A Matrix Decomposition Method for Optimal Normal Basis Multiplication

Can Kızılkale, Ömer Eğecioğlu, and Çetin Kaya Koç, *Fellow, IEEE*

***Abstract*—We introduce a matrix decomposition method and prove that multiplication in GF(2^k) with a Type 1 optimal normal basis for can be performed using k^2-1 XOR gates irrespective of the choice of the irreducible polynomial generating the field. The previous results achieved this bound only with special irreducible polynomials. Furthermore, the decomposition method performs the multiplication operation using $1.5k(k-1)$ XOR gates for Type 2a and 2b optimal normal bases, which matches previous bounds.**

I. Introduction

The subject of the paper is the multiplication operation in the field GF(2^k) whose elements are represented using a normal basis. Parallel multipliers for GF(2^k) have applications in error-correcting codes [1] for smaller values of k, usually from 16 to 32. Applications in cryptography, for example, elliptic curve cryptographic functions EC-DSA, EC-IES and EC-based random number generators [2], require hardware and software implementations of GF(2^k), but for much larger values of k. Polynomial basis multiplication is probably more suitable for such implementations, though standards (such as ANSI X9.62) suggest normal bases as well [3]. Since optimal normal bases exist only for a smaller subset of k values suggested by cryptographic standards, often sub-optimal Gaussian normal bases are used. For example, the standard ANSI X9.62 suggests selecting a Type 2 basis for GF(2^k), and if this does not exist, then and a Type 1 basis, and if neither exists, then a Type T basis with the smallest value of T.

Furthermore, the new research in elliptic curve cryptography, particularly, Edward curves and its derived versions based on binary fields [4], [5], has shown that GF(2^k) fields coupled with binary Edward curves are highly efficient and secure [5].

II. Preliminaries

All symbols and terms used in this paper are given in Table 1. An element β of the field GF(2^k) is called a normal element if any element $a \in$ GF(2^k) can be uniquely written as a linear sum of the powers of 2 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 $a_i \in \{0,1\}$. For the brevity of the notation, we will interchangeably use $\beta_i = \beta^{2^i}$ for $i = 0,1,\ldots,k-1$, and the denote the basis set by $\mathcal{B} = \{\beta_0, \beta_1, \ldots, \beta_{k-1}\}$. Also we will use $\mathbf{1}$ (boldface 1) to represent the identity element expressed in normal basis, which is equal to the sum of all basis elements:

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

The normal representation of an element in GF(2^k) is particularly useful for squaring; 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 order to describe the normal basis multiplication, we refer to the Massey-Omura algorithm [6], which follows the following steps: Given the bits a_i and b_i of the input operands a and b, the Massey-Omura multiplier first generates all partial product terms a_ib_j for $0 \le i,j \le k-1$ using AND gates, and then sums the subsets of these partial product terms using XOR gates to obtain the bits c_r of the product for $r = 0,1,\ldots,k-1$.

For uniformity of the analysis throughout this paper we assume that AND and XOR gates have 2 inputs, and we denote the individual gate delays by T_A and T_X.

There are k^2 partial product terms a_ib_j, which can be computed using k^2 2-input AND gates in a single T_A delay. This computation is space-optimal; k^2 is both upper and lower bound on the number of partial product terms, because all of them need to be computed.

In the computation of each product term c_r for $0 \le r \le k-1$, we need only a subset of the k^2 partial product terms a_ib_j. According to the optimality theorem of the normal basis multiplication [7], the number of a_ib_j terms needed to compute any of c_r is at least $2k-1$. If there exists a normal basis in GF(2^k) for which the number of a_ib_j terms for computing c_r is exactly $2k-1$, then this normal basis is called *optimal*. In this case, a c_r term can be computed using $2k-2$ XOR gates, while all c_r terms for $r = 0,1,\ldots,k-1$ would require $k(2k-2)$ XOR gates for optimal normal bases. However, this is an upper bound as there are common a_ib_j terms among the computations of c_r terms for different r values. It is shown that certain subsets of GF(2^k) fields, for example, those generated by irreducible all-one-polynomials [8], [9], require only k^2-1 XOR gates. This paper introduces a matrix decomposition method which requires k^2-1 XOR gates for the Type 1 optimal normal basis, irrespective of the choice the irreducible polynomial. Moreover the method is applicable to Type 2a and 2b bases as well, requiring $1.5k(k-1)$ XOR gates, which matches certain previous bounds [10], [11].

The authors are with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: {ckizilkale, omer, koc}@cs.ucsb.edu

Symbol or Term	Meaning
k	A nonzero positive integer
GF(2^k)	Galois field of 2^k elements
a, b, c	Arbitrary elements of GF(2^k)
a_i, b_i, c_i	Binary coefficients of a, b, c
β	A normal element of GF(2^k)
β_i	Equals to β^{2^i}
$\mathcal{B}$	The basis set $\{\beta_0, \ldots, \beta_{k-1}\}$
$\mathcal{Z}_{k+1}$	The set of integers $\{0, 1, \ldots, k\}$
p	A prime number
$\mathcal{Z}_p^*$	The set of integers $\{1, 2, \ldots, p-1\}$
γ	The primitive pth root of identity
$\gamma + \gamma^{-1}$	Equals to the normal element β
$\boldsymbol{\lambda}$	The $k \times k$ matrix; sum subsets of $\mathcal{B}$
$\boldsymbol{\lambda}$	$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0\beta_0 + \boldsymbol{\lambda}_1\beta_1 + \cdots + \boldsymbol{\lambda}_{k-1}\beta_{k-1}$
$\boldsymbol{\lambda}_{ij}$	The (i, j) entry of the matrix $\boldsymbol{\lambda}$
$\boldsymbol{\lambda}_{ij}$	Equals to $\beta^{2^i+2^j} = \beta^{2^i}\beta^{2^j} = \beta_i\beta_j$
$\boldsymbol{\lambda}_i$	The $k \times k$ matrices with entries $\{0, 1\}$
Q_p	The set of quadratic residues mod p
Q'_p	The set of quadratic nonresidues mod p
T_A and T_X	Delays of 2-input AND or XOR gates

Table 1: All symbols and terms used in this paper.

III. Optimal Normal Bases

The constructions of optimal normal bases are described in [7], [12], [13], and summarized in the following theorem:

Theorem 1: An optimal normal basis for GF(2^k) exists in either of the following cases, and can be constructed as:

1) If $k+1$ is prime and 2 is a primitive element in $\mathcal{Z}_{k+1}$. Each of the k nonunit $(k+1)$th root of identity forms an optimal normal basis in GF(2^k).
2) If $p = 2k+1$ is prime and
   2a: Either, 2 is primitive in $\mathcal{Z}_p^*$;
   2b: Or, $2k+1 = 3 \pmod 4$ and 2 generates quadratic residues in $\mathcal{Z}_p^*$.

   In this case, $\beta = \gamma + \gamma^{-1}$ generates an optimal normal basis in GF(2^k), where γ is a primitive pth root of identity.

The optimal normal bases that are derived from the first part of the theorem are named Type 1, while the ones that follow from the second part are named Type 2 bases, or more specifically, as Type 2a and Type 2b bases. For $k \leq 30$, the optimal normal bases are listed in Table 2.

	k values
Type 1	2, 4, 10, 12, 18, 28
Type 2a	2, 5, 6, 9, 14, 18, 26, 29, 30
Type 2b	3, 11, 23

Table 2: The optimal normal bases for $k \leq 30$.

IV. Normal Basis Multiplication Algorithm

Given the input operands a and b as

$$a = \sum_{i=0}^{k-1} a_i\beta_i \quad , \quad b = \sum_{i=0}^{k-1} b_i\beta_i \ ,$$

the multiplication algorithm computes each bit of the product c, which can be written as a double summation as

$$c = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1} a_i b_j \beta_i \beta_j \ .$$

This in turn can be written as a vector-matrix product

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{k-1} \end{bmatrix} \boldsymbol{\lambda} \begin{bmatrix} b_0 & b_1 & \cdots & b_{k-1} \end{bmatrix}^T \ ,$$

such that every element of the $k \times k$ matrix $\boldsymbol{\lambda}$ is the sum of a subset of the normal elements $\{\beta_0, \beta_1, \beta_2, \ldots, \beta_{k-1}\}$. Furthermore, the $\boldsymbol{\lambda}$ matrix can be expressed in terms of the $k \times k$ matrices $\boldsymbol{\lambda}_i$ for $i = 0, 1, \ldots, k-1$ with entries in $\{0, 1\}$ such that

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0\beta_0 + \boldsymbol{\lambda}_1\beta_1 + \boldsymbol{\lambda}_2\beta_2 + \cdots + \boldsymbol{\lambda}_{k-1}\beta_{k-1} \ .$$

