Hermite Polynomial Representation for Finite Fields of Characteristic Three

Sedat Akleylek, Ferruh Özbudak, Canan Özel

***Abstract*—In this paper, we give Hermite polynomial representation for finite fields of characteristic three. We explain how we obtain an arithmetic design for finite field multiplication and modular reduction in $\mathbb{F}_{3^n}$. We show that this representation allows us to perform faster modular reduction in some cases.**

***Index Terms*—representation of finite fields with characteristic three, Hermite polynomials, polynomial multiplication, modular reduction.**

I. Introduction

Arithmetic of finite fields is utilized by the majority of the cryptographic systems [1]. The complexity of the algorithms in these systems is dependent on the arithmetic operations. The efficient implementations of these operations determine the efficiency of the whole system. Many cryptosystems such as RSA, AES, elliptic curve cryptography and also recently pairing based cryptography uses the basic arithmetic operations in finite fields [2]. Binary extension fields and the arithmetic of these fields are extensively studied. Also the importance of general extension field arithmetic have been recently increasing in consequence of the progress in the pairing based cryptography. For instance, Tate Pairing is efficiently computable and achieves its maximum security in finite fields of characteristic three over supersingular elliptic curves [3]. Therefore, the studies on the arithmetic operations on extension fields of $\mathbb{F}_3$ have been recently widespread [4], [5].

By using the polynomial basis representation, the elements of an extension field $\mathbb{F}_{3^n}$ are represented as polynomials of degree $n-1$, where the coefficients of the polynomials are in $\mathbb{F}_3$. Analogous to the binary extension fields, polynomial basis representation needs an irreducible polynomial of degree n with the coefficients from $\mathbb{F}_3$. The main operation of $\mathbb{F}_{3^n}$ is multiplication which contains polynomial multiplication over $\mathbb{F}_3$ and modular reduction with respect to the irreducible polynomial.

In binary extension fields, some special polynomial representations have been proposed to make the field arithmetic efficient. These are Dickson polynomial representation [6], [7], modified redundant representation [8], Charlier polynomial representation [9] and Hermite polynomial representation [10].

S. Akleylek, F. Özbudak and C. Özel are with the Institute of Applied Mathematics, Middle East Technical University, Ankara, 06531, Turkey. e-mail: akleylek, ozbudak, ccimen@metu.edu.tr.

S. Akleylek is also with the Department of Computer Engineering, Ondokuz Mayıs University, Samsun, Turkey.

F. Özbudak is also with the Department of Mathematics, Middle East Technical University, 06531, Ankara, Turkey.

In this paper, we deal with the representation of certain extension fields of $\mathbb{F}_3$. This representation is based on Hermite polynomials. We present the method of multiplication and reduction of Hermite polynomials and we give the complexities of these arithmetic operations. We choose irreducible Hermite binomials to represent the finite field and we obtain some irreducible binomials in Hermite form which allows us an efficient modular reduction.

This paper is organized as follows: in Section 2, we describe Hermite polynomials and give some results on Hermite polynomials in $\mathbb{F}_3[x]$. We present the method of multiplying two polynomials in Hermite representation. In section 3, we give the multiplication and reduction complexities in view of the number of multiplication and addition. We conclude the paper in Section 4.

II. Preliminaries

In this section, we give preliminaries and Hermite representation of finite fields of characteristic three.

Definition 1: [11] The probabilists Hermite polynomials are $H_0(x) = 1$, $H_1(x) = x$ and for $n \geq 2$

$$H_n(x) = x \cdot H_{n-1}(x) - (n-1) \cdot H_{n-2}(x)$$

We give the Hermite polynomials in $\mathbb{F}_3[x]$ for $n \leq 10$ in Table 1.

