Reducing the Complexity of Normal Basis Multiplication

Ömer Eğecioğlu and Çetin Kaya Koç
Department of Computer Science
University of California Santa Barbara
{omer,koc}@cs.ucsb.edu

Abstract

In this paper we introduce a new transformation method and a multiplication algorithm for multiplying the elements of the field GF(2^k) expressed in a normal basis. The number of XOR gates for the proposed multiplication algorithm is fewer than that of the optimal normal basis multiplication, not taking into account the cost of forward and backward transformations. The algorithm is more suitable for applications in which tens or hundreds of field multiplications are performed before needing to transform the results back.

1 Introduction

Arithmetic operations in finite fields GF(q) have several applications in cryptography, coding, and computer algebra. Particularly of interest are fields of characteristic 2, where $q = 2^k$, which have various uses in elliptic curve cryptography for large values of k, usually in the range from 160 to 521. Furthermore, smaller fields, for example, $k = 8$ (AES/Rijndael) and $k = 2, \ldots, 32$ (Reed-Solomon and BCH codes) are also commonly used. Elliptic curve cryptographic protocols generally require fast hardware and software implementations of the multiplication and inversion operations. On the other hand circuits for these operations for small fields may be implemented completely in hardware and/or using a table lookup approach.

The subject of this paper is multiplication algorithms in the binary extension fields GF(2^k). There are essentially two categories of algorithms, based on the representation of field elements using polynomial basis or normal basis. In this paper, a new transformation method and a new multiplication algorithm for normal basis is introduced. First we will review the existing algorithms for both polynomial and normal bases, and then introduce the transformation method, which maps the elements of the field uniquely to the same set. This also slightly changes the definition of multiplication, as the product is computed in the transformed domain. The resulting algorithm is useful for applications where several normal basis multiplications are performed, as is the case for elliptic curve cryptography.

2 Polynomial Basis Multiplication

In the polynomial basis a field element $a \in$ GF(2^k) is represented as a polynomial of degree less than or equal to $k-1$, written as $a(x) = \sum_{i=0}^{k-1} a_i x^i$ with coefficients $a_i \in \{0, 1\}$. The addition of two field elements a and b is accomplished by adding the coefficients a_i and b_i in GF(2), which is the XOR of two binary values. On the other hand, the multiplication $c = a \cdot b$ is accomplished by first computing the degree $2k-2$ product polynomial $c'(x) = a(x)b(x)$ and then reducing it modulo the irreducible polynomial $p(x)$ of degree k, in order to obtain the product $c(x)$ of degree at most $k-1$:

$$c(x) = a(x)b(x) \bmod p(x) \ .$$

The multiplication of the individual coefficients a_i and b_i require 2-input AND gates, while the steps of the multiplication is accomplished by shift and XOR operations in software, or rewiring and XOR gates in hardware.

There are various polynomial basis multiplication algorithms; the work of Mastrovito is quite remarkable [14, 15]. This was followed up in [19, 25, 8, 28].

The properties of the irreducible polynomial $p(x)$ are also important, and not to be overlooked. In general, low Hamming weight irreducible polynomials [23], for example, trinomials and pentanomials are preferred. These yield many efficient algorithms [11, 9, 27]. A comprehensive list of polynomial basis multiplication algorithms can be found in [5].

A polynomial basis multiplication algorithm of interest is the Montgomery multiplication algorithm, proposed by Koç and Acar in [12]. This algorithm has three important properties that do not exist in the common algorithms found in the literature. The first property is that it works for general irreducible polynomials, not just special ones (such as trinomials, pentanomials, or all-one polynomials), making it more suitable for software implementations of cryptographic algorithms. The second property is that, it actually computes

$$\bar{c}(x) = \text{MonPro}(\bar{a}(x), \bar{b}(x)) = \bar{a}(x)\bar{b}(x)x^{-k} \bmod p(x) \ , \qquad (1)$$

instead of the usual $c(x) = a(x)b(x) \bmod p(x)$. This algorithm is actually the polynomial analogue of the Montgomery multiplication algorithm for integers [16, 13]. In order to compute a field multiplication, the elements a and b are first *forward transformed* into the polynomial Montgomery domain

$$\begin{array}{rcl} a \to \bar{a} & : & \bar{a}(x) = a(x)x^k \bmod p(x) \\ b \to \bar{b} & : & \bar{b}(x) = b(x)x^k \bmod p(x) \end{array}$$

and then, the Montgomery product is computed

$$\begin{array}{rcl} \bar{c}(x) & = & \bar{a}(x)\bar{b}(x)x^{-k} \bmod p(x) \\ & = & a(x)x^k b(x)x^k x^{-k} \bmod p(x) \\ & = & a(x)b(x)x^k \bmod p(x) \ , \end{array}$$

which is equal to $c(x)x^k$. When the result $\bar{c}$ needs to be transformed back to c, we use

$$\bar{c} \to c \quad : \quad c(x) = \bar{c}(x)x^{-k} \bmod p(x) \ .$$

Of course, in order to be useful, one should not be needing too many forward $c \to \bar{c}$ and backward $\bar{c} \to c$ transformations. This is never a problem for applications we are considering, such as elliptic curve cryptography, where tens of field multiplications are performed for each elliptic curve point addition and doubling operations, and hundreds of field multiplications are performed for elliptic curve point multiplication operations.

Finally, the third property of the Montgomery multiplication algorithm is that it is more suitable for software implementations for general irreducible polynomials, because the reduction proceeds word-by-word, due to the properties of the Montgomery multiplication for integers [16, 13]. However, it can be argued [4] that this is a moot point, since in most cases, we have low Hamming weight irreducible polynomials (trinomials and pentanomials) and there is no particular need for general irreducible polynomials.

Before closing this section we should also add that the Montgomery multiplication in GF(2^k) is not the only transformative multiplication algorithm; there are also spectral methods [21, 22], embedding techniques [24], and transformation of the field elements into polynomials [26].

3 Normal Basis Multiplication

An element β of the field GF(2^k) is called a normal element, if all 2^k elements of the field can be uniquely written as a linear sum of the powers of two powers of β as

$$a \;=\; \sum_{i=0}^{k-1} a_i \beta^{2^i} \;=\; a_0\beta + a_1\beta^2 + a_2\beta^4 + \cdots + a_{k-1}\beta^{2^{k-1}} \ ,$$

such that $a_i \in \{0,1\}$. Since the work of Kurt Wilhelm Sebastian Hensel in 1888, we know that there always exists a normal element for any prime p and integer k for the field GF(p^k).

The normal representation of $a = (a_{k-1}a_{k-2}\cdots a_1 a_0) \in \text{GF}(2^k)$ is particularly useful for squaring the element a. Since $\beta^{2^k} = \beta$, we obtain a^2 as

$$\begin{aligned} a^2 &= (a_0\beta + a_1\beta^2 + a_2\beta^4 + \cdots + a_{k-1}\beta^{2^{k-1}})^2 \\ &= a_0\beta^2 + a_1\beta^4 + a_2\beta^8 + \cdots + a_{k-1}\beta^{2^k} \\ &= a_{k-1}\beta + a_0\beta^2 + a_1\beta^4 + a_2\beta^8 + \cdots + a_{k-2}\beta^{2^{k-1}} \\ &= (a_{k-2}a_{k-3}\cdots a_1 a_0 a_{k-1}) \ . \end{aligned}$$

Therefore, the normal expression of a^2 is obtained by left-rotating the digits of the normal expression of a. The ease of squaring in normal basis is remarkable, but the multiplication is more complicated.

In the following we explain the steps of the normal basis multiplication, which will be used to develop a new transformation method and normal basis multiplication algorithm.

In order to describe the computational requirements of the normal basis multiplication, we follow the steps of the Massey-Omura algorithm [18, 20], which gives the general outline for normal basis multiplication. Given the input operands a and b, the Massey-Omura multiplier first generates all partial products $a_i b_j$ for $0 \le i,j \le k-1$ using AND gates, and then sums these partial product terms using multi-operand adders (whose unit element is an XOR gate).

There are k^2 partial product terms $a_i b_j$, a computation that can be performed using k^2 2-input AND gates in a single AND gate delay. Decidedly this computation is optimal; k^2 is both upper and lower bound on the number of partial product terms, because all of them need to be computed.