V. Direct Multiplication in GF(2^2)

Consider the smallest extension field GF(2^2), which has both Type 1 and Type 2 of optimal normal bases. We will use the Type 1 optimal normal element $\beta = x$ and the irreducible polynomial $p(x) = x^2 + x + 1$, and derive the $\boldsymbol{\lambda}$ matrix. Given the normal representations of two elements of the field $a = a_0\beta_0 + a_1\beta_1$ and $b = b_0\beta_0 + b_1\beta_1$, their product c is given as

$$\begin{aligned} c &= a_0b_0\beta_0^2 + a_0b_1\beta_0\beta_1 + a_1b_0\beta_0\beta_1 + a_1b_1\beta_1^2 \\ &= a_0b_0\beta_1 + a_0b_1(\beta_0 + \beta_1) + a_1b_0(\beta_0 + \beta_1) + a_1b_1\beta_0 \ , \end{aligned}$$

where the equalities $\beta_0^2 = \beta_1$, $\beta_0\beta_1 = \beta_0 + \beta_1$, and $\beta_1^2 = \beta_0$ are obtained using the normal element $\beta = x$ and the irreducible polynomial $p(x) = x^2 + x + 1$. The vector-matrix expansion of the product can be written as

$$c = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \ ,$$

which gives us the $\boldsymbol{\lambda}$ matrix as

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix} \ .$$

Furthermore, we obtain the $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$ matrices for GF(2^2) as

$$\begin{aligned} \boldsymbol{\lambda} = \boldsymbol{\lambda}_0\beta_0 + \boldsymbol{\lambda}_1\beta_1 &= \begin{bmatrix} \beta_1 & \beta_0 + \beta_1 \\ \beta_0 + \beta_1 & \beta_0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\beta_0 + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\beta_1 \ . \end{aligned}$$

Once all partial products a_ib_j for $0 \leq i, j \leq k-1$ are computed using k^2 AND gates, the $\boldsymbol{\lambda}_i$ matrices determine which subsets of the partial products a_ib_j are to be summed to obtain a particular product term c_r. For GF(2^2), we have

$$c_0 = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0b_1 + a_1b_0 + a_1b_1 \tag{1}$$

$$c_1 = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0b_0 + a_0b_1 + a_1b_0 \tag{2}$$

There are 3 1s in each of the $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$ matrices, and therefore, there 3 terms partial product terms a_ib_j in the expressions for c_0 or c_1. The total number of XOR gates to compute both of c_0 and c_1 is $2 \cdot 2 = 4$.

VI. Matrix Decomposition Method for GF(2^2)

However, we observe a certain similarity in the $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$ matrices: each can be written as the sum of two matrices such that the first matrix is the same for both, in other words,

$$\boldsymbol{\lambda}_0 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} , \tag{3}$$

$$\boldsymbol{\lambda}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} . \tag{4}$$

This matrix decomposition implies that the computation of c_0 and c_1 can be performed in two steps: the first step involves a common matrix for both c_0 and c_1, and while the second steps involve two different matrices.

$$\begin{aligned} c_0 &= \begin{bmatrix} a_0 & a_1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ &= \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} , \end{aligned}$$

$$\begin{aligned} c_1 &= \begin{bmatrix} a_0 & a_1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ &= \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} . \end{aligned}$$

The first vector-matrix product needs to be performed only once for both c_0 and c_1, followed by the second vector-matrix products which need to performed separately for each c_0 and c_1. After these steps, we need add the partial sums to get c_0 and c_1. Therefore, our algorithm for GF(2^2) follows the following steps:

- Step 1: First, we compute the common partial product term, which requires one XOR gate and one T_X delay:

$$s = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0 b_1 + a_1 b_0 . \tag{5}$$

- Step 2: Now, we use the decomposition of $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$ to compute t_0 and t_1; this step does not require any XOR gates and any delay:

$$t_0 = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_1 b_1 , \tag{6}$$

$$t_1 = \begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = a_0 b_0 . \tag{7}$$

- Step 3: Finally we compute c_0 and c_1 using $c_0 = s + t_0$ and $c_1 = s + t_1$. This step requires one XOR gate and one T_X delay.

The matrix decomposition method for GF(2^2) reduces the number of XOR gates to 3, while the direct computation using the formulae (1) and (2) imply 4 XOR gates. The total gate delay is $T_A + 2T_X$.

VII. Matrix Decomposition Method for GF(2^4)

The success of the decomposition method in GF(2^k) depends on the the additive components the $\boldsymbol{\lambda}_i$ matrices, i.e., whether they have common terms among the expressions for c_r. We now consider the field GF(2^4) with the Type 1 optimal normal basis $\beta = x^3$ and the irreducible polynomial $p(x) = x^4 + x + 1$. The normal representations of the powers of β can be obtained by powering β and reducing the resulting polynomials mod $p(x)$, as shown in [14]. The resulting $\boldsymbol{\lambda}$ matrix is

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_3 & \mathbf{1} & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & \mathbf{1} \\ \mathbf{1} & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & \mathbf{1} & \beta_1 & \beta_0 \end{bmatrix} .$$

The number of terms in the $\boldsymbol{\lambda}$ matrix for the optimal basis $\beta \in$ GF(2^4) is equal to $4 \cdot (2 \cdot 4 - 1) = 28$. This implies $4 \cdot (2 \cdot 4 - 2) = 24$ XOR gates in direct computation of the normal basis multiplication. To apply the matrix decomposition method, similar to the case of GF(2^2), we first derive the 4×4 $\boldsymbol{\lambda}_r$ matrices for $r = 0, 1, 2, 3$ from the 4×4 $\boldsymbol{\lambda}$ matrix. Furthermore, using exhaustive search we have obtained the decomposition of the $\boldsymbol{\lambda}_i$ matrices as follows:

$$\boldsymbol{\lambda}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\lambda}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{\lambda}_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{\lambda}_3 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The steps of our algorithm for the normal basis multiplication in GF(2^4) are:

- Step 1: First, we compute the common partial product term using 3 XOR gates. This step requires $2T_X$ gate delays, by arranging the sum computation as a binary tree with 4 leaves, with depth $2T_X$.

$$\begin{aligned} s &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ &= a_0 b_2 + a_1 b_3 + a_2 b_0 + a_3 b_1 . \end{aligned}$$

- Step 2: Then, we use the decomposition of $\boldsymbol{\lambda}_i$ to compute all 4 t_r terms $4 \times 2 = 8$ XOR gates. This step also requires $2T_X$ gate delays.

$$\begin{aligned} t_0 &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ &= a_2 b_1 + a_1 b_2 + a_3 b_3 , \end{aligned}$$

$$
\begin{aligned}
t_1 &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\
&= a_0b_0 + a_3b_2 + a_2b_3 \ ,
\end{aligned}
$$

$$
\begin{aligned}
t_2 &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\
&= a_3b_0 + a_1b_1 + a_0b_3 \ ,
\end{aligned}
$$

$$
\begin{aligned}
t_3 &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\
&= a_1b_0 + a_0b_1 + a_2b_2 \ .
\end{aligned}
$$

- Step 3: Finally, we compute c_r for $r = 0, 1, 2, 3$ using 4 XOR gates: $c_r = s + t_r$. This will require a single T_X gate delay.

The computation of c_0, c_1, c_2, c_3 using the matrix decomposition method requires $3 + 8 + 4 = 15$ XOR gates, instead 24 XOR gates required by the direct method. Since Steps 1 and 2 are independent of one another, the total gate delay is equal to $T_A + 3T_X$.

VIII. Decomposition Method for Type 1 Bases in GF(2^k)

The decomposition method reduces the number of XOR gates due to the common partial product terms a_ib_j among the computation of c_r terms. We define the intersection of two or more $\boldsymbol{\lambda}_r$ matrices as the matrix whose (i, j) element is 1 if all input matrices $\boldsymbol{\lambda}_r$ has a 1 in their (i, j) location, and 0 otherwise. The intersection of all $\boldsymbol{\lambda}_r$ matrices is the matrix used the computation of the partial product term s. We will denote this matrix by $\boldsymbol{\mu}$; for GF(2^2) we obtained it as

$$
\boldsymbol{\mu} = \boldsymbol{\lambda}_0 \bigcap \boldsymbol{\lambda}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \bigcap \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \ ,
$$

Also, we obtained $\boldsymbol{\mu} = \bigcap_{r=0}^{3} \boldsymbol{\lambda}_r$ for the field GF(2^4) as

$$
\begin{aligned}
\boldsymbol{\mu} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \bigcap \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \bigcap \\
& \quad \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \bigcap \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \ .
\end{aligned}
$$

Once the $\boldsymbol{\mu}$ matrix is available, any of $\boldsymbol{\lambda}_r$ matrices for $r = 0, 1, \ldots, k-1$ can be written in terms of $\boldsymbol{\mu}$ and a second matrix. Let us denote the second matrix with $\boldsymbol{\nu}_r$ in the computation of t_r for GF(2^k). Thus, we have $\boldsymbol{\mu} = \bigcap_{r=0}^{k-1} \boldsymbol{\lambda}_r$ and $\boldsymbol{\lambda}_r = \boldsymbol{\mu} + \boldsymbol{\nu}_r$ for $r = 0, 1, \ldots, k-1$.

Of course, it is possible that the $\boldsymbol{\mu}$ matrix can be a zero matrix, implying that there are no common 1s among all $\boldsymbol{\lambda}_r$ matrices. In this case, our method would reduce to the direct method, not offering any savings in the number of XOR gates: $\boldsymbol{\lambda}_r = \boldsymbol{\nu}_r$.