TABLE I
Hermite polynomials in $\mathbb{F}_3[x]$

$H_0(x)$	1
$H_1(x)$	x
$H_2(x)$	x^2+2
$H_3(x)$	x^3
$H_4(x)$	x^4
$H_5(x)$	x^5+2x^3
$H_6(x)$	x^6
$H_7(x)$	x^7
$H_8(x)$	x^8+2x^6
$H_9(x)$	x^9
$H_{10}(x)$	x^{10}

Remark 1: We note that $deg(H_n(x)) = n$. Since Hermite polynomials have a recursive structure, it is shown in Table 1 that all Hermite polynomials in $\mathbb{F}_3$ have the forms

$$\begin{aligned} H_{3k}(x) &= x^{3k} \\ H_{3k+1}(x) &= x^{3k+1} \\ H_{3k+2}(x) &= x^{3k+2} + 2x^{3k} \end{aligned}$$

for $k \in \mathbb{N}$.

Let $H_n(x) = \beta_n$ be the n-th Hermite polynomial in $\mathbb{F}_3[x]$ with degree n. Our aim is to represent the elements of finite field $\mathbb{F}_{3^n}$ by using Hermite polynomials in $\mathbb{F}_3[x]$. We use the elements represented with standart polynomial representation. Let $a(x) = a'_{n-1}x^{n-1} + a'_{n-2}x^{n-2} + .. + a_0$ where $a'_i \in \mathbb{F}_3$. $a(x)$ can be represented by using Hermite polynomials as $a = a_{n-1}\beta_{n-1} + .. + a_0\beta_0$. Algorithm 1 gives the way of conversion of the coefficients from standart polynomial representation to Hermite polynomial representation.

Algorithm 1 Conversion of Representation of an Element

Input: $a(x) = \sum_{i=0}^{n-1} a'_i x^i$
Output: $(a_0, a_1, ..., a_{n-1})$ where $a = \sum_{i=0}^{n-1} a_i\beta_i$

1: $T \leftarrow a$
2: **for** $i = n$ downto 1 **do**
3: **if** $deg(T) = i$ then
4: **if** $a'_i = 1$ then $a_i \leftarrow 1$, $T \leftarrow T + 2\beta_i$
5: **else if** $a'_i = 2$ then $a_i \leftarrow 2$, $T \leftarrow T + \beta_i$
6: **end if**
7: **else** $a_i \leftarrow 0$
8: **end if**
9: **end for**
10: $a_0 \leftarrow T$

A. Hermite Basis

We convert the representation of elements to Hermite polynomial representation. We need to introduce the multiplication $\cdot$ on Hermite polynomial representation over $\mathbb{F}_{3^n}$ to obtain the basis elements.

Theorem 1: Let $H_n(x) = \beta_n$ be the n-th Hermite polynomial in $\mathbb{F}_3[x]$, where $n \geq 0$. Then for all $i, j \geq 0$ the Hermite polynomials $\{\beta_0, \beta_1, ..., \beta_{n-1}, ..\}$ satisfies the following equation

$$\beta_i \cdot \beta_j = \beta_{i+j} + l \cdot (k \cdot \beta_{i+j-2} + 2 \cdot m \cdot \beta_{i+j-4}) \quad (1)$$

where $l, k, m \in \mathbb{F}_3$ is defined as

$$l = \begin{cases} 0 & \text{if } i \, or \, j \equiv 0 \bmod 3 \\ 1 & \text{otherwise.} \end{cases}$$

$$k = \begin{cases} 1 & \text{if } i \equiv j \bmod 3 \\ 2 & \text{otherwise.} \end{cases}$$

$$m = \begin{cases} 1 & \text{if } i, j \equiv 2 \bmod 3 \\ 0 & \text{otherwise.} \end{cases}$$

Let us give the sketch of the proof. We take each Hermite polynomial $H_n(x)$ in $\mathbb{F}_3[x]$ as β_n. As noted in Remark 1, $\forall k \in \mathbb{N}$, $\beta_{3k} = x^{3k}$, $\beta_{3k+1} = x^{3k+1}$ and $\beta_{3k+2} = x^{3k+2} + 2x^{3k+2}$. If we compute all the cases with respect to the residues of i and j (mod 3) respectively, then we get the equation. As a result, the multiplication of Hermite polynomials, β_i and β_j changes with respect to the indices i,j (mod 3).