However, in the computation of each of the product terms c_r for $0 \le r \le k-1$, we need only a subset of the k^2 partial product terms $a_i b_j$. According to the optimality argument [17] of the normal basis multiplication, the number of $a_i b_j$ terms needed to compute any of c_r is at least $2k-1$ for GF(2^k). If there exists a normal basis for which the number of $a_i b_j$ terms for computing any of c_r is exactly $2k-1$ for GF(2^k), then this normal basis is called *optimal*. It should be noted that optimal normal bases do not exist for every value of k in GF(2^k), which is easily verified for small values of k using exhaustive search. All values of $k \le 2000$ for which there is an optimal normal basis of GF(2^k) are listed in [6].

Several constructions of optimal normal bases are given in [17], together with a conjecture that describes all finite binary field extensions which have an optimal normal basis. It was proven by Gao and Lenstra in [7, 6] that the optimal normal basis constructions given in [17] are indeed all there is. These constructions are summarized in the theorem below:

Theorem 1. *An optimal normal basis for GF(2^k) exist only in either of the following cases:*

1. *If $k+1$ is prime and 2 is a primitive element in $\mathcal{Z}_{k+1}$, then each of the k nonunit $(k+1)$th root of unity forms an optimal normal basis in GF(2^k).*
2. *If $2k+1$ is prime and*

   2a: Either, 2 is primitive in $\mathcal{Z}_{2k+1}$;

   2b: Or, $2k+1 = 3 \pmod 4$ and 2 generates quadratic residues in $\mathcal{Z}_{2k+1}$;

   then, $\beta = \gamma + \gamma^{-1}$ generates an optimal normal basis in GF(2^k), where γ is a primitive $(2k+1)$th root of unity.

For historical reasons, the optimal normal bases that satisfy the first part of the above theorem are named Type 1, while the ones that follow from the second part are named Type 2 bases.

3.1 Optimal Normal Multiplication in GF(2^2)

The elements of GF(2^2) expressed in polynomial basis are $\{0, 1, x, x+1\}$. There is only one irreducible polynomial of degree 2 over GF(2), which is $p(x) = x^2 + x + 1$. Since $k + 1 = 3$ is prime, and 2 is a primitive element in $\mathcal{Z}_3$ (because $2^1 = 2$ and $2^2 = 1$), the field GF(2^2) has Type 1 optimal normal basis. The 2 nonunit 3rd roots of the unity in GF(2^2) are the two optimal normal basis elements of GF(2^2), and they are x and $x + 1$, because $x^3 = (x+1)^3 = 1 \bmod p(x)$.

We illustrate the normal basis multiplication in GF(2^2) using the optimal normal element $\beta = x$. Let the normal representations of two operands given as $a = a_0\beta + a_1\beta^2$ and $b = b_0\beta + b_1\beta^2$. The product c is equal to

$$c = a_0b_0\beta^2 + a_0b_1\beta^3 + a_1b_0\beta^3 + a_1b_1\beta^4 \ .$$

This expansion contains the terms β^2, β^3 and β^4. First we need to obtain the normal representation of β^3. Since $\beta = x$, we have $\beta^2 = x^2 = x + 1 \bmod p(x)$, and thus, $\beta^3 = x(x+1) = x^2 + x = 1 \bmod p(x)$. Furthermore, $\beta + \beta^2 = x + x + 1 = 1$, and thus, we have

$$\begin{aligned}
\beta^0 &= \beta + \beta^2 = 1 \ , \\
\beta^1 &= \beta = x \ , \\
\beta^2 &= \beta^2 = x + 1 \ , \\
\beta^3 &= \beta + \beta^2 = 1 \ .
\end{aligned}$$

Substituting β^3 and β^4 with $\beta + \beta^2$ and β in the expansion of the product c, we obtain

$$\begin{aligned}
c &= a_0b_0\beta^2 + a_0b_1(\beta + \beta^2) + a_1b_0(\beta + \beta^2) + a_1b_1\beta \\
&= (a_0b_1 + a_1b_0 + a_1b_1)\beta + (a_0b_0 + a_0b_1 + a_1b_0)\beta^2 \qquad (2)
\end{aligned}$$

which gives the individual terms of the product c as

$$\begin{aligned}
c_0 &= a_0b_1 + a_1b_0 + a_1b_1 \\
c_1 &= a_0b_0 + a_0b_1 + a_1b_0 \qquad (3)
\end{aligned}$$

We now define the $k \times k$ matrix $\boldsymbol{\lambda}$ such that $\boldsymbol{\lambda}_{ij} = \beta^{2^i+2^j}$ for $0 \leq i, j \leq k-1$, which for $k = 2$ is given as

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 \\ \beta^3 & \beta^4 \end{bmatrix} = \begin{bmatrix} \beta^2 & \beta+\beta^2 \\ \beta+\beta^2 & \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \beta + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \beta^2 \;=\; \boldsymbol{\lambda}^{(0)}\beta + \boldsymbol{\lambda}^{(1)}\beta^2 \tag{4}$$

We will explain some properties of $\boldsymbol{\lambda}$ in the following subsection, however, it suffices to say that the matrix $\boldsymbol{\lambda}$ has $k(2k-1) = 6$ entries (3 in each column or 3 in each row). Furthermore, when it is expanded into the powers of β, we obtain the 0-1 matrices $\boldsymbol{\lambda}^{(0)}$ and $\boldsymbol{\lambda}^{(1)}$, each of which has $2k-1 = 3$ nonzero entries.

3.2 Optimal Normal Multiplication in GF(2^3)

We now illustrate the normal basis multiplication in GF(2^3), which has Type 2 optimal normal basis. We use the optimal normal element $\beta = x+1$, and irreducible polynomial $p(x) = x^3+x+1$. In this section we will also describe certain properties of the $\boldsymbol{\lambda}$ matrices that are relevant to our proposed multiplication algorithm. Let a and b given as

$$\begin{aligned} a &= a_0\beta + a_1\beta^2 + a_2\beta^4 \ , \\ b &= b_0\beta + b_1\beta^2 + b_2\beta^4 \ . \end{aligned}$$

The product c would be

$$\begin{aligned} c \;=\;& a_0b_0\beta^2 + a_0b_1\beta^3 + a_0b_2\beta^5 + a_1b_0\beta^3 + a_1b_1\beta^4 + a_1b_2\beta^6 + \\ & a_2b_0\beta^5 + a_2b_1\beta^6 + a_2b_2\beta^8 \ . \end{aligned} \tag{5}$$

This expansion contains terms β^2, β^4, and β^8. Since $\beta^8 = \beta$, these are the powers of 2 powers of β, required for normal representation in GF(2^3). However, the above expansion of c also contains other powers: β^3, β^5, and β^6. All powers of β can be expressed in polynomial basis, reduced modulo the irreducible polynomial $p(x)$, generating a conversion table between the powers of β and the elements of the field represented in polynomial basis. Furthermore, once the polynomial representations of β, β^2 and β^4 are obtained, we can also obtain the normal representations of all elements. Table 1 contains the polynomial, the normal, and the powers of β representations of the field elements.

Table 1: The normal and the powers of $\beta = x+1$ representations of elements in GF(2^3) with irreducible polynomial $p(x) = x^3+x+1$.

$\beta^4+\beta^2+\beta$	β^0
β	β^1
β^2	β^2
$\beta^4+\beta$	β^3
β^4	β^4
$\beta^4+\beta^2$	β^5
$\beta^2+\beta$	β^6
$\beta^4+\beta^2+\beta$	β^7

Substituting the powers of β in the expansion of the product c in Eqn. (5), we obtain

$$\begin{aligned} c &= a_0b_0\beta^2 + a_0b_1(\beta+\beta^4) + a_0b_2(\beta^2+\beta^4) + a_1b_0(\beta+\beta^4) + a_1b_1\beta^4 + a_1b_2(\beta+\beta^2) + \\ & \quad a_2b_0(\beta^2+\beta^4) + a_2b_1(\beta+\beta^2) + a_2b_2\beta \\ &= (a_0b_1 + a_1b_0 + a_2b_2 + a_1b_2 + a_2b_1)\beta + (a_0b_0 + a_0b_2 + a_2b_0 + a_1b_2 + a_2b_1)\beta^2 + \\ & \quad (a_0b_1 + a_1b_0 + a_0b_2 + a_2b_0 + a_1b_1)\beta^4 \ . \end{aligned}$$

which gives the individual terms of the product c as

$$\begin{aligned} c_0 &= a_0b_1 + a_1b_0 + a_2b_2 + a_1b_2 + a_2b_1 \ , \\ c_1 &= a_0b_0 + a_0b_2 + a_2b_0 + a_1b_2 + a_2b_1 \ , \\ c_1 &= a_0b_1 + a_1b_0 + a_0b_2 + a_2b_0 + a_1b_1 \ . \end{aligned} \tag{6}$$