However, we will prove in this section that $\boldsymbol{\mu}$ matrix for Type 1 optimal normal bases in GF(2^k) is a nonzero matrix, in fact it has exactly k 1s in it. The construction of the $\boldsymbol{\mu}$ matrix and the $\boldsymbol{\nu}_r$ matrices for GF(2^k) can be accomplished using the following steps:

1) First, we construct the $\boldsymbol{\lambda}$ matrix. The (i, j) entry of $\boldsymbol{\lambda}$ matrix is equal to $\beta^{2^i+2^j}$ for $0 \leq i, j \leq k-1$, where β is the normal element.
2) We express $\beta^{2^i+2^j}$ in the normal basis, i.e., express it as a linear sum of power of two powers of β. Thus, we obtain the $\boldsymbol{\lambda}$ matrix expressed in the normal basis. This can be accomplished using the polynomial representation of β and the irreducible polynomial of the field to obtain all non-power of 2 powers of β in the normal basis.
3) We obtain the $\boldsymbol{\lambda}_r$ matrices for $r = 0, 1, \ldots, k-1$ by expanding the $\boldsymbol{\lambda}$ matrix as a linear sum of the basis elements β_r.
4) We obtain the intersection matrix $\boldsymbol{\mu} = \bigcap_{r=0}^{k-1} \boldsymbol{\lambda}_r$.
5) Each $\boldsymbol{\nu}_r$ matrix is then obtained using $\boldsymbol{\nu}_r = \boldsymbol{\lambda}_r - \boldsymbol{\mu}$ for $i = 0, 1, \ldots, k-1$.

The construction of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}_r$ matrices depend on the number common 1s in the $\boldsymbol{\lambda}_r$ matrices, which in turn depend on the structure and entries of the $\boldsymbol{\lambda}$ matrix. In order to analyze the complexity of the new multiplication algorithm, we need to look into the properties of the $\boldsymbol{\lambda}$ matrix.

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 $GF(2^k)$. We write $k = 2m$ and use $\mathcal{B}$ to represent the basis set $\mathcal{B} = \{\beta_0, \beta_1, \ldots, \beta_{k-1}\}$. The (i, j) entry of the matrix $\boldsymbol{\lambda}$ for $0 \leq i, j \leq k-1$ is given as

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

Now we refer to Lemmas 1 and 2 in [14] about the structure of the $\boldsymbol{\lambda}$ matrix. The proofs are also given in the same article; we note that the proofs do not assume a particular type of irreducible polynomial generating the field GF(2^k).

Lemma 1: The elements of $\boldsymbol{\lambda}$ with the indices $(i, i+m \bmod k)$ for $i = 0, 1, \ldots, k-1$ are all $\mathbf{1}$s, where $\mathbf{1} = \beta_0 + \beta_1 + \cdots + \beta_{k-1}$ and $m = k/2$.

Lemma 2: The row r for $0 \leq r \leq k-1$ of $\boldsymbol{\lambda}$ is a permutation of $\mathcal{B} - \{\beta_r\}$ with $\mathbf{1}$ appearing in the column index $m+r \bmod k$.

We will denote the set of indices for which the elements of $\boldsymbol{\lambda}$ are all $\mathbf{1}$s by L as

$$
L = \{(i, i+m \bmod k) \mid i = 0, 1, 2, \ldots, k-1\} \ .
$$

Note that L has k elements. As an example, for $k = 10$, L is obtained as

$$\begin{aligned} L &= \{(0,5),(1,6),(2,7),(3,8),(4,9),(5,0),(6,1), \\ &\quad (7,2),(8,3),(9,4)\}\ , \end{aligned}$$

which is seen in the $\boldsymbol{\lambda}$ matrix for GF(2^{10}) below:

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

Using Lemmas 1 and 2, we will prove the following theorem.

Theorem 2: The $\boldsymbol{\lambda}_r$ matrix of the field GF(2^k) with a Type 1 basis can be written as the sum of two matrices $\boldsymbol{\mu}$ and $\boldsymbol{\nu}_r$ such that elements of the $\boldsymbol{\mu}$ matrix with indices in the set $L = \{(i, i+m \bmod k) \mid i = 0,1,2,\ldots,k-1\}$ are 1s. All other entries of $\boldsymbol{\mu}$ are zero. Furthermore, the $\boldsymbol{\nu}_r$ matrix has $k-1$ 1s such that the row r is all zero and every other row has a single 1.

Proof: Since the entries of $\boldsymbol{\lambda}$ with indices in set L are all $\mathbf{1}$ (which is equal to the sum of all k basis elements), the entries of all $\boldsymbol{\lambda}_r$ matrices with indices in the set L will be 1. Since the $\boldsymbol{\mu}$ matrix is equal to the intersection of $\boldsymbol{\lambda}_r$ matrices, such entries of $\boldsymbol{\mu}$ will be equal to 1 as well. Furthermore, consider an entry of $\boldsymbol{\lambda}$ matrix with index $(i,j) \notin L$. This entry would not be equal to $\mathbf{1}$, thus, missing at least one basis element. This implies a zero in the $(i,j) \notin L$ location of one of the $\boldsymbol{\lambda}_r$ matrices, and therefore, a zero in the intersection of all of them, which is the $\boldsymbol{\mu}$ matrix. Therefore, the (i,j) entry of the $\boldsymbol{\mu}$ matrix will be 1 iff $(i,j) \in L$ and 0 otherwise.

On the other hand we obtained $\boldsymbol{\nu}_r$ matrices by subtracting $\boldsymbol{\mu}$ from $\boldsymbol{\lambda}_r$, however, equivalently they can be computed from the $\boldsymbol{\lambda}$ matrix by first removing $\mathbf{1}$s, and then expanding the resulting matrix (which will be denoted by $\boldsymbol{\lambda}'$) in terms of all basis elements. For example, for GF(2^4) we can obtain $\boldsymbol{\nu}_r$ matrices from the $\boldsymbol{\lambda}'$ matrix by expanding it into a sum of all basis elements

$$\boldsymbol{\lambda}' = \boldsymbol{\nu}_0\beta_0 + \boldsymbol{\nu}_1\beta_1 + \boldsymbol{\nu}_2\beta_2 + \boldsymbol{\nu}_3\beta_3$$

such that

$$\begin{bmatrix} \beta_1 & \beta_3 & 0 & \beta_2 \\ \beta_3 & \beta_2 & \beta_0 & 0 \\ 0 & \beta_0 & \beta_3 & \beta_1 \\ \beta_2 & 0 & \beta_1 & \beta_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \beta_0 +$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \beta_1 + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \beta_2 + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \beta_3$$

Due to Lemma 2, the row r of the $\boldsymbol{\lambda}'$ matrix is a permutation of all basis elements except β_r. Since the row r does not contain β_r, the entire rth row of the $\boldsymbol{\nu}_r$ matrix will be zero. Furthermore, β_r will be present in all other rows of the $\boldsymbol{\lambda}'$ matrix except in the row r, there will be a single 1 in all other rows of the $\boldsymbol{\nu}_r$ matrix, giving $k-1$ 1s in the $\boldsymbol{\nu}_r$ matrix. ■

Before we analyze the space requirements of our decomposition method, we should state that the matrix decomposition algorithm given in [8] has essentially the same properties as the one in this paper for Type 1 optimal normal bases, however there are some differences. The method in [8] uses irreducible all-one-polynomials to develop the properties of the $\boldsymbol{\lambda}$ matrix and the decomposition of the $\boldsymbol{\lambda}_r$ matrices. Specifically, our Lemma 1 describes the same property as the one in Equation (8) in [8], and our Theorem 2 describes the same decomposition as the one in Equation 7 in [8]. However, the analysis in [8] is limited to the irreducible all-one-polynomials. Since an all-one-polynomial of degree k is irreducible if $k+1$ is prime and 2 is a primitive element in $\mathcal{Z}_{k+1}$, which are also the existence conditions of the Type 1 optimal normal basis, every optimal normal basis Type 1 can be derived by selecting an irreducible all-one-polynomial [15]. However, optimal normal bases Type 1 can also be derived using other irreducible polynomials, for example, for GF(2^4) in Section VII, we used the irreducible polynomial $p(x) = x^4+x+1$. Our decomposition method does not depend on specific irreducible polynomials, and is derived irrespective of the choice of the irreducible polynomial, and furthermore, it is applicable to the Type 2 optimal normal bases for which there are no irreducible all-one-polynomials.

Theorem 3: The decomposition method for the Type 1 optimal normal basis in GF(2^k) computes all product terms c_r for $r = 0,1,\ldots,k-1$ using k^2 AND gates, k^2-1 XOR gates, and $T_A + [1+\log_2(k)]T_X$ delay.

Proof: The common term s is computed using

$$s = \begin{bmatrix} a_0 & a_1 & \cdots & a_{k-1} \end{bmatrix} \ \boldsymbol{\mu} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

According to Theorem 2, the $\boldsymbol{\mu}$ matrix has exactly k 1s with the indices in L, and all other terms are zero. This implies that we compute s using a linear sum which contains k terms:

$$s = \sum_{i=0}^{k-1} a_i b_{i+m \bmod k} .$$

The computation of s is accomplished using a binary tree of XOR gates with k leaves; the number of XOR gates to compute s is $k-1$, while the delay (the depth of tree) is $\log_2(k)T_X$. The s-tree is illustrated in Figure 1.