Any finite field $\mathbb{F}_{3^n}$ is isomorphic to $\mathbb{F}_3[x]/f(x)$ where $f(x)$ is an irreducible polynomial of degree n. The standart representation of elements of $\mathbb{F}_3[x]/f(x)$ is done by using the polynomial basis $\{1, x, x^2, ..., x^{n-1}\}$. Now, we show that $\{\beta_0, \beta_1, ..., \beta_{n-1}\}$ is a linearly independent set and contitutes a basis of $\mathbb{F}_{3^n}$.

Proposition 1: The set $\{\beta_0, \beta_1, ..., \beta_{n-1}\} \in \mathbb{F}_{3^n}$ is linearly independent.

Proof: In consequence of the recursive structure of Hermite polynomials, in the set $\{\beta_0, \beta_1, ..., \beta_{n-1}\}$, each polynomial β_i has degree i. Therefore, β_i can be expressed as a linear combination of only the polynomials $\{\beta_0, \beta_1, ..., \beta_{i-1}\}$. However, in this set none of the polynomials has degree i. If we generalize this, any of the polynomial can be expressed as a linear combination of $\{\beta_0, \beta_1, ..., \beta_{n-1}\}$. This means that the set is linearly independent. ■

Theorem 2: Let $f = f_n\beta_n + .. + f_0\beta_0$ be an irreducible polynomial of degree n where each $f_i \in \mathbb{F}_3$. The set $\{\beta_0, \beta_1, .., \beta_{n-1}\}$ is a basis of $\mathbb{F}_{3^n} \cong \mathbb{F}_3[x]/f(x)$.

Proof: $\mathbb{F}_{3^n}$ is an extension field of $\mathbb{F}_3$ and also considered as a vector space over $\mathbb{F}_3$. By using Algorithm 1, we can write each element of $\mathbb{F}_{3^n}$ uniquely as a linear combination of $\{\beta_0, \beta_1, .., \beta_{n-1}\}$. We have already shown in Proposition 1 that this set is linearly independent. Hence, the set $\{\beta_0, \beta_1, .., \beta_{n-1}\}$ is a basis of $\mathbb{F}_{3^n} \cong \mathbb{F}_3[x]/f(x)$. ■

III. Multiplication of Polynomials in Hermite Representation

In this section, we describe the multiplication of Hermite polynomials for finite fields of characteristic three and explore the complexity of the multiplication. It is well-known that multiplication of finite field elements can be performed in two steps: first multiplication of polynomials and then modular reduction with respect to the irreducible polynomial that is chosen before [12]. We give the multiplication and the reduction operations, respectively. Throughout this section, $M(n)$ and $A(n)$ denote the minimum number of multiplications and the minimum number of additions for corresponding algorithm for two n-term polynomials multiplication. The following theorem gives the required number of multiplications and additions to multiply polynomials in Hermite basis.

Theorem 3: Let $a = a_{n-1}\beta_{n-1} + ... + a_0\beta_0$ and $b = b_{n-1}\beta_{n-1} + ... + b_0\beta_0$ be n-term polynomials over $\mathbb{F}_3$ and $a \cdot b = c_{2n-2}\beta_{2n-2} + ... + c_0\beta_0$. By using any multiplication method, the coefficients of the polynomial c are computed with

