

# Efficient Algorithm for Finding Roots of Error-Locator Polynomials

SERGEI VALENTINOVICH FEDORENKO 

Department of Informatics, HSE University, HSE Campus in Saint Petersburg, 194100 Saint Petersburg, Russia

e-mail: sfedorenko@hse.ru

**ABSTRACT** A novel method for finding roots of polynomials over finite fields has been proposed. This method is based on the cyclotomic discrete Fourier transform algorithm. The improvement is achieved by using the normalized cyclic convolutions, which have a small complexity and allow matrix decomposition, as well as methods of adapting the truncated normalized cyclic convolutions calculation. For small values of degree of the error-locator polynomial the novel method has not only the smallest multiplicative complexity, but the full computational complexity of this method is also less than with other methods. Thus, the multiplicative complexity of the novel method in comparison to the method of affine decomposition (the Fedorenko–Trifonov method) is up to ten times less, although the additive complexity is approximately 10–15% more. The novel method has matrix representation convenient for implementation.

**INDEX TERMS** Convolution, decoding, discrete Fourier transforms, error correction codes, fast Fourier transforms, Galois fields, Reed–Solomon codes.

## I. INTRODUCTION


Finding roots of polynomials over finite fields is a very actual problem. The step of finding the roots of the error-locator polynomial is the second most time-consuming step of the decoding process of the Reed–Solomon codes. That is why the most important application of calculating the roots of polynomials is decoding Reed–Solomon codes. Since non-systematic encoding of Reed–Solomon codes is performed by using the truncated discrete Fourier transform (DFT) calculation (see Definition 5) then nonsystematic encoding is another application of the novel method.

Let us formulate the problem of finding the roots of a polynomial: Find all the roots of an arbitrary polynomial of degree  $t$  over finite field  $GF(2^m)$ , the roots are searched in the field of the computation  $GF(2^m)$ , and the multiplicity of the roots is not taken into account.

Several methods of finding the roots of polynomials are known:

1) Algebraic methods.

For small values of degree  $t$  ( $t \leq 4$ ) these methods are the simplest (see, for example, [8], [14], [18], [26], [38], [39]). If the polynomial is an affine polynomial (see Definition 2) (this polynomial is not an arbitrary polynomial), then we can apply the method [5, Ch. 11].

The associate editor coordinating the review of this manuscript and approving it for publication was Zilong Liu .

Also for any polynomial we can construct the least affine multiple of this polynomial [5, Ch. 11], and then apply the method of Berlekamp too. Unfortunately, with increasing degree  $t$ , the complexity of the latter methods increases significantly.

- 2) Representation of polynomials in the form of the sum of affine polynomials was first introduced in [35], method of affine decomposition (the Fedorenko–Trifonov method) was introduced in [17], and developed in [15], [18], [29].
- 3) The calculation of the polynomial roots corresponds to the polynomial evaluation for all nonzero elements of finite field. Obviously, this coincides with the calculation of the DFT over finite field (see Definition 3). Many methods for the DFT calculation over finite field are known. The best methods for the asymptotic complexity are [24], [27], [28]. To minimize the number of multiplications, the cyclotomic DFT algorithm [19], [32] should be considered, as well as improvements of this algorithm to reduce the number of multiplications [3], [20], [22], and the number of additions [3], [4], [10], [11], [34], [40], [41]. The application of several convolutions for the DFT calculation were published in [19], [20], [22], [32], [36], [40]. The methods of constructing cyclic convolutions are considered in [4], [6], [7], [21], [36], [40], [41]. These methods can be used to construct the cyclotomic

TABLE 1. The best methods of finding the roots of polynomials over finite field  $GF(2^m)$ .

$t$	Method	References
$t \leq 4$	algebraic methods	[8], [14], [26], [38], [39], [18], [5]
$4 < t \ll 2^m$	affine decomposition methods	[35], [17], [15], [18], [29]
$t \approx 2^m$	DFT calculation	[32], [19], [3], [20], [22], [4], [11], [40], [10], [34], [41]

DFT algorithm. Reducing the complexity of decoding the Reed–Solomon codes by speeding up the calculation of the DFT, Chien search or syndrome calculation is described in [9], [12], [30], [37], [42].

- 4) Several algorithms for the factorization of polynomials over finite fields are presented in the monograph [25, Ch. 14] and in [2], [31].
- 5) The above methods can be combined, which leads to a decrease of the computational complexity. Hybrid methods are described, for example, in [14], [18]. To simplify the presentation, a combination of methods for finding the roots of polynomials in this paper will not be considered.

The best methods of finding the roots of polynomials over finite field  $GF(2^m)$ , depending on the value of degree  $t$  are shown in Table 1.

The novel method is based on the cyclotomic DFT algorithm. The following points have been added to the original version of the cyclotomic DFT algorithm [32]:

- 1) the novel normalized cyclic convolutions [21] have the following properties:
  - a) they lead to a decrease in computational complexity in relation to the classical cyclic convolutions;
  - b) there is a decomposition into a product of square matrices of the same dimension;
  - c) the matrix decomposition allows recursion;
  - d) they allow easy modification by matrix transposition;
- 2) the construction of truncation of the normalized cyclic convolutions through
  - a) adaptation of input components of the convolutions;
  - b) cyclic shifts for input components of the convolutions;
  - c) the input components of convolutions cease being consecutive;
- 3) ordering the cyclotomic cosets and generators for the cyclotomic cosets;
- 4) constructing the normal bases for each subfield according to Lemma 1.