On the other hand, a single t_r term is computed using

$$t_r = \begin{bmatrix} a_0 & a_1 & \cdots & a_{k-1} \end{bmatrix} \ \boldsymbol{\nu}_r \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

Also according to Theorem 2, the row r of the $\boldsymbol{\nu}_r$ matrix is zero, while every other row has a single 1 in it. This implies that we compute t_r using a sum which contains $k-1$ terms:

$$\sum_{\substack{i=0\\ i\neq r}}^{k-1} a_{\pi_i} b_i = a_{\pi_0} b_0 + \cdots + a_{\pi_{r-1}} b_{r-1} + a_{\pi_{r+1}} b_{r+1} + \cdots a_{\pi_{k-1}} b_{k-1}$$

where π is a permutation of the indices $\{0, 1, \ldots, r-1, r+1, \ldots, k-1\}$. We create k identical binary trees of XOR gates, each of which has $k-1$ leaves, as shown in Figure 1, named as t_r-trees. The computation of a single t_r term requires $k-2$ XOR gates and $\log_2(k-1)T_X$ delay. The parallel computation of all t_r terms for $r = 0, 1, \ldots, k-1$ requires $k(k-2)$ XOR gates.

Once s and t_r for all $i = 0, 1, \ldots, k-1$ are computed, the computation of a single product term c_r requires one XOR gate and all product terms c_r for $i = 0, 1, \ldots, k-1$ require k XOR gates. However we only need one T_X delay for this computation. Therefore, the total number of gates and the required delay are found as:

1) The computation of s requires $k-1$ XOR gates and $\log_2(k)T_X$ delay.
2) The computation of t_r for all $r = 0, 1, \ldots, k-1$ requires $k(k-2)$ XOR gates and $\log_2(k-1)T_X$ delay.
3) However, we should note that, as illustrated in Figure 1, the computation of s and t_r values are independent of one another. By arranging the s-tree and t_r-trees in parallel, we find the critical path length as $\log_2(k)T_X$.
4) The computation of c_r for all $r = 0, 1, \ldots, k-1$ requires k XOR gates and a single T_X delay.

Thus we find that the total number of the XOR gates required by the matrix decomposition method as $k-1+k(k-2)+k = k^2-1$, while the total delay is $T_A + [1+\log_2(k)]T_X$. ■

IX. Decomposition for Type 2a Bases in GF(2^k)

We now analyze the complexity of the decomposition algorithm for Type 2a bases. We will first derive the $\boldsymbol{\lambda}$ matrix for the field GF(2^5), which has Type 2a basis since $p = 2k+1 = 11$ is prime and 2 is primitive mod 11. Theorem 1 states that the basis element β can be written as $\beta = \gamma + \gamma^{-1}$ such that γ is the 11th root of identity. Our objective is to discover how the $\boldsymbol{\lambda}_r$ matrices can be additively decomposed. The $\boldsymbol{\lambda}$ matrix is given as

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta^{32} \end{bmatrix} .$$

In order to obtain the $\boldsymbol{\lambda}_r$ matrices we need to express all powers of β in the $\boldsymbol{\lambda}$ matrix in terms of the powers of 2 powers of β. First we start with the diagonal entries of the $\boldsymbol{\lambda}$ matrix which already contains powers of 2 powers of β. We have $\beta_r = \beta^{2^r}$ for $r = 0, 1, 2, 3, 4$, and also $\beta^{32} = \beta = \beta_0$. Moreover we should also note that $\beta_0 = \beta = \gamma + \gamma^{-1}$, and

$$\beta_r = \beta^{2^r} = (\gamma + \gamma^{-1})^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$$

for $r = 0, 1, 2, 3, 4$. Next we obtain the normal expansions of the off-diagonal entries which contain the products of two basis elements $\beta^{2^i} \cdot \beta^{2^j}$ for $i, j = 0, 1, 2, 3, 4$ and $i \neq j$. For example, the term $\beta^{2^1} \cdot \beta^{2^3} = \beta^{10}$ is written

$$\begin{aligned} \beta^2 \cdot \beta^8 &= (\gamma^2 + \gamma^{-2})(\gamma^8 + \gamma^{-8}) \\ &= \gamma^{10} + \gamma^{-10} + \gamma^6 + \gamma^{-6} \end{aligned}$$

which contains $10, -10, 6, -6$ powers of γ. We need to express these powers of γ in terms of the powers of 2 powers of γ, and thus obtain a normal expansion for β^{10}. In order to accomplish this, we will use Theorem 1. The general form an off-diagonal product term is written as

$$\beta^{2^i + 2^j} = \gamma^{2^i + 2^j} + \gamma^{-2^i - 2^j} + \gamma^{2^i - 2^j} + \gamma^{-2^i + 2^j}$$

for $0 \leq i, j \leq 4$ and $i \neq j$. By enumerating i and j, we obtain the set of integers of the form $\pm 2^i \pm 2^j$ as

$$\pm\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 17, 18, 20, 24\} .$$

In other words, we need the above powers of γ in order to express all β powers found in the $\boldsymbol{\lambda}$ matrix in the normal basis. Referring to the properties of the Type 2a basis in Theorem 1, we make the following observations:

- $p = 2k+1 = 11$ is prime.
- γ is 11th root of identity, implying that if $u = v \pmod{11}$ then $\gamma^u = \gamma^v$. Therefore, the above set is reduced mod 11, and we only need the powers of γ from the set $\mathcal{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
- 2 is primitive mod 11, that is, the powers of 2 generates the set $\mathcal{Z}_{11}^*$. Since $2^{10} = 1 \pmod{11}$ and $2^5 = -1 \pmod{11}$, which implies that 2^u with $u > 5$ can be written as $2^u = 2^{v+5} = 2^v \cdot 2^5 = -2^v$.

Thus, we can list elements of $\mathcal{Z}_{11}^*$ as

$$\begin{array}{cccccccccc} 2^0 & 2^1 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 & 2^8 & 2^9 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 & 3 & 6 \\ 2^0 & 2^1 & 2^2 & 2^3 & 2^4 & -1 & -2^1 & -2^2 & -2^3 & -2^4 \end{array}$$

Thus, any $u \in \mathcal{Z}_{11}^*$ can be written as $u = \pm 2^v \pmod{11}$ for a $v \in \{0, 1, 2, 3, 4\}$. This implies that we can write $\gamma^u = \gamma^{\pm 2^v}$ for any $u \in \mathcal{Z}_{11}^*$ and $v \in \{0, 1, 2, 3, 4\}$. All γ equalities needed in the $\boldsymbol{\lambda}$ matrix are listed in Table 3 below.

u	$u \pmod{11}$	$u = \pm 2^v \pmod{11}$	γ expansion
3	3	$3 = -2^3$	$\gamma^3 = \gamma^{-2^3}$
5	5	$5 = 2^4$	$\gamma^5 = \gamma^{2^4}$
6	6	$6 = -2^4$	$\gamma^6 = \gamma^{-2^4}$
7	7	$7 = -2^2$	$\gamma^7 = \gamma^{-2^2}$
9	9	$9 = -2^1$	$\gamma^9 = \gamma^{-2}$
10	10	$10 = -2^0$	$\gamma^{10} = \gamma^{-1}$
12	1	$1 = 2^0$	$\gamma^{12} = \gamma$
14	3	$3 = -2^3$	$\gamma^{14} = \gamma^{-2^3}$
15	4	$4 = 2^2$	$\gamma^{15} = \gamma^{2^2}$
17	6	$6 = -2^4$	$\gamma^{17} = \gamma^{-2^4}$
18	7	$7 = -2^2$	$\gamma^7 = \gamma^{-2^2}$
20	9	$9 = -2^1$	$\gamma^{20} = \gamma^{-2}$
24	2	$2 = 2^1$	$\gamma^{24} = \gamma^2$

Table 3: The powers of γ equalities.

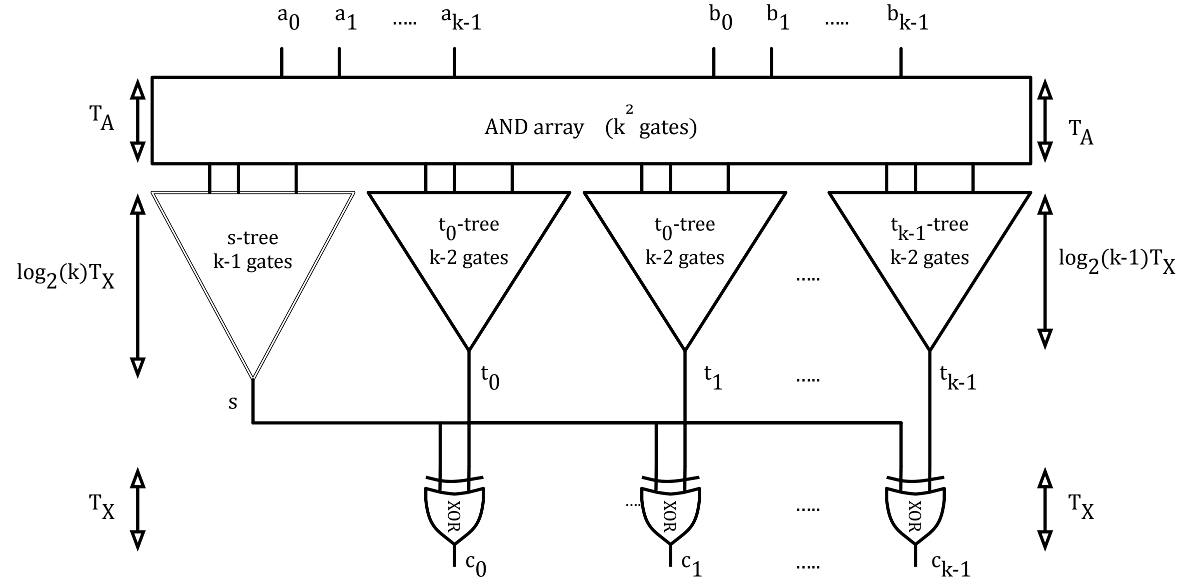

Fig. 1. The matrix decomposition method for Type 1 basis.