$$\begin{aligned} M(n) \quad &+ \quad M(n - \lceil \frac{n}{3} \rceil - \lfloor \frac{n - \lceil \frac{n}{3} \rceil}{2} \rfloor) \\ &+ \quad 4 \cdot \lfloor \frac{n - \lceil \frac{n}{3} \rceil}{2} \rfloor \cdot (n - \lceil \frac{n}{3} \rceil - \lfloor \frac{n - \lceil \frac{n}{3} \rceil}{2} \rfloor) \\ &+ \quad 3 \cdot M(\lfloor \frac{n - \lceil \frac{n}{3} \rceil}{2} \rfloor) \end{aligned}$$

multiplications and

$$A(n) + A(n-\lceil\frac{n}{3}\rceil-\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor)+A(\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor) + 2\cdot\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor\cdot(n-\lceil\frac{n}{3}\rceil-\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor) + 2\cdot\lceil\frac{2n-4}{3}\rceil+\lfloor\frac{(2n-3)-\lceil\frac{2n-4}{3}\rceil}{2}\rfloor \quad (3)$$

additions.

Proof: By using Theorem 1, the coefficients are computed as follows,

$$\begin{aligned} c_0 &= a_0b_0+a_1b_1+2a_2b_2 \\ c_1 &= a_0b_1+a_1b_0+2a_2b_1+2a_1b_2 \\ c_2 &= a_0b_2+a_2b_0+a_1b_1+a_2b_2 \\ &\vdots \\ c_{2n-3} &= a_{n-2}b_{n-1}+a_{n-1}b_{n-2} \\ c_{2n-2} &= a_{n-1}b_{n-1} \end{aligned}$$

If we compare this multiplication with standard polynomial basis representation, we can see extra terms. All these terms come from the multiplication of the basis elements $\beta_i\cdot\beta_j$ where $0\le i,j\le n-1$ and $i,j\not\equiv 0 \pmod 3$. The multiplication differs with respect to the values of $i \pmod 3$ and $j \pmod 3$, ie. if $i,j\equiv 1 \pmod 3$ then $\beta_i\cdot\beta_j=\beta_{i+j}+\beta_{i+j-2}$ or if $i,j\equiv 2 \pmod 3$ then $\beta_i\cdot\beta_j=\beta_{i+j}+\beta_{i+j-2}+2\cdot\beta_{i+j-4}$. Therefore, the number of the extra terms are related to the number of the indices which are smaller than n and equal to one or two (mod 3). The number of indices that are equal to two (mod 3) is $\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor$ and the number of indices that are equal to one (mod 3) is $n-\lceil\frac{n}{3}\rceil-\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor$. The extra terms are computed with the multiplication of the polynomials contains those numbers of terms, so the total multiplication complexity is determined as the sum of these multiplications.

Similarly, in the total addition complexity, we have additions to combine these extra terms to the ordinary multiplication terms. These are related to the indices of c_i where $0\le i\le 2n-4$ and the remainder of these indices from the division with three. ■

Remark 2: Some of this multiplication complexity comprises scalar multiplication, ie. multiplication by two. If we seperate the number of scalar multiplication from the total multiplication complexity, it is equal to:

$$2\cdot\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor\cdot(n-\lceil\frac{n}{3}\rceil-\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor)+M(\lfloor\frac{n-\lceil\frac{n}{3}\rceil}{2}\rfloor)$$

Recall that since it is just a shift operation, in hardware applications multiplication by two is free [2].

By the choice of the multiplication method, some or all elements of extra terms may be computed in the first part of the algorithm, ie. in n-term polynomial product, so the complexities added to $M(n)$ and $A(n)$ may be smaller than the given ones in the theorem. We explain the theorem with an example, by using the Karatsuba multiplication method [13]. Karatsuba Algorithm decreases the number of coefficient multiplications compared to the schoolbook or ordinary multiplication method by using the divide-conquer idea recursively. Example 1 shows the required multiplications for 4-term polynomials over $\mathbb{F}_p$ where $p\ge 2$.