The $k \times k$ matrix $\boldsymbol{\lambda}$ with entries $\boldsymbol{\lambda}_{ij} = \beta^{2^i+2^j}$ for $0 \le i, j \le k-1$ is given as

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 \\ \beta^3 & \beta^4 & \beta^6 \\ \beta^5 & \beta^6 & \beta^8 \end{bmatrix} = \begin{bmatrix} \beta^2 & \beta+\beta^4 & \beta^2+\beta^4 \\ \beta+\beta^4 & \beta^4 & \beta+\beta^2 \\ \beta^2+\beta^4 & \beta+\beta^2 & \beta \end{bmatrix} \ . \tag{7}$$

The $\boldsymbol{\lambda}$ matrix contains all powers of β needed in the computation of c, as given in Eqn. (5). It can also be expressed by separating the powers of β as

$$\boldsymbol{\lambda} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \beta + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \beta^2 + \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \beta^4 \ = \ \boldsymbol{\lambda}^{(0)}\beta + \boldsymbol{\lambda}^{(1)}\beta^2 + \boldsymbol{\lambda}^{(2)}\beta^4 \ , \tag{8}$$

where, the 3×3 matrices $\boldsymbol{\lambda}^{(r)}$ for $r = 0, 1, 2$ have entries in $\{0, 1\}$. Since the product c in Eqn. (5) can be written as

$$\begin{aligned} c &= \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\beta^{2^i+2^j} \ = \ \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\boldsymbol{\lambda}_{ij} \\ &= \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\boldsymbol{\lambda}_{ij}^{(0)} \ \beta + \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\boldsymbol{\lambda}_{ij}^{(1)} \ \beta^2 + \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\boldsymbol{\lambda}_{ij}^{(2)} \ \beta^4 \end{aligned}$$

By expressing c as $c = c_0\beta + c_1\beta^2 + c_2\beta^4$, we can write the individual terms of the product c_r as

$$c_r = \sum_{i=0}^{2}\sum_{j=0}^{2} a_ib_j\boldsymbol{\lambda}_{ij}^{(r)} \ .$$

The complexity of computing the terms c_r depends on the number of 1s in the matrices $\boldsymbol{\lambda}^{(r)}$ for $r = 0, 1, 2$. Furthermore, the matrices $\boldsymbol{\lambda}_{ij}^{(r)}$ have the following properties:

$$\boldsymbol{\lambda}_{ij}^{(r)} = \boldsymbol{\lambda}_{ji}^{(r)} \tag{9}$$

$$\boldsymbol{\lambda}_{i+1,j+1}^{(r+1)} = \boldsymbol{\lambda}_{i,j}^{(r)} \tag{10}$$

The first property is due to the fact that $2^i + 2^j = 2^j + 2^i$, and thus,

$$\boldsymbol{\lambda}_{ij} = \beta^{2^i+2^j} = \beta^{2^j+2^i} = \boldsymbol{\lambda}_{ji} \ .$$

The second property follows from the fact that $2^{i+1} + 2^{j+1} = 2(2^i + 2^j)$, and thus

$$\beta^{2^{i+1}+2^{j+1}} = (\beta^2)^{2^i+2^j} \ ,$$

which implies

$$\boldsymbol{\lambda}_{i+1,j+1}(\beta) = \boldsymbol{\lambda}_{i,j}(\beta^2) \ .$$

Considering also the fact that $\beta^{2^k} = \beta$, we obtain Eqn. (10). Note that all index arithmetic, i.e., increments such as $i+1$ and $j+1$ are considered mod 3 in GF(2^3) or mod k in GF(2^k).

The optimal basis theorem [17, 7] teaches that if the normal element β is optimal, then each one of the matrices $\boldsymbol{\lambda}_{ij}^{(r)}$ for $0 \le r \le k-1$ has $2k-1$ nonzero entries, as was the case for both GF(2^2) in Eqn. (4) and GF(2^3) in Eqn. (8), where we have $2 \cdot 2 - 1 = 3$ and $2 \cdot 3 - 1 = 5$ nonzero elements in each matrix. Equivalently, the matrix $\boldsymbol{\lambda}$ has $k(2k-1)$ individual terms such that each term is a power of 2 power of β, as was shown in Eqn. (4) for GF(2^2) and Eqn. (7) for GF(2^3), which has 15 terms.

3.3 Complexity and Implementation

While there are several different ways of putting things together, the basic outline of a normal basis multiplier has 2 steps:

- Step 1: Compute $a_i b_j$ terms using k^2 2-input AND gates.
- Step 2: Sum the subset of the terms as implied by the nonzero entries of the $\boldsymbol{\lambda}_{ij}^{(r)}$ matrix using $2k-2$ 2-input XOR gates for each c_r term.

Step 1 and Step 2 can be performed sequentially, partially parallel, or fully parallel. Since Step 1 is pretty obvious, that is, it computes k^2 different things, there is no need to dwell on it. Step 2, on the other hand, provides several different implementations and optimizations. For example, we can implement a single circuit consisting of $2k-2$ XOR gates (arranged either as a linear array or a binary tree) to compute c_0, and reuse the same circuit for computing c_r for $r = 1, 2, \ldots, k-1$, by only shifting the input operands a_i and b_j for $0 \le i, j \le k-1$. Figure 1 illustrates the construction.

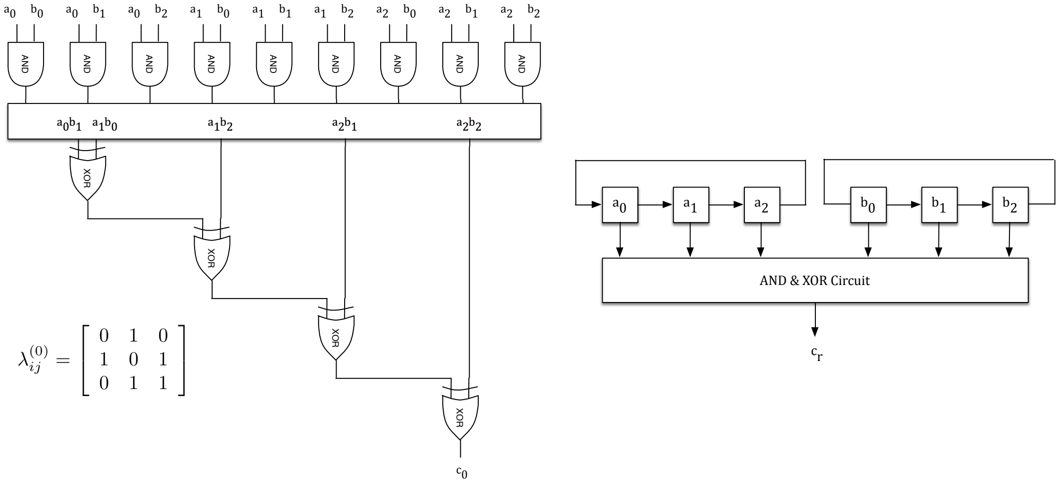

Figure 1: Optimal normal basis multiplier construction for GF(2^3).

We are intentionally ignoring some of the details of the circuit in Figure 1, since there are various ways to arrange the circuit elements, for example, sequential, parallel, systolic, and pipelined circuits have been designed [2, 3, 1]. Our focus in this paper is not on how the individual steps of the optimal normal basis multiplications are performed or how individual circuit elements are arranged. Rather, we are interested in discovering whether there is another way to multiply two elements expressed in a normal basis defined by β in GF(2^k).

4 The Proposed Method

Let a and b be expressed in an optimal normal basis, using the normal element β in GF(2^k). The multiplication of a and b produces c, expressed as

$$c = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} a_i b_j \beta^{2^i + 2^j} \;=\; \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} a_i b_j \boldsymbol{\lambda}_{ij} \ , \tag{11}$$

such that the $k \times k$ matrix $\boldsymbol{\lambda}$ has $k(2k-1)$ terms of type β^{2^i} for $i = 1, 2, \ldots, k-1$. Let α be a fixed element of GF(2^k),such that $\alpha \neq 0, 1$. We will also need α^{-1} which can be precomputed using the extended Euclidean algorithm or the Fermat's method, or the Itoh-Tsujii method [10], which is actually based on the Fermat's method.