Thus, given the equalities $\gamma^{10} = \gamma^{-1}$ and $\gamma^{6} = \gamma^{-2^4}$, we obtain the normal expansion of the product β^{10} as

$$\begin{aligned} \beta^2 \cdot \beta^8 &= \gamma^{10} + \gamma^{-10} + \gamma^{6} + \gamma^{-6} \\ &= \gamma^{-1} + \gamma + \gamma^{-2^4} + \gamma^{2^4} \\ &= \beta_0 + \beta_4 \ . \end{aligned}$$

The other powers of β can be obtained using similar derivations. We omit these derivations, and write the $\boldsymbol{\lambda}$ matrix for GF(2^5) below.

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta^{32} \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} \beta_1 & \beta_0 + \beta_3 & \beta_4 + \beta_3 & \beta_1 + \beta_2 & \beta_4 + \beta_2 \\ \beta_0 + \beta_3 & \beta_2 & \beta_4 + \beta_1 & \beta_0 + \beta_4 & \beta_2 + \beta_3 \\ \beta_4 + \beta_3 & \beta_4 + \beta_1 & \beta_3 & \beta_0 + \beta_2 & \beta_0 + \beta_1 \\ \beta_1 + \beta_2 & \beta_0 + \beta_4 & \beta_0 + \beta_2 & \beta_4 & \beta_1 + \beta_3 \\ \beta_4 + \beta_2 & \beta_2 + \beta_3 & \beta_0 + \beta_1 & \beta_1 + \beta_3 & \beta_0 \end{bmatrix}$$

We observe that the $\boldsymbol{\lambda}$ matrix for GF(2^5) does not have any $\mathbf{1}$ entries, and therefore, the intersection of all $\boldsymbol{\lambda}_r$ matrices is a zero matrix. Unfortunately, a decomposition as in the Type 1 case (which was of the form $\boldsymbol{\lambda}_r = \boldsymbol{\mu} + \boldsymbol{\nu}_r$) is not possible. However, we will show that another decomposition exists.

Theorem 4: The diagonal entries of the $\boldsymbol{\lambda}$ matrix for the field GF(2^k) with a Type 2a basis contain one basis element, while all other entries are the sum of two basis elements.

Proof: The normal element β of the field GF(2^k) with a Type 2a basis is given as $\beta = \gamma + \gamma^{-1}$ where $p = 2k + 1$ is prime, 2 is primitive mod p, and γ is the primitive pth root of identity.

First we observe that all diagonal elements are of the form β^{2^r} for $r = 0, 1, \ldots, k-1$, therefore, each contains a single basis element $\beta^{2^r} = \beta_r$ for $r = 1, 2, \ldots, k-1$ and $\beta^{2^k} = \beta = \beta_0$ for $r = k$. Moreover $\beta_r = \beta^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$ for $r = 0, 1, \ldots, k-1$.

Now consider the (i, j) element of the $\boldsymbol{\lambda}$ for $0 \leq i, j \leq k-1$ and $i \neq j$. This element $\beta^{2^i + 2^j}$ is a product and can be written as

$$\begin{aligned} \beta^{2^i} \cdot \beta^{2^j} &= (\gamma^{2^i} + \gamma^{-2^i})(\gamma^{2^j} + \gamma^{-2^j}) \\ &= \gamma^{2^i + 2^j} + \gamma^{-(2^i + 2^j)} + \gamma^{2^i - 2^j} + \gamma^{-(2^i - 2^j)} \ , \end{aligned}$$

Since γ^p is the identity, the powers of γ above can be reduced mod p, and therefore, we can write

$$\beta^{2^i + 2^j} = \gamma^{u_1} + \gamma^{-u_1} + \gamma^{u_2} + \gamma^{-u_2} \ , \tag{9}$$

such that $u_1 = 2^i + 2^j \pmod{p}$ and $u_2 = 2^i - 2^j \pmod{p}$, where $0 \leq i, j \leq k-1$ and $i \neq j$. Now we will prove that any integer $u \in \mathcal{Z}_p^* = \{1, 2, \ldots, p-1\}$ can be uniquely written as $u = \pm 2^v \pmod{p}$ for some $v \in \mathcal{Z}_k = \{0, 1, \ldots, k-1\}$. Since $p = 2k+1$ prime and 2 is primitive mod p, we have $2^{2k} = 1 \pmod{p}$ and $2^k = -1 \pmod{p}$. Thus, we can generate all elements of $\mathcal{Z}_p^*$ using powers of 2, and furthermore, using the identity $2^k = -1 \pmod{p}$ we obtain

$$\begin{aligned} \mathcal{Z}_p^* &= \{2^0, 2^1, 2^2, \ldots, 2^{k-1}, 2^k, 2^{k+1}, 2^{k+2}, \ldots, 2^{2k-1}\} \\ &= \{2^0, 2^1, 2^2, \ldots, 2^{k-1}, -1, -2^1, -2^2, \ldots, -2^{k-1}\} \end{aligned}$$

This implies that any $u \in \mathcal{Z}_p^*$ can be written as $u = \pm 2^v \pmod{p}$ with $v \in \mathcal{Z}_k$. Thus, we conclude that $\gamma^u = \gamma^{\pm 2^v}$, and write Eqn. (9) as

$$\beta^{2^i + 2^j} = \gamma^{2^{v_1}} + \gamma^{-2^{v_1}} + \gamma^{2^{v_2}} + \gamma^{-2^{v_2}} \ ,$$

Therefore, every off-diagonal element of the $\boldsymbol{\lambda}$ matrix constructed using Type 2a normal basis of the field GF(2^k) contains the sum of 2 basis elements. ■

In order to decompose $\boldsymbol{\lambda}_r$ matrices, we will first separate the diagonal entries and place each of them in different matrices for each r, which we denote as $\boldsymbol{\mu}_r$. As the off-diagonal entries are concerned, we notice that the $\boldsymbol{\lambda}$ matrix is symmetric, implying these pairs of elements appear in two different (and symmetrical) locations. For example, $\beta_0+\beta_1$ is in the locations $(2,4)$ and $(4,2)$ of the $\boldsymbol{\lambda}$ matrix for GF(2^5). Since $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$ matrices respectively hold the coefficients of the basis elements β_0 and β_1, these matrices would have 1s in the same locations $(2,4)$ and $(4,2)$, and thus, their intersection would be a nonzero matrix. Furthermore, β_0 is coupled with every other β_r, the intersection of $\boldsymbol{\lambda}_0$ with $\boldsymbol{\lambda}_r$ for $r = 1,2,\ldots,k-1$ would all be nonzero matrices. These observations suggest a decomposition of the $\boldsymbol{\lambda}_r$ matrices, as expressed in the following theorem.

Theorem 5: The $\boldsymbol{\lambda}_r$ matrix for the field GF(2^k) with a Type 2a basis can be written as the sum of k matrices such that

$$\boldsymbol{\lambda}_r = \boldsymbol{\mu}_r + \sum_{\substack{i=0\\ i\neq r}}^{k-1} \boldsymbol{\nu}_{ri} \ ,$$

where each $\boldsymbol{\mu}_r$ matrix has a single 1 in location $(k-1,k-1)$ for $r=0$ and $(r-1,r-1)$ for $r=1,2,\ldots,k-1$. Furthermore, each $\boldsymbol{\nu}_{ri}$ matrix is symmetric and contains only two 1s.

Proof: The $\boldsymbol{\mu}_r$ matrix contains only the diagonal entries of $\boldsymbol{\lambda}_r$ matrix. As illustrated for GF(2^5) in Eqn. (8) the diagonal entries of the $\boldsymbol{\lambda}$ matrix has the basis elements β_r for $r = 1,2,\ldots,k-1,0$.

$$\begin{bmatrix}
\beta_1 & \beta_0+\beta_3 & \beta_4+\beta_3 & \beta_1+\beta_2 & \beta_4+\beta_2 \\
\beta_0+\beta_3 & \beta_2 & \beta_4+\beta_1 & \beta_0+\beta_4 & \beta_2+\beta_3 \\
\beta_4+\beta_3 & \beta_4+\beta_1 & \beta_3 & \beta_0+\beta_2 & \beta_0+\beta_1 \\
\beta_1+\beta_2 & \beta_0+\beta_4 & \beta_0+\beta_2 & \beta_4 & \beta_1+\beta_3 \\
\beta_4+\beta_2 & \beta_2+\beta_3 & \beta_0+\beta_1 & \beta_1+\beta_3 & \beta_0
\end{bmatrix} .$$