Example 1: Let $a(x)=a_3x^3+...+a_0$ and $b(x)=b_3x^3+...+b_0$ be 4-term polynomials over $\mathbb{F}_p$ where $p\ge 2$. Karatsuba algorithm computes the product $c(x)=c_6x^6+..+c_0$ with the following multiplications:

$$\begin{aligned} m_0 &= a_0b_0 \\ m_1 &= a_1b_1 \\ m_2 &= a_2b_2 \\ m_3 &= a_3b_3 \\ m_4 &= (a_0+a_1)(b_0+b_1) \\ m_5 &= (a_0+a_2)(b_0+b_2) \\ m_6 &= (a_1+a_3)(b_1+b_3) \\ m_7 &= (a_2+a_3)(b_2+b_3) \\ m_8 &= (a_0+a_1+a_2+a_3)(b_0+b_1+b_2+b_3) \end{aligned}$$

By appropriate additions of m_i's the coefficients of c are obtained. Here, Karatsuba algorithm uses 9 multiplications and 24 additions [13].

Now, in the second example we multiply two 4-term polynomials in Hermite basis representation by using Karatsuba multiplication method.

Example 2: Let $a=a_3\beta_3+a_2\beta_2+a_1\beta_1+a_0\beta_0$ and $b=b_3\beta_3+b_2\beta_2+b_1\beta_1+b_0\beta_0$ be 4-term polynomials over $\mathbb{F}_3$. Let $a\cdot b=c=c_6\beta_6+c_5\beta_5+...+c_0\beta_0$. Then

$$\begin{aligned} c_0 &= a_0b_0+a_1b_1+2a_2b_2 \\ c_1 &= a_0b_1+a_1b_0+2a_2b_1+2a_1b_2 \\ c_2 &= a_0b_2+a_2b_0+a_1b_1+a_2b_2 \\ c_3 &= a_0b_3+a_3b_0+a_1b_2+a_2b_1 \\ c_4 &= a_1b_3+a_3b_1+a_2b_2 \\ c_5 &= a_2b_3+a_3b_2 \\ c_6 &= a_3b_3 \end{aligned}$$

The extra terms in this multiplication are $a_1b_1+2a_2b_2$, $2a_2b_1+2a_1b_2$ and a_2b_2. By applying Karatsuba algorithm for performing an ordinary polynomial multiplication as in Example 1, we compute the terms except these extra terms by using 9 multiplications and 24 additions. Now we compute the extra costs.

We recall that a_1b_1 and a_2b_2 are obtained. To compute $2a_2b_2$, we need 1 multiplication. To compute $2a_2b_1+2a_1b_2$, we do

$$2\cdot[(a_1+a_2)\cdot(b_1+b_2)-a_1b_1-a_2b_2]$$

The cost is 2 multiplications and 4 additions. Also we need 4 additions to add these terms to the general result. So we need totally 3 multiplications and 8 additions as an extra cost. As

a result, by using Karatsuba algorithm we need $9+3=12$ multiplications and $24+8=32$ additions to multiply $c=a\cdot b$.

By this way, we give the upper bound for the complexity of the multiplication of two elements in Hermite basis representation. Now, we explore the reduction part of the multiplication of the elements in $\mathbb{F}_{3^n}$.

A. Irreducible Hermite Binomials

The Hermite polynomials in $\mathbb{F}_3$ are given in Table I for $n \leq 10$. Because of the recursive structure of the Hermite polynomials, we can say that the polynomials including constant terms are only $H_0(x)$ and $H_1(x)$ or we call them β_0 and β_1. We explore low weight irreducible Hermite polynomials for the performance of the reduction operation, so we look for the Hermite binomials. Since irreducible polynomials should include the constant term, there are two forms of the Hermite binomials that are $\beta_n+\beta_0$ and $\beta_n+\beta_2$. We give selected irreducible Hermite binomials in Table II.