The cyclotomic DFT algorithm [32] and its improvements [3], [4], [10], [11], [19], [20], [22], [32], [34], [40], [41] have the smallest multiplicative complexity among all known algorithms calculating DFT for short lengths  $n \leq 4095$ . The additive and the full complexities are also comparable to the complexities of other algorithms. The comparison of the complexity of different full and partial DFT is presented in [40, Table IV], [10, Table I, II], [29, Table I].

It is well known that if DFT calculation has the lowest complexity [19], [22], [32], then by using partial

DFT calculation we have the lowest complexity, too. According to this statement, the author believes that at present there are no better methods for finding the roots of polynomials over finite field  $GF(2^m)$  for  $4 < t \ll 2^m$  than the one introduced in this article.

The presented method is a nontrivial development of the ideas of the methods [17], [18]. The main differences between the proposed method and the affine decomposition method [17], [18] are

- 1) the affine decomposition of the error-locator polynomial is replaced by the decomposing of the original polynomial into a sum of linearized polynomials;
- 2) in the polynomial decomposition there are no monomials of type  $x^3$  and  $x^5$ .

The applicability of the method: the method is applicable for any values of degree  $t$  over finite field  $GF(2^m)$ , but for  $4 < t \ll 2^m$  this method will have less computational complexity than any other known methods.

The main contributions of the novel method for finding roots of polynomials over finite fields consist of the following points:

- 1) the novel method is the algorithm with the smallest multiplicative complexity for finite fields;
- 2) the full computational complexity of this method is also less than with other methods;
- 3) the concept of *truncated* normalized cyclic convolutions is introduced;
- 4) adapting the *truncated* cyclotomic DFT and the *truncated* normalized cyclic convolutions.

*Notations:* Cursive bold lower case letters denote vectors, but cursive bold upper case letter  $\mathbf{F}$  is used for denoting the resulting vector of the DFT. Bold upper case letters denote matrices. The expression  $a \mid b$  denotes the fact that integer  $a$  divides integer  $b$ .

The remainder of this paper is organized as follows.

In Section II, the description of the Fedorenko–Trifonov method for finding roots of polynomials over finite fields [17] and the concept of the affine decomposition are given. In Section III, the cyclotomic DFT algorithm [32] is introduced. These two Sections are also necessary, since they introduce the concepts and definitions necessary for the presentation of the novel method. In Section IV, the concept of the truncated normalized cyclic convolution is described. In Section V, the novel method of finding roots of polynomials via the concept of the truncated cyclotomic DFT is proposed. In Section VI, the complexity of the truncated cyclotomic DFT computation is calculated, and the Tables for the complexity of the truncated cyclotomic DFT computation over often applicable finite fields are shown. In Conclusion,

the comparison of the computational complexities of the novel and well-known methods is presented.

## II. FEDORENKO–TRIFONOV METHOD FOR FINDING ROOTS OF POLYNOMIALS OVER FINITE FIELDS

This Section is based on the method [17]. Let us introduce the following definition for polynomials.

**Definition 1:** A linearized polynomial over  $GF(2^m)$  is a polynomial of the form

$$L(x) = \sum_i b_i x^{2^i}, \quad b_i \in GF(2^m).$$

A main property of the linearized polynomials is  $L(\lambda + \mu) = L(\lambda) + L(\mu)$ , where  $\lambda, \mu \in GF(2^m)$ .

**Definition 2:** An affine polynomial over  $GF(2^m)$  is a polynomial of the form

$$A(x) = L(x) + \beta,$$

where  $L(x)$  is a linearized polynomial over  $GF(2^m)$  and constant  $\beta \in GF(2^m)$ .

### A. AFFINE DECOMPOSITION

Each polynomial can be decomposed into a sum of multiples of affine polynomials according to Theorem 1.

**Theorem 1 ([17]):** Each polynomial  $f(x) = \sum_{i=0}^t f_i x^i$ ,  $f_i \in GF(2^m)$ , of degree  $t$  can be represented as

$$f(x) = f_3 x^3 + \sum_{k=0}^{\lceil (t-4)/5 \rceil} x^{5k} (f_{5k} + L_k(x)),$$

where  $L_k(x) = \sum_{s=0}^3 f_{5k+2^s} x^{2^s} = f_{5k+1}x + f_{5k+2}x^2 + f_{5k+4}x^4 + f_{5k+8}x^8$ ,  $\lceil p \rceil$  is the least integer greater than or equal to  $p$ ,  $f_i = 0$  for  $i > t$ .

Another representation for the affine decomposition of the polynomial is

$$f(x) = f_3 x^3 + \left( A_0(x) + x^5 \left[ A_1(x) + x^5 \left\{ A_2(x) + \dots \right\} \right] \right),$$

where  $A_k(x) = f_{5k} + L_k(x)$ .

Let  $\alpha$  be a primitive element of the field  $GF(2^m)$ . The binary representations for all elements  $\{0, \alpha_0, \alpha_1, \dots, \alpha_{2^m-2}\}$  of  $GF(2^m)$  are ordered as a Gray code. Let  $\delta(\alpha_j, \alpha_{j-1}) \in \{0, 1, \dots, m-1\}$ ,  $j = 1, 2, \dots, 2^m-2$ , be a position in which binary representations of elements  $\alpha_j$  and  $\alpha_{j-1}$  are different. For  $j = 0$  we assume  $\delta(\alpha_0, 0) = 0$ .

