbb{F}_p$. We assume that the addition and subtraction operations are the same in terms of the time estimate.

The complexities of Algorithms 1 and 2: Let A denotes the required number of additions and M denotes the required number of multiplications in prime field $\mathbb{F}_p$. We know that $\{y_1, y_2, x, x_1, x_2, m\} \subset \mathbb{F}_p$ and $\bar{\alpha}[i], \bar{\beta}[j] \in \mathbb{F}_{p^n}$ for $i, j \in \{1, 2, \ldots, p\}$. Note that for $i, j \in \{1, 2, \ldots, p\}$, the elements $\bar{\alpha}[i] = (\bar{\alpha}[i]_1, \bar{\alpha}[i]_2, \ldots, \bar{\alpha}[i]_n)$ and $\bar{\beta}[j] = (\bar{\beta}[j]_1, \bar{\beta}[j]_2, \ldots, \bar{\beta}[j]_n)$ are coordinates in $\mathbb{F}_{p^n}$ of the vectors $m_{\bar{\alpha}} = (\bar{\alpha}[1], \bar{\alpha}[2], \ldots, \bar{\alpha}[p])$ and $m_{\bar{\beta}} = (\bar{\beta}[1], \bar{\beta}[2], \ldots, \bar{\beta}[p])$, where the elements $\bar{\alpha}[i]_k, \bar{\beta}[j]_k \in \mathbb{F}_p$ for $k \in \{1, 2, \ldots, n\}$.

In Algorithm 1: There exist n field additions over $\mathbb{F}_p$ in Step 2. For each $j \in \{1, \ldots, \frac{p-1}{2}\}$, the required number of field additions and multiplications over $\mathbb{F}_p$ is equal to $4n$ and $4n + 4$, respectively. For each

$i \in \{1, \ldots, \frac{p-1}{2}\}$, in addition to the above operations, there are two multiplications and two additions in $\mathbb{F}_p$. Therefore, the required number of field addition and multiplication operations over $\mathbb{F}_p$ in Algorithm 1 are given by

$$\begin{aligned} A &= \tfrac{p-1}{2}\left(\tfrac{p-1}{2}4n+2\right)+n = np^2 - p(2n-1) + 2n - 1, \\ M &= \tfrac{p-1}{2}\left(\tfrac{p-1}{2}(4n+4)+2\right) = p^2(n+1) - p(2n+1) + n. \end{aligned}$$

Under big-O notation, the required number of field operations over $\mathbb{F}_p$ in Algorithm 1 is $O(np^2)$. Similarly, one can easily compute the required number of field operations over $\mathbb{F}_p$ in Algorithm 2. Then the required number of field addition and multiplication operations over $\mathbb{F}_p$ are given by

$$\begin{aligned} A &= \tfrac{p-1}{2}\left((p-1)2n + 2n\right) = np^2 - np, \\ M &= \tfrac{p-1}{2}\left((p-1)(2n+4)\right) = p^2(n+2) - p(2n+4) + n + 2. \end{aligned}$$

Under big-O notation, the required number of field operations over $\mathbb{F}_p$ in Algorithm 2 is $O(np^2)$.

3.4. Comparison with previous result

There are some conversion algorithms in the literature from polynomial basis to normal basis and vice versa in a general extension field. The storage-efficient basis conversion algorithm in the extension field was proposed in [7]. Moreover, we propose a free storage basis conversion algorithm over a special extension field. To the best of our knowledge in the literature in terms of storage complexity of algorithm, there is no such basis conversion algorithm over an extension field. Although both the algorithm in [7] and the proposed one in this paper have approximately the same time complexity, the latter has no storage requirements. Note that our proposed algorithm computes Tate pairing on elliptic and hyperelliptic curves of genus 3 without any storage while the method in [7] computes it with a huge storage complexity. This makes the proposed algorithm usable in some implementation platforms. The following Table gives the results in [7] for the general extension field and our results for the special extension field.