We define a new multiplication function, which we denote as NewPro, that takes two elements $\bar{a}$ and $\bar{b}$ of the field GF(2^k), which are the forward transformations of a and b, as

$$a \rightarrow \bar{a} \quad : \quad \bar{a} = a \cdot \alpha^{-1} \ , \tag{12}$$

$$b \rightarrow \bar{b} \quad : \quad \bar{b} = b \cdot \alpha^{-1} \ . \tag{13}$$

The transformation requires the precomputed α^{-1} value. The operands a and b are now expressed in "bar" domain. The NewPro algorithm takes $\bar{a}$ and $\bar{b}$ as input and computes $\bar{c}$ as

$$\bar{c} \ = \ \text{NewPro}(\bar{a}, \bar{b}) \ = \ \bar{a} \cdot \bar{b} \cdot \alpha \ . \tag{14}$$

After the multiplication, the resulting $\bar{c}$ can be backward transformed to "nobar" domain using

$$\bar{c} \rightarrow c \ : \ c \ = \ \bar{c} \cdot \alpha \ . \tag{15}$$

since

$$\bar{c} \cdot \alpha = (\bar{a} \cdot \bar{b} \cdot \alpha) \cdot \alpha = (a \cdot \alpha^{-1}) \cdot (b \cdot \alpha^{-1}) \cdot \alpha^2 = a \cdot b \ .$$

We call the fixed element α as the *NewPro transformation constant*. We apply forward transformation by multiplying with α^{-1} and backward transformation by multiplying with α.

This new transformation method reminds us of the Montgomery transformation, however, no polynomial analogue of the Montgomery multiplication algorithm is implied here. Instead, we will propose a direct method to obtain $\bar{c}$ from $\bar{a}$ and $\bar{b}$, that will require fewer XOR gates than the optimal normal basis multiplication. However, this does not mean that the optimal normal basis multiplication is not optimal. The optimal normal basis multiplication computes $c = a \cdot b$, while our algorithm computes $\bar{c} = \bar{a} \cdot \bar{b} \cdot \alpha$ for a judiciously selected (and fixed) element $\alpha \in$ GF(2^k).

Furthermore, in order for our algorithm to be useful, one should not be needing too many forward $c \rightarrow \bar{c}$ and backward $\bar{c} \rightarrow c$ transformations. Again, this is not a problem for applications we are considering.

Before proceeding, we should also add that both forward and backward transformations are trivially performed using the NewPro algorithm:

$$\begin{array}{lclcl} a \to \bar{a} & : & \mathrm{NewPro}(a, \alpha^{-2}) & = & a \cdot \alpha^{-2} \cdot \alpha & = & \bar{a} \ , \\ \bar{a} \to a & : & \mathrm{NewPro}(\bar{a}, 1) & = & (a \cdot \alpha^{-1}) \cdot 1 \cdot \alpha & = & a \ . \end{array} \tag{16}$$

For these computations, we need α^{-2}, which is easily obtained from the normal representation of α^{-1} by a left rotation of the digits. We also need the normal representation of the unity element, which is given as $(11 \cdots 1) = \sum_{i=0}^{k-1} \beta^{2^i}$ for any normal element β.

Also, if two elements are expressed in the bar domain, then their additions produce the output in the bar domain, that is

$$\bar{a} + \bar{b} = a \cdot \alpha^{-1} + b \cdot \alpha^{-1} = (a + b) \cdot \alpha^{-1} = c \cdot \alpha^{-1} = \bar{c} \ .$$

Finally we should remark that, similar to (11), the NewPro function for computing $\bar{c} = \bar{a} \cdot \bar{b} \cdot \alpha$ can be expanded as

$$\bar{c} = \left(\sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \bar{a}_i \bar{b}_j \boldsymbol{\lambda}_{ij} \right) \cdot \alpha \ = \ \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \bar{a}_i \bar{b}_j (\alpha \boldsymbol{\lambda}_{ij}) \ .$$

This is the usual normal basis multiplication, however, the matrix involved is $\alpha\boldsymbol{\lambda}$, instead of just $\boldsymbol{\lambda}$, for a fixed element α of the field $\mathrm{GF}(2^k)$. The matrix $\boldsymbol{\lambda}$ has $k(2k-1)$ terms of type β^{2^i}, which determines the number of 2-input XOR gates as $2k-2$ for computing each component of c_r for $0 \le r \le k-1$.

In order to have a reduced complexity (in terms of the XOR gates) normal basis multiplication, we need to show that there exists a special element α of $\mathrm{GF}(2^k)$ for which $\alpha\boldsymbol{\lambda}$ has fewer than $k(2k-1)$ terms of type β^{2^i}. We will show the construction of α and the analyses for the fields $\mathrm{GF}(2^2)$, $\mathrm{GF}(2^3)$, and $\mathrm{GF}(2^4)$ below, and then describe the general cases.

4.1 Complexity of NewPro Multiplication in GF(2^2)

The proposed NewPro algorithm requires the existence of a special element α of $\mathrm{GF}(2^k)$ such that the matrix $\alpha\boldsymbol{\lambda}$ has fewer than $k(2k-1)$ terms of type β^{2^i}. Since $\mathrm{GF}(2^2)$ has only two suitable elements β and β^2, we can easily try each one to see if the number of terms in the matrix $\alpha\boldsymbol{\lambda}$ is less than 6. First we consider $\alpha = \beta$:

$$\alpha\boldsymbol{\lambda} = \beta\boldsymbol{\lambda} = \beta \begin{bmatrix} \beta^2 & \beta + \beta^2 \\ \beta + \beta^2 & \beta \end{bmatrix} = \begin{bmatrix} \beta^3 & \beta^2 + \beta^3 \\ \beta^2 + \beta^3 & \beta^2 \end{bmatrix} = \begin{bmatrix} \beta + \beta^2 & \beta \\ \beta & \beta^2 \end{bmatrix}$$

Indeed the resulting $\alpha\boldsymbol{\lambda}$ has 5 terms, instead of 6. This gives the $\alpha\boldsymbol{\lambda}^{(r)}$ matrices as

$$\alpha\boldsymbol{\lambda} = \beta\boldsymbol{\lambda} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \beta + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \beta^2$$

We obtain the individual components of the product $\bar{c}$ as

$$\begin{array}{rcl} \bar{c}_0 & = & \bar{a}_0\bar{b}_0 + \bar{a}_0\bar{b}_1 + \bar{a}_1\bar{b}_0 \ , \\ \bar{c}_0 & = & \bar{a}_0\bar{b}_0 + \bar{a}_1\bar{b}_0 \ . \end{array}$$

Therefore we showed that the NewPro multiplication $\bar{c} = \text{NewPro}(\bar{a}, \bar{b}) = \bar{a}\bar{b}\alpha$ requires only 3 2-input XOR gates using the above formulae with the selection of $\alpha = \beta$, instead of 4 2-input XOR gates required by the normal product computation $c = ab$, as given by the formulae in Eqn. (3).

It turns out that $\alpha = \beta^2$ also reduces the complexity:

$$\alpha\boldsymbol{\lambda} = \beta^2\boldsymbol{\lambda} = \beta^2 \begin{bmatrix} \beta^2 & \beta + \beta^2 \\ \beta + \beta^2 & \beta \end{bmatrix} = \begin{bmatrix} \beta^4 & \beta^3 + \beta^4 \\ \beta^3 + \beta^4 & \beta^3 \end{bmatrix} = \begin{bmatrix} \beta & \beta^2 \\ \beta^2 & \beta + \beta^2 \end{bmatrix}$$

This gives the $\alpha\boldsymbol{\lambda}^{(r)}$ matrices as

$$\alpha\boldsymbol{\lambda} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \beta + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \beta^2$$

We obtain the individual components of the product $\bar{c}$ as

$$\begin{aligned} \bar{c}_0 &= \bar{a}_0\bar{b}_0 + \bar{a}_1\bar{b}_1 \ , \\ \bar{c}_0 &= \bar{a}_0\bar{b}_1 + \bar{a}_1\bar{b}_0 + \bar{a}_1\bar{b}_1 \ . \end{aligned}$$

The NewPro multiplication $\bar{c} = \text{NewPro}(\bar{a}, \bar{b}) = \bar{a}\bar{b}\alpha$ requires only 3 2-input XOR gates using the above formulae with the selection of $\alpha = \beta^2$, instead of 4 2-input XOR gates required by the regular normal basis multiplication $c = ab$, as given by the formulae in Eqn. (3).