The algorithm for finding roots consists of four steps:

- 1) Compute  $b_{k,p} = L_k(\alpha^p)$ ,  $k = 0, 1, \dots, \lceil (t-4)/5 \rceil$ ,  $p = 0, 1, \dots, m-1$ ;
- 2) Initialize  $A_{k,0} = f_{5k}$ ,  $k = 0, 1, \dots, \lceil (t-4)/5 \rceil$ , and  $f(0) = f_0$ ;
- 3) Compute  $A_{k,j} = A_{k,j-1} + b_{k,\delta(\alpha_j, \alpha_{j-1})}$ ,  $k = 0, 1, \dots, \lceil (t-4)/5 \rceil$ ,  $j = 0, 1, \dots, 2^m-2$ ;
- 4) Compute  $f(\alpha_j) = f_3(\alpha_j)^3 + \sum_{k=0}^{\lceil (t-4)/5 \rceil} (\alpha_j)^{5k} A_{k,j}$ ,  $j = 0, 1, \dots, 2^m-2$ .

If  $f(\alpha_j) = 0$  then  $\alpha_j$  is a root of the polynomial  $f(x)$ .

### B. COMPUTATIONAL COMPLEXITY

The computational complexity of the Fedorenko–Trifonov method over  $GF(2^m)$  consists of three parts: the number of multiplications, the number of additions, and the number of exponentiations.

According to [17, eq. (2)] the multiplicative complexity is

$$4m \left\lceil \frac{t+1}{5} \right\rceil + \left\lceil \frac{t+1}{5} \right\rceil (2^m - 1),$$

the additive complexity is

$$3m \left\lceil \frac{t+1}{5} \right\rceil + 2 \left\lceil \frac{t+1}{5} \right\rceil (2^m - 1),$$

and the exponential complexity is

$$2(2^m - 1).$$

In this method the exponentiation means calculating  $(\alpha_j)^3$  and  $(\alpha_j)^5$  for all nonzero elements of the finite field  $GF(2^m)$ . One way to do these calculations is  $(\alpha_j)^3 = (\alpha_j)^2(\alpha_j)$  and  $(\alpha_j)^5 = (\alpha_j)^3(\alpha_j)^2$  with complexity equal to  $2 + 1 = 3$  multiplications. Other options for implementing these calculations are possible too.

The computational complexity for examples of this method are shown in Table 6 and Table 7.

Other variants of the affine decomposition for polynomial of degrees  $t = 8$  and  $t = 16$  are published in [18] and [15]:

$$\begin{aligned} f(x) &= (f_0 + f_1 x + f_2 x^2 + f_4 x^4 + f_8 x^8) \\ &\quad + x^3 \left( (f_3 + f_5 x^2 + f_7 x^4) + f_6 x^3 \right), \\ f(x) &= (f_0 + f_1 x + f_2 x^2 + f_4 x^4 + f_8 x^8 + f_{16} x^{16}) \\ &\quad + x^3 \left( (f_3 + f_5 x^2 + f_7 x^4) \right. \\ &\quad \left. + x^3 \left[ f_6 + x^3 \left\{ (f_9 + f_{10}x + f_{11}x^2 + f_{13}x^4) \right. \right. \right. \\ &\quad \left. \left. \left. + x^3 \left( (f_{12} + f_{14}x^2) + f_{15}x^3 \right) \right\} \right] \right). \end{aligned}$$

### III. CYCLOTOMIC DFT ALGORITHM

This Section is based on the method [32].

#### A. BASIC NOTIONS AND DEFINITIONS

**Definition 3:** The discrete Fourier transform (DFT) of length  $n$  of a vector  $\mathbf{f} = (f_i)$ ,  $i = 0, 1, \dots, n-1$ ,  $f_i \in GF(2^m)$ ,  $n \mid (2^m - 1)$ , in the field  $GF(2^m)$  is the vector  $\mathbf{F} = (F_j)$ ,

$$F_j = \sum_{i=0}^{n-1} f_i \alpha^{ij}, \quad j = 0, 1, \dots, n-1,$$

where  $\alpha$  is an element of order  $n$  in  $GF(2^m)$  and a transform kernel.

We assume that the length of the  $n$ -point Fourier transform over  $GF(2^m)$  is  $n = 2^m - 1$ . Let  $\alpha$  be a primitive element of the field  $GF(2^m)$ . Every vector  $\mathbf{f} = (f_i)$ ,  $i = 0, 1, \dots, n-1$ , is associated with a polynomial  $f(x) = \sum_{i=0}^{n-1} f_i x^i$ , and we have  $F_j = f(\alpha^j)$ . The field of the computation is the finite field  $GF(2^m)$ .

**Definition 4:** The set  $C_k = \{c_k, c_k 2, c_k 2^2, \dots, c_k 2^{m_k-1}\}$  is the cyclotomic coset modulo  $n$  over  $GF(2)$ , where  $c_k \equiv c_k 2^{m_k} \pmod{n}$ ,  $c_k$  is a generator of the  $k$ th cyclotomic coset  $C_k$ ,  $m_k$  is a cardinality of the  $k$ th cyclotomic coset  $C_k$ ,  $m_k \mid m$ ,  $C_0 = \{c_0\} = \{0\}$ ,  $l + 1$  is the number of cyclotomic cosets modulo  $n$  over  $GF(2)$ ,  $k = 0, 1, \dots, l$ .

**B. CYCLOTOMIC ALGORITHM**

The cyclotomic algorithm is based on representing an original polynomial  $f(x)$  as a sum of linearized polynomials  $L_k(x)$  (cyclotomic decomposition of the polynomial), finding their values in a set of basis points  $\Gamma_k$  (cyclic convolution), and computing the resulting vector as a linear combination of these values with coefficients  $a_{j k p}$  (see (3)) from a prime field (multiplication of a binary matrix by a vector).

The cyclotomic algorithm consists of three steps:

0) (preliminary step) decomposing an original polynomial into a sum of linearized polynomials

$$f(x) = f_0 + \sum_k L_k(x);$$

1) evaluating the linearized polynomials at a set of basis points

$$\{L_k(\gamma_p)\}, \quad \gamma_p \in \Gamma_k;$$

2) components of the Fourier transform are computed as linear combinations of these values with coefficients from a prime field

$$F_j = f_0 + \sum_{k,p} L_k(\gamma_p), \quad j = 0, 1, \dots, n - 1.$$

Consider these steps in detail.