Therefore, the diagonal of the $\boldsymbol{\lambda}_r$ matrix has a single 1, and thus, the entire $\boldsymbol{\mu}_r$ matrix has only 1 in it; all remaining elements are 0. The $\boldsymbol{\mu}_0$ matrix has a 1 in the location $(k-1,k-1)$ while $\boldsymbol{\mu}_r$ has a 1 in the location $(r-1,r-1)$ for $r=1,2,\ldots,k-1$. We obtain the $\boldsymbol{\lambda}_r$ matrices as

$$\begin{array}{ccc}
\boldsymbol{\lambda}_0 & \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 \\
\begin{bmatrix} 0&1&0&0&0\\ 1&0&0&1&0\\ 0&0&0&1&1\\ 0&1&1&0&0\\ 0&0&1&0&1 \end{bmatrix} &
\begin{bmatrix} 1&0&0&1&0\\ 0&0&1&0&0\\ 0&1&0&0&1\\ 1&0&0&0&1\\ 0&0&1&1&0 \end{bmatrix} &
\begin{bmatrix} 0&0&0&1&1\\ 0&1&0&0&1\\ 0&0&0&1&0\\ 1&0&1&0&0\\ 1&1&0&0&0 \end{bmatrix} \\
\boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_4 & \\
\begin{bmatrix} 0&1&1&0&0\\ 1&0&0&0&1\\ 1&0&1&0&0\\ 0&0&0&0&1\\ 0&1&0&1&0 \end{bmatrix} &
\begin{bmatrix} 0&0&1&0&1\\ 0&0&1&1&0\\ 1&1&0&0&0\\ 0&1&0&1&0\\ 1&0&0&0&0 \end{bmatrix} &
\end{array}$$

Furthermore, we obtain the $\boldsymbol{\mu}_r$ matrices as follows:

$$\begin{array}{ccc}
\boldsymbol{\mu}_0 & \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 \\
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1 \end{bmatrix} &
\begin{bmatrix} 1&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} &
\begin{bmatrix} 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} \\
\boldsymbol{\mu}_3 & \boldsymbol{\mu}_4 & \\
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} &
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&1&0\\ 0&0&0&0&0 \end{bmatrix} &
\end{array}$$

We denote the matrix as $\boldsymbol{\nu}_{ri}$ as the intersection of the $\boldsymbol{\lambda}_r$ and $\boldsymbol{\lambda}_i$ matrices as

$$\boldsymbol{\nu}_{ri} = \boldsymbol{\lambda}_r \bigcap \boldsymbol{\lambda}_i \quad \text{for} \quad r \neq i \ .$$

The sum $\beta_u + \beta_v$ of a pair of basis elements β_u and β_v appears in exactly two locations in the $\boldsymbol{\lambda}$ matrix, and thus, the intersection of $\boldsymbol{\lambda}_r$ and $\boldsymbol{\lambda}_i$, i.e., the $\boldsymbol{\nu}_{ri}$ matrix contains only two 1s, and all other elements are zero. For example, $\boldsymbol{\nu}_{0i}$ matrices for GF(2^5) are obtained as

$$\begin{array}{cc}
\boldsymbol{\nu}_{01} & \boldsymbol{\nu}_{02} \\
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&1&0&0 \end{bmatrix} &
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&1&0\\ 0&0&1&0&0\\ 0&0&0&0&0 \end{bmatrix} \\
\boldsymbol{\nu}_{03} & \boldsymbol{\nu}_{04} \\
\begin{bmatrix} 0&1&0&0&0\\ 1&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} &
\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&1&0\\ 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&0&0 \end{bmatrix}
\end{array}$$

Therefore, the $\boldsymbol{\lambda}_r$ matrix of GF(2^k) decomposes into k matrices $\boldsymbol{\mu}_r$ and $\boldsymbol{\nu}_{ri}$ for $i = 0,1,\ldots,r-1,r+1,\ldots,k-1$ such that the $\boldsymbol{\mu}_r$ matrix contains a single 1, and all $\boldsymbol{\nu}_{ri}$ matrices contain 2 1s. ■

The space complexity of the multiplication using decomposition method is analyzed in the following theorem.

Theorem 6: The decomposition method for the Type 2a optimal normal basis in GF(2^k) computes all product terms c_r for $r = 0,1,\ldots,k-1$ using k^2 AND gates, $1.5k(k-1)$ XOR gates, and a total delay of $T_A + [1+\log_2(k)]T_X$.

Proof: According to Theorem 5, the $\boldsymbol{\lambda}_r$ matrix can be written as the sum of k matrices as

$$\boldsymbol{\lambda}_r = \boldsymbol{\mu}_r + \sum_{\substack{i=0\\ i\neq r}}^{k-1} \boldsymbol{\nu}_{ri} \ .$$

The computation of the product term c_r is accomplished using

$$c_r = \begin{bmatrix} a_0 & a_1 & \cdots & a_{k-1} \end{bmatrix} \left(\boldsymbol{\mu}_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} \boldsymbol{\nu}_{ri} \right) \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix} .$$

For brevity, we will denote the input vectors by $\boldsymbol{a}^T$ and $\boldsymbol{b}$, and break the above product computation into the sum of k matrix-vector products as

$$c_r = \boldsymbol{a}^T \ \boldsymbol{\mu}_r \ \boldsymbol{b} + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} \left(\boldsymbol{a}^T \ \boldsymbol{\nu}_{ri} \ \boldsymbol{b} \right) = s_r + \sum_{\substack{i=0 \\ i \neq r}}^{k-1} t_{ri} \ . \qquad (10)$$

The individual components of the above sum, s_r and t_{ri}, are defined as

$$\begin{aligned} s_r &= \boldsymbol{a}^T \ \boldsymbol{\mu}_r \ \boldsymbol{b} \ , \\ t_{ri} &= \boldsymbol{a}^T \ \boldsymbol{\nu}_{ri} \ \boldsymbol{b} \ , \end{aligned}$$

for $0 \leq i \leq k-1$ and $i \neq r$. Once the terms s_r and t_{ri} are computed we can obtain the product c_r using Eqn. (10). Steps of the computation of all c_r terms are described below and illustrated in Figure 2.

1) The computation of s_r does not require any XOR gates. The matrix $\boldsymbol{\mu}_r$ has a single 1 in it; the location is $(k-1, k-1)$ for $r = 0$ and $(r-1, r-1)$ for all other $r = 1, 2, \ldots, k-1$. Therefore, $s_0 = a_{k-1} b_{k-1}$ and $s_r = a_{r-1} b_{r-1}$ for $r = 1, 2, \ldots, k-1$. There is no delay involved, either, the selection logic works by routing the logic signals.
2) The $\boldsymbol{\nu}_{ri}$ has only two 1s and it is also symmetric. If the (u, v) element of the $\boldsymbol{\nu}_{ri}$ matrix is 1, then so is (v, u) element, while all the other elements are zero. This gives the value of t_{ri} as $a_u b_v + a_v b_u$. Therefore, the computation of a single t_{ri} requires 1 XOR gate and T_X delay. Furthermore, we have $\boldsymbol{\nu}_{ri} = \boldsymbol{\nu}_{ir}$, and thus, $t_{ri} = t_{ir}$. This implies that we only need to compute half of the t_{ir} terms due to the symmetry. For example, for $k = 5$ the following terms need to be computed: t_{0i} for $i = 1, 2, 3, 4$; t_{1i} for $i = 2, 3, 4$; t_{2i} for $i = 3, 4$; finally t_{34}. For GF(2^k) the number of terms that need to be computed is
$$(k-1) + (k-2) + \cdots + 1 = k(k-1)/2 \ ,$$
which gives the total number of XOR gates for computing all t_{ri} terms as $0.5k(k-1)$, while the delay is still equal to one T_X.
3) Having obtained all s_r and t_{ri} values, we compute c_r using the summation Eqn. (10) which has k terms. We arrange this summation using a binary tree of XOR gates, which has k leaves. There is a separate binary for each value of $r = 0, 1, \ldots, k-1$; there are k inputs for each tree such that s_r, t_{ri} except t_{rr} term. The computation of a single c_r term requires $k-1$ XOR gates and $\log_2(k) T_X$ units of delay, while all c_r terms would require a total of $k(k-1)$ XOR gates.

Therefore the total number of XOR gates is found as $1.5k(k-1)$, and the total delay is $T_A + [1 + \log_2(k)] T_X$. ■

X. Decomposition for Type 2b Bases in GF(2^k)

The smallest field with the Type 2b basis is GF(2^3). For $k = 3$, we have $p = 2k + 1 = 7$ prime, $p = 3 \pmod 4$, and 2 generates the quadratic residues in $\mathcal{Z}_7^*$. Furthermore, a basis element $\beta_i = \beta^{2^i}$ is equal to $\gamma^{2^i} + \gamma^{-2^i}$ for $i = 0, 1, 2$, where γ is the 7th root of identity according to Theorem 1. We can write $\gamma^3 = \gamma^{-4}$, $\gamma^5 = \gamma^{-2}$, and $\gamma^6 = \gamma^{-1}$, and obtain the products of the basis elements as

$$\begin{aligned} \beta\beta^2 = \beta^3 &= \gamma^3 + \gamma^{-3} + \gamma + \gamma^{-1} \\ &= \gamma^{-4} + \gamma^4 + \gamma + \gamma^{-1} \\ &= \beta_0 + \beta_2 \\ \beta\beta^4 = \beta^5 &= \gamma^5 + \gamma^{-5} + \gamma^3 + \gamma^{-3} \\ &= \gamma^{-2} + \gamma^2 + \gamma^4 + \gamma^{-4} \\ &= \beta_1 + \beta_2 \\ \beta^2\beta^4 = \beta^6 &= \gamma^6 + \gamma^{-6} + \gamma^2 + \gamma^{-2} \\ &= \gamma^{-1} + \gamma + \gamma^2 + \gamma^{-2} \\ &= \beta_0 + \beta_1 \end{aligned}$$

Therefore, the $\boldsymbol{\lambda}$ matrix is obtained as

$$\boldsymbol{\lambda} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 \\ \beta^3 & \beta^4 & \beta^6 \\ \beta^5 & \beta^6 & \beta^8 \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_0 + \beta_2 & \beta_1 + \beta_2 \\ \beta_0 + \beta_2 & \beta_2 & \beta_0 + \beta_1 \\ \beta_1 + \beta_2 & \beta_0 + \beta_1 & \beta_0 \end{bmatrix} .$$

Similar to the Type 2a case, we see that the $\boldsymbol{\lambda}$ matrix for GF(2^3) contains a single basis on the diagonal, while all off-diagonal elements are equal to and the sum of two bases. We prove that this property holds true for any k.