4.2 Complexity of NewPro Multiplication in GF(2^3)

The proposed NewPro algorithm requires the existence of a special element α of GF(2^k) such that the matrix $\alpha\boldsymbol{\lambda}$ has fewer than $k(2k-1)$ terms of type β^{2^i}. For a small field such as GF(2^3), we can try all possible candidates for α. Since our construction excludes α as 0 or 1, we need to try only 6 different α values in GF(2^3). We have performed this search using a simple Mathematica code, and obtained the number of elements in the $\alpha\boldsymbol{\lambda}$ matrix for each value of α, as shown in Table 2. Note that without the transformation, the matrix $\boldsymbol{\lambda}$ has $k(2k-1) = 15$ terms.

Table 2: The number of terms in the matrix $\alpha\boldsymbol{\lambda}$ for GF(2^3).

α	Terms
β	17
β^2	17
β^4	17
$\beta + \beta^2$	14
$\beta + \beta^4$	14
$\beta^2 + \beta^4$	14

The search shows that for $\alpha = \beta + \beta^2$, $\alpha = \beta + \beta^4$, and $\alpha = \beta^2 + \beta^4$ values the matrix $\alpha\boldsymbol{\lambda}$ has only 14 terms, which is 1 fewer than 15, the optimal value for the matrix $\boldsymbol{\lambda}$.

We show how to obtain the $\alpha\boldsymbol{\lambda}$ matrix for $\alpha = \beta + \beta^2$. We first multiply $(\beta + \beta^2)$ with every term of the matrix $\boldsymbol{\lambda}$, and then substitute the powers of β which are not powers of 2, using the normal representations of all elements given in Section 3.1.

$$\alpha\boldsymbol{\lambda} = (\beta + \beta^2) \begin{bmatrix} \beta^2 & \beta + \beta^4 & \beta^2 + \beta^4 \\ \beta + \beta^4 & \beta^4 & \beta + \beta^2 \\ \beta^2 + \beta^4 & \beta + \beta^2 & \beta \end{bmatrix} = \begin{bmatrix} \beta & \beta^2 & \beta^4 \\ \beta^2 & \beta + \beta^4 & \beta^2 + \beta^4 \\ \beta^4 & \beta^2 + \beta^4 & \beta + \beta^2 + \beta^4 \end{bmatrix} .$$

As is observed, this matrix has 14 terms. We can expand this matrix in terms of $\boldsymbol{\lambda}^{(r)}$ matrices and powers of β, to obtain

$$\alpha\boldsymbol{\lambda} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\beta + \begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}\beta^2 + \begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}\beta^4 \ .$$

Since we destroyed the shift property (Eqn. 10) of $\boldsymbol{\lambda}^{(r)}$ matrices by multiplying α with $\boldsymbol{\lambda}$, these matrices no longer have the same number of 1s, however, the total number of 1s is $3+5+6=14$, instead of $5+5+5=15$. Finally, we obtain the individual components of the $\bar{c}$ vector as

$$\begin{aligned}\bar{c}_0 &= \bar{a}_0\bar{b}_0 + \bar{a}_1\bar{b}_1 + \bar{a}_2\bar{b}_2 \ ,\\ \bar{c}_1 &= \bar{a}_0\bar{b}_1 + \bar{a}_1\bar{b}_0 + \bar{a}_1\bar{b}_2 + \bar{a}_2\bar{b}_1 + \bar{a}_2\bar{b}_2 \ ,\\ \bar{c}_2 &= \bar{a}_0\bar{b}_2 + \bar{a}_1\bar{b}_1 + \bar{a}_1\bar{b}_2 + \bar{a}_2\bar{b}_0 + \bar{a}_2\bar{b}_1 + \bar{a}_2\bar{b}_2 \ .\end{aligned}$$

which requires 11 2-input XOR gates, instead of 12. The other two α values also produce the $\boldsymbol{\lambda}$ matrices with exactly 14 terms, and the associated formulae for the components $\bar{c}_r$ are easily obtained.

4.3 Complexity of NewPro Multiplication in GF(2^4)

Given the irreducible polynomial $p(x) = x^4+x+1$ generating the field GF(2^4), we obtain an optimal normal element $\beta = x^3$ using the construction in Theorem 1. This field has Type 1 optimal normal basis since $k+1=5$ is prime, and 2 is primitive in $\mathcal{Z}_5$.

Table 3: The normal and the powers of $\beta = x^3$ representations
of elements in GF(2^4) with irreducible polynomial $p(x) = x^4 + x + 1$.

$\beta^8+\beta^4+\beta^2+\beta$	β^0	β^8	β^8
β	β^1	β^9	β^4
β^2	β^2	β^{10}	$\beta^8+\beta^4+\beta^2+\beta$
β^8	β^3	β^{11}	β
β^4	β^4	β^{12}	β^2
$\beta^8+\beta^4+\beta^2+\beta$	β^5	β^{13}	β^8
β	β^6	β^{14}	β^4
β^2	β^7	β^{15}	$\beta^8+\beta^4+\beta^2+\beta$

The 4×4 $\boldsymbol{\lambda}$ matrix is obtained as

$$\begin{bmatrix}\beta^2 & \beta^3 & \beta^5 & \beta^9\\ \beta^3 & \beta^4 & \beta^6 & \beta^{10}\\ \beta^5 & \beta^6 & \beta^8 & \beta^{12}\\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16}\end{bmatrix} = \begin{bmatrix}\beta^2 & \beta^8 & \beta^8+\beta^4+\beta^2+\beta & \beta^4\\ \beta^8 & \beta^4 & \beta & \beta^8+\beta^4+\beta^2+\beta\\ \beta^8+\beta^4+\beta^2+\beta & \beta & \beta^8 & \beta^2\\ \beta^4 & \beta^8+\beta^4+\beta^2+\beta & \beta^2 & \beta\end{bmatrix}$$

The number of terms in the $\boldsymbol{\lambda}$ matrix for an optimal basis $\beta \in$ GF(2^4) is equal to $k(2k-1) = 28$. The matrix $\boldsymbol{\lambda}$ can be expanded as in terms of $\boldsymbol{\lambda}^{(r)}$ matrices and powers of 2 powers of β as

$$\boldsymbol{\lambda} \;=\; \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 1\end{bmatrix}\beta + \begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0\end{bmatrix}\beta^2 + \begin{bmatrix}0 & 0 & 1 & 1\\ 0 & 1 & 0 & 1\\ 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\end{bmatrix}\beta^4 + \begin{bmatrix}0 & 1 & 1 & 0\\ 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\beta^8 \ .$$

The total number of 1s in these matrices is also 28, each of which has 7 1s. Therefore, we need 6 2-input XOR gates to computes each one of the c_r terms for $r = 0, 1, 2, 3$, which totals to 24 XOR gates. Similar to the case GF(2^3), we performed an exhaustive search over the set GF(2^4) and obtained the list of α values, and the minimum number of terms in the matrix $\alpha\boldsymbol{\lambda}$, as summarized in Table 4.

Table 4: The minimum number of terms in the matrix $\alpha\boldsymbol{\lambda}$ for GF(2^4).

α	Terms
β	25
β^2	25
β^4	25
β^8	25

It turns out that there are only 4 α values, which minimize the number of terms in the matrix $\alpha\boldsymbol{\lambda}$; the minimum value is found as 25. We obtain the matrix $\alpha\boldsymbol{\lambda}$ for $\alpha = \beta$, by multiplying every element of the matrix by α. The elements of the matrix which contains the powers of β which are not powers of 2 are then replaced with their normal expansions.

$$\alpha\boldsymbol{\lambda} \;=\; \begin{bmatrix} \beta^8 & \beta^4 & \beta & \beta+\beta^2+\beta^4+\beta^8 \\ \beta^4 & \beta+\beta^2+\beta^4+\beta^8 & \beta^2 & \beta \\ \beta & \beta^2 & \beta^4 & \beta^8 \\ \beta+\beta^2+\beta^4+\beta^8 & \beta & \beta^8 & \beta^2 \end{bmatrix} .$$

The matrix $\alpha\boldsymbol{\lambda}$ has exactly 25 terms. It can be expanded in terms of $\boldsymbol{\lambda}^{(r)}$ matrices and powers of 2 powers of β as

$$\alpha\boldsymbol{\lambda} \;=\; \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \beta + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \beta^2 + \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \beta^4 + \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \beta^8 \; .$$

The total number of 1s in these matrices is also equal to 25. The number of 2-input XOR gates to compute $\bar{c}_r$ terms for $r = 0, 1, 2, 3$ is found as $6 + 5 + 5 + 5 = 21$.