0) CYCLOTOMIC DECOMPOSITION

The polynomial  $f(x) = \sum_{i=0}^{n-1} f_i x^i$ ,  $f_i \in GF(2^m)$ , can be decomposed as

$$f(x) = f_0 + \sum_{k=1}^l L_k(x^{c_k}), \quad L_k(y) = \sum_{s=0}^{m_k-1} f_{c_k 2^s \pmod{n}} y^{2^s},$$

$$f(x) = f_0 + \sum_{k=1}^l \sum_{s=0}^{m_k-1} f_{c_k 2^s} x^{c_k 2^s}.$$

1) LINEARIZED POLYNOMIAL EVALUATION AT THE BASIS POINTS

Let  $\Gamma_k = (\gamma_k, \gamma_k^2, \gamma_k^4, \dots, \gamma_k^{2^{m_k-1}})$  be a normal basis for each subfield  $GF(2^{m_k}) \subset GF(2^m)$  over  $GF(2)$ , where  $\gamma_k \in GF(2^{m_k})$  is a generator of this normal basis. Each of the linearized polynomials  $L_k(y)$  can be evaluated at the basis points of the corresponding subfield by the formula

$$L_k(\gamma_k^{2^p}) = \sum_{s=0}^{m_k-1} \gamma_k^{2^{p+s}} f_{c_k 2^s},$$

$$k = 1, 2, \dots, l, \quad p = 0, 1, \dots, m_k - 1.$$

Let us rewrite the latter formula in matrix form:

$$\begin{pmatrix} L_k(\gamma_k^{2^0}) \\ L_k(\gamma_k^{2^1}) \\ \dots \\ L_k(\gamma_k^{2^{m_k-1}}) \end{pmatrix} \stackrel{\text{def}}{=} \mathcal{L}_k f[c_k],$$

where

$$\mathcal{L}_k \stackrel{\text{def}}{=} \begin{pmatrix} \gamma_k^{2^0} & \gamma_k^{2^1} & \dots & \gamma_k^{2^{m_k-1}} \\ \gamma_k^{2^1} & \gamma_k^{2^2} & \dots & \gamma_k^{2^0} \\ \dots & \dots & \dots & \dots \\ \gamma_k^{2^{m_k-1}} & \gamma_k^{2^0} & \dots & \gamma_k^{2^{m_k-2}} \end{pmatrix} \quad (1)$$

and

$$f[c_k] \stackrel{\text{def}}{=} \begin{pmatrix} f_{c_k 2^0} \\ f_{c_k 2^1} \\ \dots \\ f_{c_k 2^{m_k-1}} \end{pmatrix}. \quad (2)$$

It means the calculation of a normalized cyclic convolution of length  $m_k$  [21]. The matrix  $\mathcal{L}_k$  is called a basis circulant matrix [19], because its first row is a normal basis.

2) COMPUTING THE RESULTING VECTOR  $F = (F_j)$ ,  $j = 0, 1, \dots, n - 1$ , AS A LINEAR COMBINATION OF  $L_k(\gamma_k^{2^p})$   
The element  $(\alpha^j)^{c_k} = (\alpha^{c_k})^j \in GF(2^{m_k})$  can be decomposed with respect to basis  $\Gamma_k$  of the subfield  $GF(2^{m_k})$ :

$$\alpha^{j c_k} = \sum_{p=0}^{m_k-1} a_{j k p} \gamma_k^{2^p}, \quad a_{j k p} \in GF(2). \quad (3)$$

According to the main property of the linearized polynomials, we have

$$L_k(\alpha^{j c_k}) = \sum_{p=0}^{m_k-1} a_{j k p} L_k(\gamma_k^{2^p}).$$

Components of the Fourier transform of a polynomial  $f(x)$  are linear combinations of these values:

$$F_j = f(\alpha^j) = f_0 + \sum_{k=1}^l L_k((\alpha^j)^{c_k})$$

$$= f_0 + \sum_{k=1}^l \sum_{p=0}^{m_k-1} a_{j k p} L_k(\gamma_k^{2^p})$$

$$= f_0 + \sum_{k=1}^l \sum_{p=0}^{m_k-1} a_{j k p} \left( \sum_{s=0}^{m_k-1} \gamma_k^{2^{p+s}} f_{c_k 2^s} \right),$$

$$j = 0, 1, \dots, n - 1, \quad a_{j k p} \in GF(2).$$

**C. MATRIX FORM OF THE CYCLOTOMIC ALGORITHM**

The DFT calculation is divided into two steps: multiplying the block diagonal matrix  $\mathcal{L}$  by the vector  $\mathbf{\Pi}f$  and multiplying the binary matrix  $\mathbf{A} = (a_{j k p})$  by the vector  $\mathcal{L}(\mathbf{\Pi}f)$ :

$$\mathbf{F} = \mathbf{A} \mathcal{L}(\mathbf{\Pi}f), \quad (4)$$

where the block diagonal matrix

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathcal{L}_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & \mathcal{L}_l \end{pmatrix},$$

$\mathcal{L}_0 = 1$ ,  $\mathbf{O}$  denotes all-zero matrices of appropriate dimensions, and

$$\Pi f \stackrel{\text{def}}{=} \begin{pmatrix} f_0 \\ f[c_1] \\ f[c_2] \\ \dots \\ f[c_l] \end{pmatrix} \quad (5)$$

is a permutation of the original vector  $f$ , where  $f[c_k]$ ,  $k = 1, 2, \dots, l$ , is defined by formula (2).