TABLE II
IRREDUCIBLE HERMITE BINOMIALS

$\beta_3+\beta_2$	$\beta_4+\beta_2$	$\beta_{11}+\beta_0$
$\beta_{12}+\beta_2$	$\beta_7+\beta_2$	$\beta_{26}+\beta_0$
$\beta_{15}+\beta_2$	$\beta_{19}+\beta_2$	$\beta_{35}+\beta_0$
$\beta_{60}+\beta_2$	$\beta_{28}+\beta_2$	$\beta_{119}+\beta_0$
$\beta_{111}+\beta_2$	$\beta_{67}+\beta_2$	$\beta_{146}+\beta_0$
$\beta_{183}+\beta_2$	$\beta_{151}+\beta_2$	$\beta_{242}+\beta_0$

We list irreducible binomials with respect to the indices $n \pmod 3$. Note that, there are two forms of irreducible binomials and we can group these forms with respect to the indices in mod 3. In the first column, the binomials are in the form $\beta_n+\beta_2$, where $n\equiv 0 \pmod 3$. In the second column, the binomials are in the form $\beta_n+\beta_2$, where $n\equiv 1 \pmod 3$ and in the last column, they are in the form $\beta_n+\beta_0$, $n\equiv 2 \pmod 3$. The reduction operation is different due to the chosen binomials from each column of the Table II, because the multiplication of Hermite polynomials differs with respect to the values of the indices in mod 3, as it is given in Theorem 1.

1) Reduction: Let us call the irreducible Hermite binomial f. By using f, reduction operation with respect to modulo f can be performed as follows:

We deal with each binomial form respectively, so first let's take $f=\beta_n+\beta_2$ where $n\equiv 0 \pmod 3$ and let $n\leq i\leq 2n-2$. Then,

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i\\ 2\beta_2\cdot\beta_{i-n} &= \beta_i\\ \beta_i &= 2\beta_{i-n}.\beta_2\end{aligned}$$

We formulate this by using Theorem 1 as follows:

$$\beta_i = 2\beta_{i-n+2} + r\cdot(s\cdot\beta_{i-n}+t\cdot\beta_{i-n-2}) \qquad (4)$$

$$r=\begin{cases}0 & \text{if } i-n\equiv 0 \bmod 3\\ 1 & \text{otherwise.}\end{cases}$$

$$s=\begin{cases}1 & \text{if } i-n\equiv 1 \bmod 3\\ 2 & \text{if } i-n\equiv 2 \bmod 3\end{cases}$$

$$t=\begin{cases}1 & \text{if } i-n\equiv 2 \bmod 3\\ 0 & \text{otherwise.}\end{cases}$$

Remark 3: If $n\equiv 0 \pmod 3$ then by theorem 1, l is zero and note that $\beta_n=2\cdot\beta_2$.

Now, let $n\equiv 1 \pmod 3$ and let $f=\beta_n+\beta_2$. The reduction of β_i where $n\leq i\leq 2n-2$ differs with respect to the value of $i-n \pmod 3$. Let's give the reduction formulas for the values zero, one and two, respectively.

If $i-n\equiv 0 \pmod 3$:

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i\\ \beta_i &= 2\beta_2\cdot\beta_{i-n}\\ \beta_i &= 2\beta_{i-n+2}\end{aligned}$$

If $i-n\equiv 1 \pmod 3$:

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i+\beta_{i-2}\\ 2\beta_2\cdot\beta_{i-n} &= \beta_i+\beta_{i-2}\\ \beta_i &= 2\beta_2\cdot\beta_{i-n}+2\beta_{i-2}\\ \beta_i &= 2\beta_{i-n+2}+\beta_{i-n}+2\beta_{i-2}\end{aligned}$$

If $i-n\equiv 2 \pmod 3$:

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i+2\beta_{i-2}\\ 2\beta_2\cdot\beta_{i-n} &= \beta_i+2\beta_{i-2}\\ \beta_i &= 2\beta_2\cdot\beta_{i-n}+\beta_{i-2}\\ \beta_i &= 2\beta_{i-n-2}+2\beta_{i-n}+\beta_{i-n-2}+\beta_{i-2}\end{aligned}$$