5 The General Case for GF(2^k)

The NewPro transformation and the multiplication algorithms require the existence of an element α of GF(2^k), that minimizes the number of terms in the $\alpha\boldsymbol{\lambda}$ matrix. We gave detailed analyses of the NewPro multiplication for the fields GF(2^2), GF(2^3) and GF(2^4) together with all special α values. It turns out that the number of terms in the $\alpha\boldsymbol{\lambda}$ matrix are equal to 5, 14 and 25 for GF(2^2), GF(2^3) and GF(2^4) respectively, while the original $\boldsymbol{\lambda}$ matrices have 6, 15 and 28 terms.

However, we need a more detailed analysis of the proposed NewPro algorithm, specifically, we identify the following types of problems to study:

1. Due to the optimal normal basis theorem, we know that $\boldsymbol{\lambda}$ has $k(2k-1)$ terms for GF(2^k). What is the exact number of terms in the $\alpha\boldsymbol{\lambda}$ matrix for GF(2^k) for different values of k and different (Type 1 and 2) bases?

2. Does there exist an α for any k such that number of terms in $\alpha\boldsymbol{\lambda}$ is less than $k(2k-1)$?
3. Is there a constructive or non-exhaustive method for finding α that reduces the number of terms to fewer than $k(2k-1)$?

The answers to these questions for Type 2 optimal normal bases seem to be negative for $k > 3$. For such normal bases, no α can bring down the number of terms in $\alpha\boldsymbol{\lambda}$ to a quantity below $k(2k-1)$. However, we settle the above questions for Type 1 optimal normal bases in the following fashion.

6 Optimality for the Type 1 Case

Let us assume that GF(2^k) has a Type 1 optimal normal basis; this implies that $k+1$ is prime and 2 is primitive in $\mathcal{Z}_{k+1}$. Moreover, the optimal normal element β is a primitive $(k+1)$st root of 1 in $GF(2^k)$. For the brevity of the notation, we write $k+1 = 2m+1$, $\beta_i = \beta^{2^i}$, and $\mathbf{1}$ stands for the unity element in normal basis:

$$\mathbf{1} = \beta + \beta^2 + \beta^4 + \cdots + \beta^{2^{k-1}} = \beta_0 + \beta_1 + \beta_2 + \cdots + \beta_{k-1} \ .$$

We also use $\mathcal{B}$ to represent the basis set $\mathcal{B} = \{\beta_0, \beta_1, \ldots, \beta_{k-1}\}$. As before, the $k \times k$ matrix $\boldsymbol{\lambda}$ is defined as

$$\boldsymbol{\lambda}_{ij} = \beta^{2^i+2^j} = \beta_i\beta_j$$

for $0 \le i, j \le k-1$. For example, for $k = 4$, we have

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \end{bmatrix} \ .$$

Lemma 1. *The elements in the entries $(0, m), (1, m+1), \ldots, (k-1, m+k-1)$ of $\boldsymbol{\lambda}$, where the indices are computed mod k, are all $\mathbf{1}$s.*

Proof. What we need to show is $\beta_i\beta_{m+i} = \mathbf{1}$ for $0 \le i \le k-1$, where the indices are computed mod k. Let $\theta_i = \beta_i\beta_{m+i}$, and put $\theta = \theta_0 = \beta_0\beta_m$. Then

$$\theta_i = \beta^{2^i+2^{m+i}} = \left(\beta^{2^m+1}\right)^{2^i} = \theta^{2^i} \ .$$

Therefore it suffices to show that $\theta = \mathbf{1}$. Calculating,

$$\theta^{2^m} = \beta^{2^m}\beta^{2^{2m}} = \beta_m\beta_0 = \theta \ ,$$

so that $\theta^{2^m-1} = \mathbf{1}$. On the other hand, θ is a power of β and $\beta^{2m+1} = \mathbf{1}$, so $\theta^{2m+1} = \mathbf{1}$. Therefore the order of θ divides $d = \gcd(2^m - 1, 2m+1)$. Since $p = 2m+1$ is prime, d is either 1 or p. But $2^m - 1 = 2^{\frac{p-1}{2}} - 1$ and this cannot be divisible by p for otherwise 2 is a quadratic residue modulo p and so cannot be primitive. Therefore, $d = 1$ and $\theta = \mathbf{1}$. □

The entry $\beta_i\beta_{m+i} = \beta_0 + \beta_1 + \cdots + \beta_{k-1}$ contributes k to the sum of the number of basis vectors appearing in row i. Since the basis is optimal, the total number of these is $2k-1$. Each of the remaining $k-1$ entries is a single β_j. By optimality, the elements in row i excluding the unit in column $m+i$ is a permutation of $\beta_0, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots, \beta_{k-1}$. We record the fact that the elements in row r of $\boldsymbol{\lambda}$ is a permutation of the elements $\mathcal{B} - \{\beta_r\}$.

Lemma 2. *For an optimal normal basis of Type 1 with $k+1 = 2m+1$, generated by $\beta = \beta_0$, the row r for $0 \leq r \leq k-1$ of $\boldsymbol{\lambda}$ is a permutation of $\mathcal{B} - \{\beta_r\}$ with* **1** *appearing in the column index $m+r$ modulo k. Therefore $\beta_r \cdot \{\beta_0, \beta_1, \ldots, \beta_{k-1}\} = \mathcal{B} - \{\beta_r\}$.*

Next, we consider the matrix $\beta_r\boldsymbol{\lambda}$.

Lemma 3. *Row $m+r$ of $\beta_r\boldsymbol{\lambda}$ is a permutation of $\mathcal{B}$. Each of the other rows is a permutation of $\mathcal{B}$ minus some basis element.*

Proof. We will give the proof for $r = 0$. Note that every row in $\boldsymbol{\lambda}$ has a **1** (the entries in positions $(0, m), (1, m+1), \ldots, (k-1, m+k-1)$ are **1**s), so in $\beta\boldsymbol{\lambda}$, β_0 appears in each row. By Lemma 2, the rth row of $\boldsymbol{\lambda}$ is a permutation of $\mathcal{B} - \{\beta_r\}$. Therefore the rth row of $\beta\boldsymbol{\lambda}$ is

1. a permutation of $\mathcal{B} - \{1\}$ for $r = m$,

2. a permutation of $\mathcal{B} - \{\beta_0\beta r\}$ for $r \neq m$ (note: $\beta_0\beta_r = \beta_j$ for some j in this case).

□

Therefore, we conclude that the total number of basis vectors appearing in the matrix is

$$k + (k-1)(2k-1) = k(2k-1) - (k-1) \ ,$$

which is $k-1$ fewer than that of the multiplication matrix $\boldsymbol{\lambda}$. We state the following theorem for the Type 1 case, but omit its proof.

Theorem 2. *Suppose α has t nonzero coefficients in its normal basis expansion. Then the number of terms in the matrix $\alpha\boldsymbol{\lambda}$ is $k(2k-1) + (k-t)\left(t(2k-2) - (2k-1)\right)$. In particular $\alpha = \beta_i$ for $0 \leq i \leq k-1$ are the only αs with the property that $\alpha\boldsymbol{\lambda}$ matrix has smaller than $k(2k-1)$, i.e., $k(2k-1) - (k-1)$ basis vectors.*

7 Further Work

By slightly changing the definition of the multiplication operation, we introduced a new normal basis multiplication algorithm which requires fewer XORs than the optimal normal multiplication algorithm. We proved for the Type 1 case that the number of terms in the $\alpha\boldsymbol{\lambda}$ is $k-1$ fewer than that of $\boldsymbol{\lambda}$ matrix for $\alpha = \beta_i$ for some $0 \leq i \leq k-1$. Appendix A gives the $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ matrices for GF(2^k) for $k = 2, 4, 10, 12$, which are the first 4 fields that has Type 1 optimal normal bases.