**IV. TRUNCATED NORMALIZED CYCLIC CONVOLUTION**

**A. NORMALIZED CYCLIC CONVOLUTION**

First, we present auxiliary Lemma on the connection of the normal bases for subfields.

*Lemma 1 ([23]):* Let  $(\gamma^{2^0}, \gamma^{2^1}, \dots, \gamma^{2^{m-1}})$  be the normal basis for the field  $GF(2^m)$  over  $GF(2)$ ,  $m_k \mid m$ , and  $\varepsilon_p = \sum_{s=0}^{m/m_k-1} \gamma^{2^{sm_k+p}}$ ,  $p = 0, 1, \dots, m_k - 1$ ,

then  $(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{m_k-1})$  is the normal basis for subfield  $GF(2^{m_k}) \subset GF(2^m)$  over  $GF(2)$ .

To each cyclotomic coset  $C_k$ ,  $k = 0, 1, \dots, l$ , assign the basis circulant  $\mathcal{L}_k$ , which is defined by formula (1). For each cardinality  $m_k \mid m$  of the cyclotomic coset we construct the basis circulant  $\Lambda_{m_k}$  of dimension  $m_k \times m_k$ . Secondly, we choose the normal basis for  $GF(2^m)$  over  $GF(2)$ :  $\Gamma(m) = (\gamma, \gamma^2, \gamma^{2^2}, \dots, \gamma^{2^{m-1}})$  and construct the normal bases  $\Gamma(m_k) = (\varepsilon, \varepsilon^2, \varepsilon^{2^2}, \dots, \varepsilon^{2^{m_k-1}})$  for each subfield  $GF(2^{m_k}) \subset GF(2^m)$  over  $GF(2)$ , which we will choose according to the condition of Lemma 1. Further, we define the basis circulants  $\Lambda_{m_k}$  for all dimensions  $m_k \times m_k$ ,  $m_k \mid m$ , as

$$\Lambda_{m_k} \stackrel{\text{def}}{=} \left( (\varepsilon^{2^{p+s}}), p, s = 0, 1, \dots, m_k - 1, \varepsilon \in \Gamma(m_k) \right) \quad (6)$$

over the corresponding subfields  $GF(2^{m_k})$ . Then all  $\mathcal{L}_k \in \{\Lambda_{m_k} : m_k \mid m\}$ .

The next theorem is a modification of theorem [21, Th. 1].

*Theorem 2:* For any finite field  $GF(2^m)$  with even  $m$  there exists a decomposition of the  $m \times m$  basis circulant

$$\begin{aligned} \Lambda_m &= \left( (\gamma^{2^{p+s}}), p, s = 0, 1, \dots, m - 1 \right) \\ &= \begin{pmatrix} \gamma^{2^0} & \gamma^{2^1} & \dots & \gamma^{2^{m-1}} \\ \gamma^{2^1} & \gamma^{2^2} & \dots & \gamma^{2^0} \\ \dots & \dots & \dots & \dots \\ \gamma^{2^{m-1}} & \gamma^{2^0} & \dots & \gamma^{2^{m-2}} \end{pmatrix} \end{aligned}$$

$$\begin{aligned} &= \mathbf{B} \left( \begin{array}{c|c} \Lambda_{m/2} & \mathbf{O} \\ \mathbf{O} & \Lambda_{m/2} \end{array} \right) \\ &\times \left( \begin{array}{c|c} \mathbf{I}_{m/2} & \mathbf{O} \\ \alpha^{c_k} & \\ \alpha^{c_k 2} & \\ \alpha^{c_k 2^2} & \\ \dots & \\ \alpha^{c_k 2^{m/2-1}} & \end{array} \middle| \begin{array}{c} \mathbf{O} \\ \mathbf{I}_{m/2} \end{array} \right) \\ &\times \left( \begin{array}{c|c} \mathbf{I}_{m/2} & \mathbf{I}_{m/2} \\ \mathbf{O} & \mathbf{I}_{m/2} \end{array} \right), \end{aligned}$$

where  $\gamma$  is the generator of the normal basis  $\Gamma(m) = (\gamma, \gamma^2, \gamma^{2^2}, \dots, \gamma^{2^{m-1}})$  for  $GF(2^m)$  over  $GF(2)$ ,  $\mathbf{B}$  is an  $m \times m$  nonsingular binary matrix,  $\Lambda_{m/2} = (\varepsilon^{2^{p+s}}) = \left( (\gamma + \gamma^{2^{m/2}})^{2^{p+s}} \right)$ ,  $p, s = 0, 1, \dots, m/2 - 1$ , is an  $m/2 \times m/2$  basis circulant over  $GF(2^{m/2})$  (according to Lemma 1),  $\mathbf{O}$  is an  $m/2 \times m/2$  all-zero matrix,  $\mathbf{I}_{m/2}$  is an  $m/2 \times m/2$  identity matrix,  $c_k$  is a generator of the cyclotomic coset  $C_k$  of cardinality  $m$  satisfying the conditions  $\alpha^{c_k} = \alpha^{c_k 2^{m/2}} + 1$  [21, Th. 2].

Note that for the basis circulant  $\Lambda_{m_k}$  with dimension smaller than  $m \times m$  this basis circulant  $\Lambda_{m_k}$  belongs to the subfield  $GF(2^{m_k}) \subset GF(2^m)$  and can be decomposed over this subfield in the same way as the basis circulant  $\Lambda_m$ .