If we combine these, we get the equation:

$$\beta_i = 2\beta_{i-n+2}+u\cdot(v\cdot\beta_{i-n}+w\cdot\beta_{i-n-2}+2v\cdot\beta_{i-2}) \qquad (5)$$

$$u=\begin{cases}0 & \text{if } i-n\equiv 0 \bmod 3\\ 1 & \text{otherwise.}\end{cases}$$

$$v=\begin{cases}1 & \text{if } i-n\equiv 1 \bmod 3\\ 2 & \text{if } i-n\equiv 2 \bmod 3\end{cases}$$

$$w=\begin{cases}1 & \text{if } i-n\equiv 2 \bmod 3\\ 0 & \text{otherwise.}\end{cases}$$

Now let $f=\beta_n+\beta_0$, where $n\equiv 2 \pmod 3$. Since the value of $n\equiv 2 \pmod 3$, the multiplication changes with respect to the values of $i-n \pmod 3$, where $n\leq i\leq 2n-2$. Let us write this seperately again.

If $i-n\equiv 0 \pmod 3$:

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i\\ 2\beta_0\cdot\beta_{i-n} &= \beta_i\\ \beta_i &= 2\beta_{i-n}\end{aligned}$$

If $i-n\equiv 1 \pmod 3$:

$$\begin{aligned}\beta_n\cdot\beta_{i-n} &= \beta_i+2\beta_{i-2}\\ 2\beta_0\cdot\beta_{i-n} &= \beta_i+2\beta_{i-2}\\ \beta_i &= 2\beta_{i-n}+\beta_{i-2}\end{aligned}$$

If $i-n \equiv 2 \pmod 3$:

$$\begin{aligned} \beta_n \cdot \beta_{i-n} &= \beta_i + \beta_{i-2} + 2\beta_{i-4} \\ 2\beta_0 \cdot \beta_{i-n} &= \beta_i + \beta_{i-2} + 2\beta_{i-4} \\ \beta_i &= 2\beta_{i-n} + 2\beta_{i-2} + \beta_{i-4} \end{aligned}$$

If we combine these in a formula:

$$\beta_i = 2\beta_{i-n} + h \cdot (y \cdot \beta_{i-2} + z \cdot \beta_{i-4}) \quad (6)$$

$$h = \begin{cases} 0 & \text{if } i-n \equiv 0 \bmod 3 \\ 1 & \text{otherwise.} \end{cases}$$

$$y = \begin{cases} 1 & \text{if } i-n \equiv 1 \bmod 3 \\ 2 & \text{if } i-n \equiv 2 \bmod 3 \end{cases}$$

$$z = \begin{cases} 1 & \text{if } i-n \equiv 2 \bmod 3 \\ 0 & \text{otherwise.} \end{cases}$$

We perform the reduction operations by choosing all possible irreducible Hermite binomials as some are given in the Table II. Since we have three kinds of irreducible Hermite binomials, we get three reduction formulas in equations 4, 5 and 6. Each reduction formulas changes due to the difference $i-n$ and so the value of the index i (mod 3), which is the index of reduced element β_i. In the form $\beta_n+\beta_2$ where $n \equiv 0$ (mod 3), the multiplication of $\beta_n \cdot \beta_{i-n}$ directly gives β_i, so we can reduce β_i in one step for this form. However, in the other forms reduction is not completed in one step that we can see from equations 5 and 6. The terms β_{i-2}, β_{i-4} should be reduced until the indices drop into the interval $[0, n-1]$. We do not carry on these reductions, since it depends on the value of the index i.

The following table gives the reduction complexity for each form of the irreducible Hermite binomials.