Table. Complexity of basis conversion over finite field.

Algorithm	Storage complexity	Time complexity	Field, $q = p^n$
[7]	$O(mn \log p)$	$O(mn \log p)$	$\mathbb{F}_{q^m}$
Proposed	$O(1)$	$O(p^2 n)$	$\mathbb{F}_{q^p}$

4. Conclusion

In this paper, we propose storage efficient techniques for conversion from polynomial basis to normal basis and vice versa in the special extension field $\mathbb{F}_{q^p}$. The transition matrix M is of special form and then its inverse M^{-1} can be obtained efficiently by performing the rows of M. The special forms of these transition matrices provide storage efficient conversion algorithms to convert the representation of a field element from polynomial basis to normal basis and vice versa, which require no storage complexity.

Acknowledgment

This paper is a part of the MS thesis of the second author under the supervision of the first author at the Institute of Applied Mathematics at Middle East Technical University, 2014. The second author is supported by Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı. The third author is partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK)-BİDEB 2211 program.

We would like to thank to the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. We are deeply grateful to Sedat Akleylek for his valuable discussions and suggestions that contributed greatly to the presentation and quality of the paper. Lastly, we owe Fuat Erdem a debt of gratitude for his valuable corrections on the typos and the language of the paper.

References

[1] Akleylek S. On the representation of finite fields. PhD, Middle East Technical University, Ankara, Turkey, 2010.

[2] Akleylek S, Cenk M, Özbudak F. Polynomial multiplication over binary fields using Charlier polynomial representation with low space complexity. In: Gong G, Gupta KC, editors. 11th International Conference on Cryptology-INDOCRYPT 2010 in India; 12–15 December 2010; Hyderabad, India. Berlin, Germany: Springer, 2010, pp. 227-237.

[3] Gashkov SB, Bolotov AA, Burtsev AA, Zhebet SY, Frolov AB. On hardware and software implementation of arithmetic in finite fields of characteristic 7 for calculation of Pairings. J Math Sci-Univ Toky 2010; 168: 49-75.

[4] Gathen JVZ. Irreducible trinomials over finite fields. Math Comput 2002; 72: 1987-2000.

[5] Guajardo J, Paar C. Itoh-Tsujii inversion in standard basis and its application in Cryptography and Codes. Design Code Cryptogr 2002; 25: 207-216.

[6] Hankerson D, Menezes A, Vanstone S. Guide to Elliptic Curve Cryptography. New York, NY, USA: Springer Science & Business Media, 2006.

[7] Kaliski BS, Yin YL. Storage efficient finite fields basis conversion. In: Tavares S, Meijer H, editors. Proceedings of the Selected Areas in Cryptography-SAC '98; 17–18 August 1998; Kingston, ON, Canada. Berlin, Germany: Springer-Verlag, 1999, pp. 81-93.

[8] Lidl R, Niederreiter H. Introduction to Finite Fields and Its Applications. Cambridge, UK: Cambridge University Press, 1997.

[9] Menezes A, Blake I, Gao X, Mullen R, Vanstone S, Yaghobian T. Applications of Finite Fields. Boston, MA, USA: Kluwer Academic, 1993.

[10] Muchtadi-Alamsyah I, Yuliawan F. Basis conversion in composite field. International Journal of Mathematics and Computation 2013; 11-17.

[11] Özbudak F, Akleylek S, Cenk M. A new representation of elements in binary fields with subquadratic space complexity multiplication of polynomials. Ieice T Fund Electr 2013; 96-A: 2016-2024.

[12] Schwarz S. Irreducible polynomials over finite fields with linearly independent roots. Math Slovaca 1988; 38: 147-158.

[13] Sial MR, Akyıldız E. Storage free basis conversion over composite finite fields of odd characteristics. Proceedings of 6th International Conference on Information Security and Cryptology-ISCTURKEY; 20–21 September 2013; Ankara, Turkey. 2013, pp. 199-204.