Further, while decomposing the matrix  $\Lambda_m$  we execute the decomposition of the matrix  $\Lambda_{m/2}$ . The latter operation is executed recursively or we can apply short cyclic convolutions from [6, Fig. 11.1] for small odd dimensions of this basis circulant. Note that there are no general algorithms for efficient short cyclic convolutions over finite fields except [4], [6, Fig. 11.1], [7], [12], [21], [40], [41]. Finally, we obtain the decomposition of the initial matrix  $\Lambda_m$  into  $\Lambda_m = \mathcal{P}_m \mathcal{S}_m$ , where  $\mathcal{P}_m$  is the matrix of postadditions.

**B. EXAMPLES**

*Example 1 (Normalized cyclic convolution of length  $m = 2$ ):*

The finite field  $GF(2^2)$  is defined by an element  $\omega$ , which is a root of the primitive polynomial  $x^2 + x + 1$ . Let  $\Gamma(2) = (\omega, \omega^2)$  be a normal basis for the field  $GF(2^2)$  over  $GF(2)$ . Let  $c_k = 1$ . Let us write the decomposition of the  $2 \times 2$  basis circulant according to Theorem 2

$$\begin{aligned} \Lambda_2 &= \begin{pmatrix} \omega^1 & \omega^2 \\ \omega^2 & \omega^1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \omega^1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \omega^1 & \omega^2 \end{pmatrix} = \mathcal{P}_2 \mathcal{S}_2, \end{aligned}$$

where  $\mathcal{P}_2$  is the binary matrix of postadditions for the calculation of the normalized cyclic convolution of length  $m = 2$ .

*Example 2 (Normalized cyclic convolution of length  $m = 4$ ):*

The finite field  $GF(2^4)$  is defined by an element  $\alpha$ , which is a root of the primitive polynomial  $x^4 + x + 1$ . Let  $\Gamma(4) = (\gamma, \gamma^2, \gamma^4, \gamma^8)$  and  $\Gamma(2) = (\gamma + \gamma^4, \gamma^2 + \gamma^8) = (\alpha^5, \alpha^{10})$  be the normal bases for the field  $GF(2^4)$  and the subfield  $GF(2^2) \subset GF(2^4)$  over  $GF(2)$ , respectively, where  $\gamma = \alpha^6$ . Let  $c_k = 1$ . Let us write the decomposition of the  $4 \times 4$  basis circulant according to Theorem 2

$$\begin{aligned} \Lambda_4 &= \begin{pmatrix} \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 \\ \alpha^{12} & \alpha^9 & \alpha^3 & \alpha^6 \\ \alpha^9 & \alpha^3 & \alpha^6 & \alpha^{12} \\ \alpha^3 & \alpha^6 & \alpha^{12} & \alpha^9 \end{pmatrix} \\ &= \begin{pmatrix} 0010 \\ 1101 \\ 1010 \\ 1001 \end{pmatrix} \left( \begin{array}{cc|cc} \alpha^5 & \alpha^{10} & 0 & 0 \\ \alpha^{10} & \alpha^5 & 0 & 0 \\ \hline 0 & 0 & \alpha^5 & \alpha^{10} \\ 0 & 0 & \alpha^{10} & \alpha^5 \end{array} \right) \\ &\quad \times \left( \begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array} \right) \left( \begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array} \right) \\ &= \begin{pmatrix} 0001 \\ 1011 \\ 0101 \\ 0111 \end{pmatrix} \left( \begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ \alpha^5 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & \alpha^5 & 1 \end{array} \right) \left( \begin{array}{c|c} 1 & 1 \\ \hline 0 & 1 \\ \hline 0 & 0 \\ 0 & 1 \end{array} \right) \\ &\quad \times \left( \begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array} \right) \left( \begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array} \right) \\ &= \begin{pmatrix} 0001 \\ 1011 \\ 0101 \\ 0111 \end{pmatrix} \left( \begin{array}{cccc} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^5 & \alpha^{10} & \alpha^5 & \alpha^{10} \\ \alpha^1 & \alpha^2 & \alpha^4 & \alpha^8 \\ \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 \end{array} \right) = \mathcal{P}_4 \mathcal{S}_4, \end{aligned}$$

where  $\mathcal{P}_4$  is the binary matrix of postadditions for the calculation of the normalized cyclic convolution of length  $m = 4$ .

The decomposition of the matrix  $\Lambda_4$  is calculated recursively using equality  $\alpha^5 = \omega \in GF(2^2) \subset GF(2^4)$  and the decomposition for the matrix  $\Lambda_2$  (see Example 1). Matrices of postadditions for the multiplication by the matrix  $\Lambda_2$  are absorbed into the first binary matrix for the decomposition of  $\Lambda_4$  (see Theorem 2).

**C. TRUNCATED MATRIX**

According to formulae (1) and (6), the matrix  $\Lambda_{m_k} = \mathcal{L}_k$  is the basis circulant matrix, and the multiplication by this matrix is the calculation of a normalized cyclic convolution. From  $\Lambda_{m_k} \stackrel{\text{def}}{=} (\Lambda_{left} \mid \Lambda_{right})$  and  $f[c_k] \stackrel{\text{def}}{=} \begin{pmatrix} f[c_k]_{top} \\ f[c_k]_{bottom} \end{pmatrix}$  it follows that the matrix  $\Lambda_{left}$  is a left truncated matrix, and  $\Lambda_{left} f[c_k]_{top}$  is the calculation of a truncated normalized cyclic convolution.