TABLE III
REDUCTION COMPLEXITY

Form	# Constant Multiplications	# Additions
$\beta_n + \beta_2$ ($n \equiv 0 \bmod 3$)	4	2
$\beta_n + \beta_2$ ($n \equiv 1 \bmod 3$)	6*	3*
$\beta_n + \beta_0$ ($n \equiv 2 \bmod 3$)	4*	2*

Remark 4: Since the reductions in equations 5 and 6 are not completed, we give the signed numbers in Table III which are the numbers of constant multiplications and additions for one step of the reduction. The total number of multiplications and additions for these binomials are computed with multiplying these numbers by the number of reduction steps.

As a result, between these binomials, $\beta_n+\beta_2$, where $n \equiv 0$ (mod 3) has the least number of constant multiplications and the least number of additions.

IV. CONCLUSION

In this paper, we give Hermite polynomial representation for finite fields of characteristic three. We explain how we obtain an arithmetic design for finite field multiplication and modular reduction in $\mathbb{F}_{3^n}$. We show that this representation allows us to perform efficient modular reduction if we select the irreducible Hermite binomial $\beta_n + \beta_2$, where $n \equiv 0$ (mod 3). Hermite trinomials also can be chosen as low weight irreducible polynomials. However, we realize that in this case extra multiplications and additions are obtained redundantly in the reduction operation due to the multiplication on Hermite polynomial representation and this increases the reduction complexity.

ACKNOWLEDGMENT

Sedat Akleylek and Ferruh Özbudak were partially supported by TUBITAK under the Grant No. TBAG-109T672.

REFERENCES

[1] I.F. Blake, G. Seroussi, N.P. Smart, Elliptic curves in cryptography *London Mathematical Society Lecture Note Series 265*, Cambridge Univ. Press, 1999
[2] H. Cohen, G. Frey, Handbook of Elliptic and Hyperelliptic Curve Cryptography, *Discrete Math. Appl., Chapman Hall/CRC*, 2006)
[3] T. Kerins, W.P. Marnane, E. M. Popovici and P.S.L.M. Barreto. Efficient Hardware for the Tate Pairing Calculation in Characteristic Three. *CHES 2005*, pp. 412-426, 2005.
[4] K. Harrison, D. Page and N. Smart. Software implementation of finite fields of characteristic three, for use in pairing-based cryptosystems.*LMS Journal of Computation and Mathematics*, 5 (2002), 181-193.
[5] R. Granger, D. Page and M. Stam. Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three. *IEEE Transactions on Computers*, 54 (2005), 852-860.
[6] M.A. Hasan and C. Negre. Subquadratic Space Complexity Multiplication over Binary Fields with Dickson Polynomial Representation.*WAIFI 2008, LNCS 5130*, pp.88-102, 2008.
[7] M.A. Hasan and C. Negre. Low Space Complexity Multiplication over Binary Fields with Dickson Polynomial Representation. *IEEE Trans. on Computers*, vol.60, no.4, pp.602-607, 2011.
[8] S. Akleylek, F. Özbudak, Modified Redundant Representation for Designing Arithmetic Circuits with Small Complexity. *IEEE Trans. Computers* 61(3): 427-432 (2012)
[9] S. Akleylek, M. Cenk, F. Özbudak Polynomial Multiplication over Binary Fields Using Charlier Polynomial Representation with Low Space Complexity. *INDOCRYPT* 2010: 227-237
[10] S. Akleylek, M. Cenk, F. Özbudak, A New Representation of Elements of Binary Fields with Subquadratic Space Complexity Multiplication of Polynomials, submitted, 2010.
[11] D. Drake, The Combinatorics of Associated Hermite Polynomials, European Journal of Combinatorics, vol.30 no.4, pp.1005-1021, 2009.
[12] D. Hankerson, A. Menezes and S. Vanstone, Guide to Elliptic Curve Cryptography, *Springer*, 2004.
[13] A. Weimerskirch, C. Paar, Generalizations of the Karatsuba Algorithm for Efficient Implementations. http://eprint.iacr.org/2006/224, 2006.