Moreover, experimentation shows that the field GF(2^3) and Type 2 optimal normal basis matrix $\alpha\boldsymbol{\lambda}$ has 14 terms for $\alpha = \beta + \beta^2$, $\alpha = \beta + \beta^4$, and $\alpha = \beta^2 + \beta^4$ instead of 15 terms, which is clearly unlike the Type 1 case, i.e., it is not $k-1 = 2$ fewer than $k(2k-1) = 15$. However the case of GF(2^3) seems to be an anomaly, and it appears that for a Type 2 optimal normal basis GF(2^k) for $k > 3$, there is no α value which gives smaller than $k(2k-1)$ terms in $\alpha\boldsymbol{\lambda}$.

It is also possible to view $\alpha\boldsymbol{\lambda}$ as matrix multiplication by $\alpha\boldsymbol{I}$, so one can study the suitability of transformations defined by premultiplying $\boldsymbol{\lambda}$ by other kinds of matrices.

References

[1] G. B. Agnew, T. Beth, R. C. Mullin, and S. A. Vanstone. Arithmetic operations in $GF(2^m)$. *Journal of Cryptology*, 6(1):3–13, 1993.

[2] G. B. Agnew, R. C. Mullin, I. Onyszchuk, and S. A. Vanstone. An implementation for a fast public-key cryptosystem. *Journal of Cryptology*, 3(2):63–79, 1991.

[3] G. B. Agnew, R. C. Mullin, and S. A. Vanstone. An implementation of elliptic curve cryptosystems over $F_{2^{155}}$. *IEEE Journal on Selected Areas in Communications*, 11(5):804–813, June 1993.

[4] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography.* Cambridge University Press, 1999.

[5] S. S. Erdem, T. Yanık, and Ç. K. Koç. Polynomial basis multiplication in GF(2^m). *Acta Applicandae Mathematicae*, 93(1-3):33–55, September 2006.

[6] S. Gao. *Normal Bases over Finite Fields.* PhD thesis, University of Waterloo, 1993.

[7] S. Gao and H. W. Lenstra, Jr. Optimal normal bases. *Designs, Codes and Cryptography*, 2(4):315–323, December 1992.

[8] A. Halbutoğulları and Ç. K. Koç. Mastrovito multiplier for general irreducible polynomials. *IEEE Transactions on Computers*, 49(5):503–518, May 2000.

[9] M. A. Hasan, M. Z. Wang, and V. K. Bhargava. Modular construction of low complexity parallel multipliers for a class of finite fields $GF(2^m)$. *IEEE Transactions on Computers*, 41(8):962–971, August 1992.

[10] T. Itoh and S. Tsujii. A fast algorithm for computing multiplicative inverses in $GF(2^m)$ using normal bases. *Information and Computation*, 78(3):171–177, September 1988.

[11] T. Itoh and S. Tsujii. Structure of parallel multipliers for a class of finite fields $GF(2^m)$. *Information and Computation*, 83:21–40, 1989.

[12] Ç. K. Koç and T. Acar. Montgomery multiplication in GF(2^k). *Designs, Codes and Cryptography*, 14(1):57–69, April 1998.

[13] Ç. K. Koç, T. Acar, and B. S. Kaliski Jr. Analyzing and comparing Montgomery multiplication algorithms. *IEEE Micro*, 16(3):26–33, June 1996.

[14] E. D. Mastrovito. VLSI architectures for multiplication over finite field GF(2^m). In T. Mora, editor, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 297–309. Springer, LNCS Nr. 357, 1988.

[15] E. D. Mastrovito. *VLSI Architectures for Computation in Galois Fields.* PhD thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden, 1991.

[16] P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, April 1985.

[17] R. Mullin, I. Onyszchuk, S. Vanstone, and R. Wilson. Optimal normal bases in $GF(p^n)$. *Discrete Applied Mathematics*, 22:149–161, 1988.

[18] J. Omura and J. Massey. Computational method and apparatus for finite field arithmetic, May 1986. U.S. Patent Number 4,587,627.

[19] C. Paar. A new architecture for a paralel finite field multiplier with low complexity based on composite fields. *IEEE Transactions on Computers*, 45(7):856–861, July 1996.

[20] A. Reyhani-Masoleh and M. A. Hasan. A new construction of Massey-Omura parallel multiplier over GF(2^m). *IEEE Transactions on Computers*, 51(5):511–520, May 2001.

[21] G. Saldamlı. *Spectral Modular Arithmetic*. PhD thesis, Oregon State University, 2005.

[22] G. Saldamlı, Y.-J. Baek, and Ç. K. Koç. Spectral modular arithmetic for binary extension fields. In *The 2011 International Conference on Information and Computer Networks (ICICN)*, pages 323–328, 2011.

[23] G. Seroussi. Table of low-weight binary irreducible polynomials, August 1998. Hewlett-Packard, HPL-98-135.

[24] J. H. Silverman. Fast multiplication in finite fields GF(2^n). In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 1999*, pages 122–134. Springer, LNCS Nr. 1965, 1999.

[25] B. Sunar and Ç. K. Koç. Mastrovito multiplier for all trinomials. *IEEE Transactions on Computers*, 48(5):522–527, May 1999.

[26] J. von zur Gathen, A. Shokrollahi, and J. Shokrollahi. Efficient multiplication using type 2 optimal normal bases. In C. Carlet and B. Sunar, editors, *Arithmetic of Finite Fields, WAIFI 2007*, pages 55–68. Springer, LNCS Nr. 4547, 2005.

[27] H. Wu and M. A. Hasan. Low complexity bit-parallel multipliers for a class of finite fields. *IEEE Transactions on Computers*, 47(8):883–887, August 1998.

[28] T. Zhang and K. K. Parhi. Systematic design of original and modified Mastrovito multipliers for general irreducible polynomials. *IEEE Transactions on Computers*, 50(7):734–749, July 2001.

Appendix A: The Type 1 $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ Matrices

The $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ Matrices for GF(2^2)

The irreducible polynomial is $x^2 + x + 1$. The optimal normal element is $\beta = x$. The total count of 1s in $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ matrices are 6 and 5, respectively.

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta_1 & \beta_0 \\ \beta_0 & \beta_1 \end{bmatrix} \;,\; \beta_0\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{1} & \beta_0 \\ \beta_0 & \beta_1 \end{bmatrix} \;,\; \beta_1\boldsymbol{\lambda} = \begin{bmatrix} \beta_0 & \beta_1 \\ \beta_1 & \mathbf{1} \end{bmatrix}$$

The $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ Matrices for GF(2^4)

The irreducible polynomial is $x^4 + x^3 + 1$. The optimal normal element is $\beta = x + 1$. The total count of 1s in $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ matrices are 28 and 25, respectively.

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \end{bmatrix} \;,\; \beta_0\boldsymbol{\lambda} = \begin{bmatrix} \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \end{bmatrix} \;,\; \beta_1\boldsymbol{\lambda} = \begin{bmatrix} \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 \end{bmatrix}$$

$$\beta_2\boldsymbol{\lambda} = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 & \beta_3 \\ \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \end{bmatrix} \;,\; \beta_3\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \end{bmatrix}$$

The $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ Matrices for GF(2^{10})

The irreducible polynomial is $x^{10} + x^7 + 1$. The optimal normal element is $\beta = x^6 + x^3 + x^2 + x$. The total count of 1s in $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ matrices are 190 and 181, respectively.