*Example 3 (Truncated normalized cyclic convolution of length  $m = 4$  for two nonzero consecutive input components):*

From  $\Lambda_4 = \mathcal{P}_4 \mathcal{S}_4 \implies$

$$\begin{aligned} \Lambda_4 \begin{pmatrix} f_1 \\ f_2 \\ 0 \\ 0 \end{pmatrix} &= \mathcal{P}_4 \left[ \mathcal{S}_4 \begin{pmatrix} f_1 \\ f_2 \\ 0 \\ 0 \end{pmatrix} \right] \implies \\ \mathcal{S}_4 \begin{pmatrix} f_1 \\ f_2 \\ 0 \\ 0 \end{pmatrix} &= \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^5 & \alpha^{10} & \alpha^5 & \alpha^{10} \\ \alpha^1 & \alpha^2 & \alpha^4 & \alpha^8 \\ \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ 0 \\ 0 \end{pmatrix} \implies \\ &\quad \times \begin{pmatrix} \alpha^0 & \alpha^0 \\ \alpha^5 & \alpha^{10} \\ \alpha^1 & \alpha^2 \\ \alpha^6 & \alpha^{12} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha^5 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & \alpha^5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \hline \alpha^1 & 0 \\ 0 & \alpha^2 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \end{aligned}$$

Notes. In this Example we do not perform the multiplication by the matrix of postadditions  $\mathcal{P}_4$ . In the latter formula zero rows and columns of matrices are deleted.

**V. NOVEL METHOD FOR FINDING ROOTS OF POLYNOMIALS**

**A. DERIVATION OF ANOTHER FORMULA FOR THE CYCLOTOMIC DFT ALGORITHM**

Using Theorem 2 and a recursion for calculating the decomposition of the basis circulant  $\Lambda_{m_k}, m_k \mid m$ , we obtain that the decomposition of this matrix has the form  $\Lambda_{m_k} = \mathcal{P}_{m_k} \mathcal{S}_{m_k}$ , where  $\mathcal{P}_{m_k}$  is the matrix of postadditions. Then each matrix  $\mathcal{L}_k, k = 0, 1, \dots, l$ , can be decomposed as

$$\mathcal{L}_k = \mathbf{P}_k \mathbf{S}_k, \tag{7}$$

where  $\mathbf{P}_k \in \{\mathcal{P}_{m_k} : m_k \mid m\}$  and  $\mathbf{S}_k \in \{\mathcal{S}_{m_k} : m_k \mid m\}$ .

We rewrite the matrix form of the cyclotomic DFT algorithm (4) as follows:

$$\begin{aligned} F &= \mathbf{A} \mathcal{L}(\Pi f) \\ &= \left( \mathbf{A} \begin{pmatrix} P_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & P_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & P_l \end{pmatrix} \right) \begin{pmatrix} S_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & S_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & S_l \end{pmatrix} \\ &\quad \times (\Pi f) \stackrel{\text{def}}{=} \mathbf{Post} \mathbf{S}(\Pi f), \end{aligned}$$

where  $\mathbf{P}_k \in \{\mathcal{P}_{m_k} : m_k \mid m\}, k = 0, 1, \dots, l$ , is the  $m_k \times m_k$  nonsingular binary matrix of postadditions for the calculation of the normalized cyclic convolution,  $\mathbf{S}_k$  is the  $m_k \times m_k$  second matrix in decomposition (7),  $\mathbf{O}$  denotes all-zero matrices of

appropriate dimensions,  $\mathbf{S}$  is a block diagonal matrix consisting of matrices  $\mathbf{S}_k$ ,  $\mathbf{Post} \stackrel{\text{def}}{=} \mathbf{A} \begin{pmatrix} P_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & P_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & P_l \end{pmatrix}$  is the binary matrix of postadditions for the calculation of the DFT. Taking into account (5), we obtain

$$\mathbf{F} = \mathbf{Post} \begin{pmatrix} S_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & S_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & S_l \end{pmatrix} \begin{pmatrix} f_0 \\ f[c_1] \\ f[c_2] \\ \dots \\ f[c_l] \end{pmatrix}. \quad (8)$$

The matrices of postadditions  $\mathbf{P}_k$ ,  $k = 0, 1, \dots, l$ , are absorbed into the binary matrix  $\mathbf{Post}$ . The multiplication by the matrix  $\mathbf{S}_k \in \{\mathcal{S}_{m_k} : m_k \mid m\}$  we call the calculation of a normalized cyclic convolution too.

**B. CYCLOTOMIC COSETS CHOICE**

We can choose the cyclotomic cosets, so that the following properties are satisfied:

- 1) For any cyclotomic coset  $C_k$  the generator  $c_k$  is selected as the smallest number in the cyclotomic coset;
- 2) The cyclotomic cosets are ordered by increasing their generators.

**C. UNION OF THE CYCLOTOMIC COSETS C**

Let us introduce for the degree  $t$  of the error-locator polynomial the union of the cyclotomic cosets  $\mathcal{C}$  as follows:

$$C_k \subset \mathcal{C} \iff \exists i: \begin{cases} i \in C_k \\ i = 0, 1, \dots, t. \end{cases}$$

Let  $k_{\max}$  be the maximum value of the index  $k$  of the cyclotomic coset  $C_k$ :

$$k_{\max} = \max \{k : C_k \subset \mathcal{C}\},$$

then the union of the cyclotomic cosets is

$$\mathcal{C} = \bigcup_{k=0}^{k_{\max}} C_k$$

and we define a number  $v$  as a cardinality of the union of the cyclotomic cosets  $\mathcal{C}$ :

$$v = \sum_{k=0}^{k_{\max}} m_k.$$

**D. TRUNCATED CYCLOTOMIC DFT**

*Definition 5:* Assume that the input polynomial  $f(x)$  for the cyclotomic DFT algorithm has degree  $t$ , then coefficients  $f_i = 0$  for  $i = t + 1, t + 2, \dots, n - 1$ . We call the DFT for this input polynomial a truncated cyclotomic DFT.