Theorem 7: The diagonal entries of the $\boldsymbol{\lambda}$ matrix for the field GF(2^k) with a Type 2b basis contain one basis element, while all other entries are the sum of two basis elements.

Proof: All diagonal elements of the $\boldsymbol{\lambda}$ matrix are of the form β^{2^r}, and therefore, each contains a single basis element $\beta^{2^r} = \beta_r$ for $0 = 1, 2, \ldots, k-1$. Furthermore, we have $\beta = \gamma + \gamma^{-1}$ where γ is the $p = 2k+1$ primitive root of identity. A diagonal element is of the form $\beta^{2^r} = \gamma^{2^r} + \gamma^{-2^r}$ for $r = 0, 1, \ldots, k-1$.

Similar to the Type 2a case, an off-diagonal element is given as $\beta^{2^i + 2^j}$ for $i = 1, 2, \ldots, j-1, j+1, \ldots, k-1$, which is equal to

$$\beta^{2^i} \cdot \beta^{2^j} = \gamma^{2^i + 2^j} + \gamma^{-(2^i + 2^j)} + \gamma^{2^i - 2^j} + \gamma^{-(2^i - 2^j)} \ .$$

Since γ^p is the identity, the powers of γ above are reduced mod p, and therefore, we can write

$$\beta^{2^i + 2^j} = \gamma^{u_1} + \gamma^{-u_1} + \gamma^{u_2} + \gamma^{-u_2} \ , \qquad (11)$$

such that $u_1 = 2^i + 2^j \pmod p$ and $u_2 = 2^i - 2^j \pmod p$, where $0 \leq i, j \leq k-1$ and $i \neq j$. Next we will prove that any integer $u \in \mathcal{Z}_p^* = \{1, 2, \ldots, p-1\}$ can be uniquely written as $u = \pm 2^v \pmod p$ for some $v \in \mathcal{Z}_k = \{0, 1, \ldots, k-1\}$.

Theorem 1 states that for Type 2b basis, $p = 3 \pmod 4$ and 2 generates quadratic residues mod p. We use Q_p to denote the set of quadratic residues, which has $(p-1)/2$ elements. An element $u \in \mathcal{Z}_p^*$ is in Q_p if there is a solution x for the equation $x^2 = u \pmod p$, otherwise u is a quadratic nonresidue. The set of quadratic nonresidues, denoted by Q'_p,

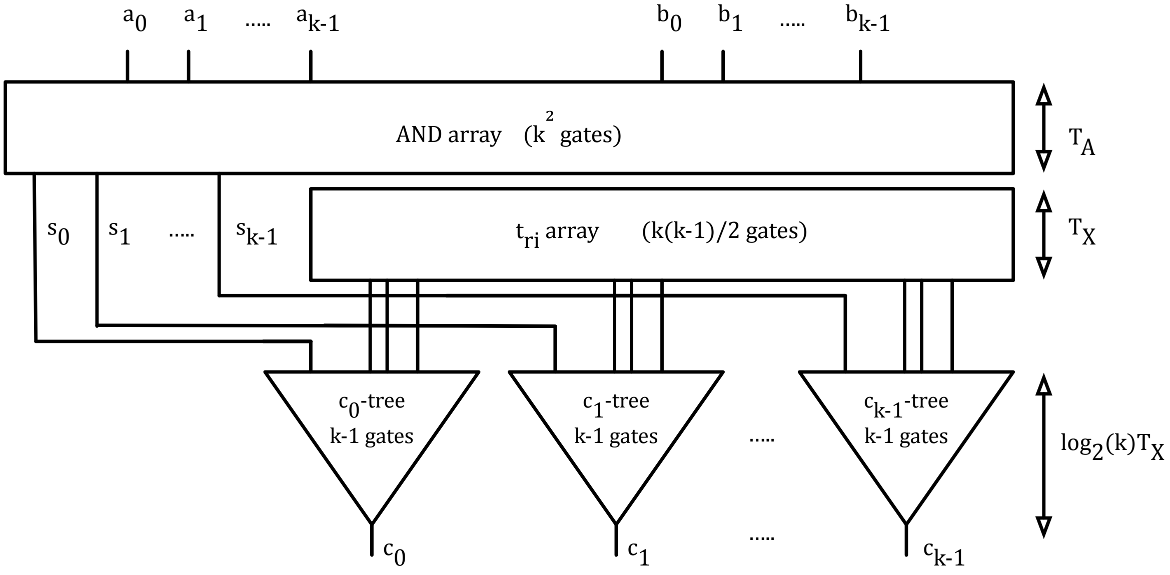

Fig. 2. The matrix decomposition method for Type 2a and 2b bases.

consists of the remaining $(p-1)/2$ elements of $\mathcal{Z}_p^*$. For example, for $k=11$, $p=23$, these two sets are given as

$$\begin{aligned} Q_{23} &= \{1,2,3,4,6,8,9,12,13,16,18\} \ , \\ Q'_{23} &= \{5,7,10,11,14,15,17,19,20,21,22\} \ . \end{aligned}$$

The Euler criterion determines if $u \in Q_p$ or $u \in Q'_p$:

$$u^{(p-1)/2} = \begin{cases} 1 & \text{if} \quad u \in Q_p \ , \\ -1 & \text{if} \quad u \in Q'_p \ . \end{cases}$$

An important observation is that $-1 \in Q'_p$ if $p = 3 \pmod 4$, since

$$(-1)^{(p-1)/2} = \begin{cases} 1 & \text{if} \quad p = 1 \pmod 4 \ , \\ -1 & \text{if} \quad p = 3 \pmod 4 \ . \end{cases}$$

Another relevant property of quadratic residues is that if $u \in Q_p$ and $v \in Q'_p$ then the product $uv \in Q'_p$. Particularly, in our case, we can write $-u \in Q'_p$ if $u \in Q_p$, since $-1 \in Q'_p$. Since Q_p is generated by powers of 2, it follows that

$$Q_p = \{2^v \pmod p \mid v \in \mathcal{Z}_k\} \ .$$

We can generate Q'_p by multiplying every element of Q_p by -1, in other words,

$$Q'_p = \{-2^v \pmod p \mid v \in \mathcal{Z}_k\} \ .$$

Since $\mathcal{Z}_p^* = Q_p \bigcup Q'_p$, we can write

$$\mathcal{Z}_p^* = \{\pm 2^v \pmod p \mid v \in \mathcal{Z}_k\} \ .$$

This implies that any $u \in \mathcal{Z}_p^*$ can be written as $u = \pm 2^v \pmod p$ with $v \in \mathcal{Z}_k$. Thus, we conclude that $\gamma^u = \gamma^{\pm 2^v}$, and write Eqn. (11) as

$$\beta^{2^i+2^j} = \gamma^{2^{v_1}} + \gamma^{-2^{v_1}} + \gamma^{2^{v_2}} + \gamma^{-2^{v_2}} \ ,$$

Therefore, every off-diagonal element of the $\boldsymbol{\lambda}$ matrix constructed using Type 2a normal basis of the field GF(2^k) contains the sum of 2 basis elements. ■

Therefore, the same complexity analysis for Type 2a applies for Type 2b as well. The complexity of the multiplication using decomposition method for the Type 2b bases is the same as that of Type 2a bases.

Theorem 8: The matrix decomposition method for the Type 2b optimal normal basis in GF(2^k) computes all product terms c_r for $r = 0, 1, \ldots, k-1$ using k^2 AND gates, $1.5k(k-1)$ XOR gates, and a total delay of $T_A + [1 + \log_2(k)]T_X$.

XI. Conclusions

We introduced a matrix decomposition method and described the underlying algorithms for normal basis multiplication in the field GF(2^k) with Type 1 and Type 2 bases.

We developed the matrix decomposition method explicitly on small fields; for $k = 2$ and $k = 4$ for Type 1 basis, and $k = 5$ for Type 2a basis, $k = 3$ for Type 2b basis. However, we derived the space complexity results for general values of k for all three types of bases, as given in Theorems 3, 6, and 8, respectively.

The decomposition algorithm computes all product terms for the Type 1 basis using $k^2 - 1$ XOR gates, irrespective of the irreducible polynomial generating the field. The previous Massey-Omura multiplication algorithms [16], [9], [11] accomplished the same bound using all-one-polynomials. Furthermore, our matrix decomposition algorithm computes all product terms for the Type 2a and 2b bases using $1.5k(k-1)$ XOR gates, which matches previous bounds [10], [11].