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta_1 & \beta_8 & \beta_4 & \beta_6 & \beta_9 & \mathbf{1} & \beta_5 & \beta_3 & \beta_2 & \beta_7 \\ \beta_8 & \beta_2 & \beta_9 & \beta_5 & \beta_7 & \beta_0 & \mathbf{1} & \beta_6 & \beta_4 & \beta_3 \\ \beta_4 & \beta_9 & \beta_3 & \beta_0 & \beta_6 & \beta_8 & \beta_1 & \mathbf{1} & \beta_7 & \beta_5 \\ \beta_6 & \beta_5 & \beta_0 & \beta_4 & \beta_1 & \beta_7 & \beta_9 & \beta_2 & \mathbf{1} & \beta_8 \\ \beta_9 & \beta_7 & \beta_6 & \beta_1 & \beta_5 & \beta_2 & \beta_8 & \beta_0 & \beta_3 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_8 & \beta_7 & \beta_2 & \beta_6 & \beta_3 & \beta_9 & \beta_1 & \beta_4 \\ \beta_5 & \mathbf{1} & \beta_1 & \beta_9 & \beta_8 & \beta_3 & \beta_7 & \beta_4 & \beta_0 & \beta_2 \\ \beta_3 & \beta_6 & \mathbf{1} & \beta_2 & \beta_0 & \beta_9 & \beta_4 & \beta_8 & \beta_5 & \beta_1 \\ \beta_2 & \beta_4 & \beta_7 & \mathbf{1} & \beta_3 & \beta_1 & \beta_0 & \beta_5 & \beta_9 & \beta_6 \\ \beta_7 & \beta_3 & \beta_5 & \beta_8 & \mathbf{1} & \beta_4 & \beta_2 & \beta_1 & \beta_6 & \beta_0 \end{bmatrix} \;,\; \beta\boldsymbol{\lambda} = \begin{bmatrix} \beta_8 & \beta_2 & \beta_9 & \beta_5 & \beta_7 & \beta_0 & \mathbf{1} & \beta_6 & \beta_4 & \beta_3 \\ \beta_2 & \beta_4 & \beta_7 & \mathbf{1} & \beta_3 & \beta_1 & \beta_0 & \beta_5 & \beta_9 & \beta_6 \\ \beta_9 & \beta_7 & \beta_6 & \beta_1 & \beta_5 & \beta_2 & \beta_8 & \beta_0 & \beta_3 & \mathbf{1} \\ \beta_5 & \mathbf{1} & \beta_1 & \beta_9 & \beta_8 & \beta_3 & \beta_7 & \beta_4 & \beta_0 & \beta_2 \\ \beta_7 & \beta_3 & \beta_5 & \beta_8 & \mathbf{1} & \beta_4 & \beta_2 & \beta_1 & \beta_6 & \beta_0 \\ \beta_0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 & \beta_9 \\ \mathbf{1} & \beta_0 & \beta_8 & \beta_7 & \beta_2 & \beta_6 & \beta_3 & \beta_9 & \beta_1 & \beta_4 \\ \beta_6 & \beta_5 & \beta_0 & \beta_4 & \beta_1 & \beta_7 & \beta_9 & \beta_2 & \mathbf{1} & \beta_8 \\ \beta_4 & \beta_9 & \beta_3 & \beta_0 & \beta_6 & \beta_8 & \beta_1 & \mathbf{1} & \beta_7 & \beta_5 \\ \beta_3 & \beta_6 & \mathbf{1} & \beta_2 & \beta_0 & \beta_9 & \beta_4 & \beta_8 & \beta_5 & \beta_1 \end{bmatrix}$$

The $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ Matrices for GF(2^{12})

The irreducible polynomial is $x^{12}+x^{10}+x^2+x+1$. The optimal normal element is $\beta = x^{11}+x^7+x^3+x^2+x$. The total count of 1s in $\boldsymbol{\lambda}$ and $\alpha\boldsymbol{\lambda}$ matrices are 276 and 265, respectively.

$$\boldsymbol{\lambda} = \begin{bmatrix}
\beta_1 & \beta_4 & \beta_9 & \beta_8 & \beta_2 & \beta_{11} & \mathbf{1} & \beta_6 & \beta_{10} & \beta_5 & \beta_7 & \beta_3 \\
\beta_4 & \beta_2 & \beta_5 & \beta_{10} & \beta_9 & \beta_3 & \beta_0 & \mathbf{1} & \beta_7 & \beta_{11} & \beta_6 & \beta_8 \\
\beta_9 & \beta_5 & \beta_3 & \beta_6 & \beta_{11} & \beta_{10} & \beta_4 & \beta_1 & \mathbf{1} & \beta_8 & \beta_0 & \beta_7 \\
\beta_8 & \beta_{10} & \beta_6 & \beta_4 & \beta_7 & \beta_0 & \beta_{11} & \beta_5 & \beta_2 & \mathbf{1} & \beta_9 & \beta_1 \\
\beta_2 & \beta_9 & \beta_{11} & \beta_7 & \beta_5 & \beta_8 & \beta_1 & \beta_0 & \beta_6 & \beta_3 & \mathbf{1} & \beta_{10} \\
\beta_{11} & \beta_3 & \beta_{10} & \beta_0 & \beta_8 & \beta_6 & \beta_9 & \beta_2 & \beta_1 & \beta_7 & \beta_4 & \mathbf{1} \\
\mathbf{1} & \beta_0 & \beta_4 & \beta_{11} & \beta_1 & \beta_9 & \beta_7 & \beta_{10} & \beta_3 & \beta_2 & \beta_8 & \beta_5 \\
\beta_6 & \mathbf{1} & \beta_1 & \beta_5 & \beta_0 & \beta_2 & \beta_{10} & \beta_8 & \beta_{11} & \beta_4 & \beta_3 & \beta_9 \\
\beta_{10} & \beta_7 & \mathbf{1} & \beta_2 & \beta_6 & \beta_1 & \beta_3 & \beta_{11} & \beta_9 & \beta_0 & \beta_5 & \beta_4 \\
\beta_5 & \beta_{11} & \beta_8 & \mathbf{1} & \beta_3 & \beta_7 & \beta_2 & \beta_4 & \beta_0 & \beta_{10} & \beta_1 & \beta_6 \\
\beta_7 & \beta_6 & \beta_0 & \beta_9 & \mathbf{1} & \beta_4 & \beta_8 & \beta_3 & \beta_5 & \beta_1 & \beta_{11} & \beta_2 \\
\beta_3 & \beta_8 & \beta_7 & \beta_1 & \beta_{10} & \mathbf{1} & \beta_5 & \beta_9 & \beta_4 & \beta_6 & \beta_2 & \beta_0
\end{bmatrix}$$

$$\beta\boldsymbol{\lambda} = \begin{bmatrix}
\beta_4 & \beta_2 & \beta_5 & \beta_{10} & \beta_9 & \beta_3 & \beta_0 & \mathbf{1} & \beta_7 & \beta_{11} & \beta_6 & \beta_8 \\
\beta_2 & \beta_9 & \beta_{11} & \beta_7 & \beta_5 & \beta_8 & \beta_1 & \beta_0 & \beta_6 & \beta_3 & \mathbf{1} & \beta_{10} \\
\beta_5 & \beta_{11} & \beta_8 & \mathbf{1} & \beta_3 & \beta_7 & \beta_2 & \beta_4 & \beta_0 & \beta_{10} & \beta_1 & \beta_6 \\
\beta_{10} & \beta_7 & \mathbf{1} & \beta_2 & \beta_6 & \beta_1 & \beta_3 & \beta_{11} & \beta_9 & \beta_0 & \beta_5 & \beta_4 \\
\beta_9 & \beta_5 & \beta_3 & \beta_6 & \beta_{11} & \beta_{10} & \beta_4 & \beta_1 & \mathbf{1} & \beta_8 & \beta_0 & \beta_7 \\
\beta_3 & \beta_8 & \beta_7 & \beta_1 & \beta_{10} & \mathbf{1} & \beta_5 & \beta_9 & \beta_4 & \beta_6 & \beta_2 & \beta_0 \\
\beta_0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 & \beta_9 & \beta_{10} & \beta_{11} \\
\mathbf{1} & \beta_0 & \beta_4 & \beta_{11} & \beta_1 & \beta_9 & \beta_7 & \beta_{10} & \beta_3 & \beta_2 & \beta_8 & \beta_5 \\
\beta_7 & \beta_6 & \beta_0 & \beta_9 & \mathbf{1} & \beta_4 & \beta_8 & \beta_3 & \beta_5 & \beta_1 & \beta_{11} & \beta_2 \\
\beta_{11} & \beta_3 & \beta_{10} & \beta_0 & \beta_8 & \beta_6 & \beta_9 & \beta_2 & \beta_1 & \beta_7 & \beta_4 & \mathbf{1} \\
\beta_6 & \mathbf{1} & \beta_1 & \beta_5 & \beta_0 & \beta_2 & \beta_{10} & \beta_8 & \beta_{11} & \beta_4 & \beta_3 & \beta_9 \\
\beta_8 & \beta_{10} & \beta_6 & \beta_4 & \beta_7 & \beta_0 & \beta_{11} & \beta_5 & \beta_2 & \mathbf{1} & \beta_9 & \beta_1
\end{bmatrix}$$