The Type 1 normal basis multiplication algorithm given in [11] is also based on a matrix decomposition in which the $\boldsymbol{\lambda}$ matrix is decomposed into upper and lower triangular matrices and a diagonal matrix. The XOR complexity of this algorithm is given for all-one-polynomials as k^2-1, however, an analysis for a *general irreducible polynomial* is not given. Instead, it was shown that the algorithm for GF(2^5) requires

8 XOR gates. However, one has to note that this is a straightforward decomposition which follows directly the definition of symmetric matrices, and separates the multiplication terms into three groups. Their algorithm then rearranges the terms of this sum. In our approach however, we find an optimal decomposition with respect to the chosen normal basis and the corresponding multiplication matrix. After creating the optimal decomposition we are able to create the circuit without any intermediate steps. For the optimal normal basis, our results match the results in [11], but we do not restrict our algorithm to all-one polynomials, and we extend to arbitrary normal bases without additional effort.

It is also interesting to note that the Mastrovito algorithms, which work only for the polynomial basis, achieve the $k^2 - 1$ space complexity with irreducible trinomials [17], [18], [19], [20]. Furthermore, the space complexity falls to $k^2 - \Delta$ for equally-spaced polynomials [21], [22], where Δ is the distance factor; in other words, the irreducible polynomial is of the form

$$p(x) = x^{m\Delta} + x^{(m-1)\Delta} + \cdots + x^{\Delta} + 1 \ .$$

In a highly special case of equally-spaced-trinomial $x^k + x^{k/2} + 1$, the space complexity becomes $k^2 - k/2$ [21]. This implies that the bound $k^2 - 1$ is not very tight and there may be more special cases in which the space complexity falls further from that. However, it is highly likely that the result of this paper provides the lower bound for optimal normal bases, irrespective of the irreducible polynomial. This remains to be proven.

Another promising direction for future work is to investigate if we can reduce the space complexity for Gaussian normal basis multiplication using our matrix decomposition approach. Optimal normal bases do not exist for all k, however, non-optimal but still low complexity normal bases do exist, and are called Gaussian normal bases [23], [24]. The Type T of a Gaussian normal basis in GF(2^k) is a positive integer describing the structure and measuring the complexity of the multiplication in the basis [3].

For a given k and T, there exists at most one Gaussian normal basis of Type T. A Type T Gaussian normal basis for a given field GF(2^k) exists if and only if $p = Tk + 1$ is prime and $\gcd(Tk/m, k) = 1$ where m is the multiplicative order of 2 in $\mathcal{Z}_p^*$. When $T = 1$, the Gaussian normal basis Type 1 is the same as the optimal normal basis Type 1, since Part 1 conditions of Theorem 1 are satisfied: $p = k + 1$ is prime, the multiplicative order of 2 in $\mathcal{Z}_p^*$ is $k = p - 1$, that is 2 is primitive, and thus $\gcd(Tk/m, k) = \gcd(k/k, k) = \gcd(1, k) = 1$. Similarly, when $T = 2$, the Gaussian normal basis Type 2 is the same as the optimal normal basis Type 2a: $p = 2k + 1$ is prime, the multiplicative order of 2 in $\mathcal{Z}_p^*$ is $p - 1 = 2k$, that is 2 is primitive, and thus $\gcd(Tk/m, k) = \gcd(2k/(2k), k) = \gcd(1, k) = 1$.

Our analysis of Type 2a basis in Section IX showed that for $T = 2$, all rows of the $\boldsymbol{\lambda}_r$ (except row 0) has two nonzero entries. This fact was also stated in Remark 1 of [25]. Both the Remark 1 in [25] and our analysis in Section IX address Type 2a only. However, we were also able to show in this paper (in Section X) that Type 2b bases have the same complexity as Type 2a bases. We believe it is worthwhile to investigate the complexity of the Gaussian normal basis with even $T = 2N$ with properties $p = Tk + 1 = 2N + 1$ prime, the multiplicative order of 2 in $\mathcal{Z}_p^*$ is m, and $\gcd(Tk/m, k) = \gcd(2Nk/m, k) = 1$. It was shown in [26] that the Gaussian normal basis multipliers for GF(2^k) for odd k can be more efficient in terms of space complexity. The multiplication algorithms described in [26] require 16% or 27% fewer XOR gates than the standard parallel-input parallel-output multiplier for $k = 163$ and $k = 409$, respectively. These fields have important applications in the Elliptic Curve Digital Signature Algorithm (ECDSA) of the NIST standard FIPS 186-3 [27]. Moreover the algorithms in [26] yield new elliptic curve point addition and doubling formulations [28] which utilize a novel digit-level hybrid-double Gaussian normal basis multiplier [29]. This shows the importance of the Gaussian normal basis multipliers; its applications in elliptic curve cryptography make them highly useful and new research in this direction highly worthwhile.

References

[1] R. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

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

[3] D. Johnson, A. Menezes, and S. A. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, pp. 36–63, 2001.

[4] D. J. Bernstein, T. Lange, and R. R. Farashahi, "Binary Edwards curves," in *Cryptographic Hardware and Embedded Systems - CHES 2008*, E. Oswald and P. Rohatgi, Eds. Springer, LNCS Nr. 5154, 2008, pp. 244–265.

[5] K. H. Kim, C. O. Lee, and C. Negre, "Binary Edwards curves revisited," in *Progress in Cryptology – INDOCRYPT 2014*, W. Meier and D. Mukhopadhyay, Eds. Springer, LNCS Nr. 8885, 1999, pp. 393–408.

[6] J. Omura and J. Massey, "Computational method and apparatus for finite field arithmetic," May 1986, US Patent Number 4,587,627.

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

[8] M. A. Hasan, M. Z. Wang, and V. K. Bhargava, "A modified Massey-Omura parallel multiplier for a class of finite fields," *IEEE Transactions on Computers*, vol. 42, no. 10, pp. 1278–1280, Nov. 1993.

[9] Ç. K. Koç and B. Sunar, "Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields," *IEEE Transactions on Computers*, vol. 47, no. 3, pp. 353–356, Mar. 1998.

[10] B. Sunar and Ç. K. Koç, "An efficient optimal normal basis type II multiplier," *IEEE Transactions on Computers*, vol. 50, no. 1, pp. 83–87, Jan. 2001.

[11] A. Reyhani-Masoleh and M. A. Hasan, "A new construction of Massey-Omura parallel multiplier over GF(2^m)," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 511–520, May 2002.

[12] S. Gao and H. W. Lenstra, Jr., "Optimal normal bases," *Designs, Codes and Cryptography*, vol. 2, no. 4, pp. 315–323, Dec. 1992.

[13] S. Gao, "Normal bases over finite fields," Ph.D. dissertation, University of Waterloo, 1993.

[14] Ö. Eğecioğlu and Ç. K. Koç, "Reducing the complexity of normal basis multiplication," in *Arithmetic of Finite Fields*, Ç. K. Koç, S. Mesnager, and E. Savaş, Eds. LNCS Nr. 9061, Springer, 2014, pp. 61–80.

[15] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, 2005.

[16] 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*, vol. 41, no. 8, pp. 962–971, Aug. 1992.

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

[18] ——, "VLSI architectures for computation in Galois fields," Ph.D. dissertation, Linköping University, Department of Electrical Engineering, Linköping, Sweden, 1991.

[19] C. Paar, "Efficient VLSI architectures for bit parallel computation in Galois fields," Ph.D. dissertation, Universität GH Essen, VDI Verlag, 1994.

[20] ——, "A new architecture for a paralel finite field multiplier with low complexity based on composite fields," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 856–861, Jul. 1996.

[21] B. Sunar and Ç. K. Koç, "Mastrovito multiplier for all trinomials," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 522–527, May 1999.

[22] A. Halbutoğulları and Ç. K. Koç, "Mastrovito multiplier for general irreducible polynomials," *IEEE Transactions on Computers*, vol. 49, no. 5, pp. 503–518, May 2000.

[23] D. W. Ash, I. F. Blake, and S. A. Vanstone, "Low complexity normal bases," *Discrete Applied Mathematics*, vol. 25, pp. 191–210, 1989.

[24] R. Mullin, I. Onyszchuk, S. A. Vanstone, and R. Wilson, "Optimal normal bases in GF(p^n)," *Discrete Applied Mathematics*, vol. 22, pp. 149–161, 1988/1989.

[25] A. Reyhani-Masoleh, "Efficient algorithms and architectures for field multiplication using Gaussian normal bases," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 34–47, Jan. 2006.

[26] R. Azarderakhsh, D. Jao, and H. Lee, "Common subexpression algorithms for space-complexity reduction of Gaussian normal basis multiplication," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2357–2369, May 2015.

[27] National Institute for Standards and Technology, "Digital Signature Standard (DSS)," FIPS 186-3, June 2009.

[28] R. Azarderakhsh and A. Reyhani-Masoleh, "Parallel and high-speed computations of elliptic curve cryptography using hybrid-double multipliers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1668–1677, Jun. 2015.

[29] ——, "Low complexity multiplier architectures for single and hybrid-double multiplications in Gaussian normal bases," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 744–757, Apr. 2013.