Using (8), we get the notation for the truncated cyclotomic DFT

$$\mathbf{F} = \mathbf{Post}[0..v - 1]$$

$$\begin{aligned} & \times \begin{pmatrix} S_0 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & S_1 & \dots & \mathbf{O} \\ \dots & \dots & \ddots & \dots \\ \mathbf{O} & \mathbf{O} & \dots & S_l \end{pmatrix} \begin{pmatrix} f_0 \\ f[c_1] \\ f[c_2] \\ \dots \\ f[c_{k_{\max}}] \end{pmatrix} \\ & \stackrel{\text{def}}{=} \mathbf{Post}[0..v - 1] \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \dots \\ r_{k_{\max}} \end{pmatrix} \stackrel{\text{def}}{=} \mathbf{Post}[0..v - 1]\mathbf{r}, \quad (9) \end{aligned}$$

where  $\mathbf{Post}[0..v - 1]$  is the first  $v$  columns of the matrix  $\mathbf{Post}$ ,  $\mathbf{O}$  denotes all-zero matrices of appropriate dimensions, the calculation result of the (truncated) normalized cyclic convolution  $\mathbf{r}_k = \mathbf{S}_k f[c_k]$ ,  $k = 1, 2, \dots, k_{\max}$ , is the column vector,  $r_0 = S_0 f_0 = f_0$ .

**E. ALGORITHM FOR THE TRUNCATED CYCLOTOMIC DFT CALCULATION**

Let us describe the algorithm for the truncated cyclotomic DFT calculation more formally.

*Algorithm input:* the length of the DFT over  $GF(2^m)$  is  $n = 2^m - 1$ ,  $t$  is a degree of the error-locator polynomial, the input components  $f_i$ ,  $i = 0, 1, \dots, n - 1$ ,  $f_i \in GF(2^m)$ , correspond to the vector  $\mathbf{f} = (f_i)$  or the error-locator polynomial  $f(x)$ . We assume that  $f_i = 0$  for  $i > t$ .

*Algorithm output:* the output vector of the DFT  $\mathbf{F} = (F_j)$ ,  $i = 0, 1, \dots, n - 1$ ,  $F_i \in GF(2^m)$ .

*The preliminary calculations:* calculation of  $k_{\max}$ , the union of the cyclotomic cosets  $\mathcal{C}$ , and the cardinality of the union of the cyclotomic cosets  $v$  (V-C), the matrices  $\mathbf{S}_k$ ,  $k = 0, 1, \dots, k_{\max}$  (7), and the submatrix  $\mathbf{Post}[0..v - 1]$  (9).

The algorithm consists of two steps.

- 1) Compute the (truncated) normalized cyclic convolutions  $r_0 = f_0$ ;  
 $\mathbf{r}_k = \mathbf{S}_k f[c_k]$ ,  $k = 1, 2, \dots, k_{\max}$ .

If  $c_k 2^s > t$  for some number  $s \in \{0, 1, \dots, m_k - 1\}$  then  $f_{c_k 2^s} = 0$  and we compute  $\mathbf{r}_k$  as the truncated normalized cyclic convolution.

- 2) Multiplying the binary matrix  $\mathbf{Post}[0..v - 1]$  by the

$$\text{vector } \mathbf{r} = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \dots \\ r_{k_{\max}} \end{pmatrix} \text{ from an extended finite field } GF(2^m): \mathbf{F} = \mathbf{Post}[0..v - 1]\mathbf{r}.$$

The algorithm *pseudocode* representation is

- 1) Computing the truncated normalized cyclic convolutions  $\mathbf{r}_k = \mathbf{S}_k f[c_k]$ ,  $k = 0, 1, \dots, k_{\max}$ .
- 2) Multiplying the binary matrix  $\mathbf{Post}[0..v - 1]$  by the vector  $\mathbf{r}$ .

*Example 4 (DFT for length  $n = 15$ ):*

See conditions in Example 2. From formula (4), we get the matrix form of the cyclotomic algorithm (10), as shown at the bottom of the next page.

We continue to consider the Example for the degree of polynomial  $t = \deg f(x) = 5$ .  $k_{\max} = 3$ . The union of the cyclotomic cosets is

$$\begin{aligned} \mathcal{C} &= C_0 \cup C_1 \cup C_3 \cup C_5 \\ &= \{0\} \cup \{1, 2, 4, 8\} \cup \{3, 6, 12, 9\} \cup \{5, 10\}. \end{aligned}$$

The cardinality of  $\mathcal{C}$  is  $v = 11$ . We take into account that the binary matrix of postadditions for the calculation of the normalized cyclic convolution  $\mathbf{P}_k$ ,  $k = 0, 1, \dots, l$ , can be absorbed into the binary matrix  $\mathbf{Post}$  of the DFT calculation. Further, we rewrite formula (9) of the truncated cyclotomic DFT as  $\mathbf{F} = \mathbf{Post} \tilde{\mathbf{S}}(\tilde{\mathbf{N}}f)$  (see (11)), as shown at the bottom of the next page, where zero rows and columns of matrices are deleted.

Consider the first step of the algorithm for the truncated cyclotomic DFT calculation.

- 0) Compute the trivial convolution  $r_0 = f_0$ ;
- 1) Compute the column vector  $\mathbf{r}_1$  of length 4 as the truncated normalized cyclic convolution:

$$\mathbf{r}_1 = \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^5 & \alpha^{10} & \alpha^5 \\ \alpha^1 & \alpha^2 & \alpha^4 \\ \alpha^6 & \alpha^{12} & \alpha^9 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_4 \end{pmatrix}$$

$$\begin{aligned} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha^5 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \alpha^5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &\times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha^1 & 0 & 1 \\ 0 & \alpha^2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_4 \end{pmatrix}; \end{aligned}$$

- 2) Compute the column vector  $\mathbf{r}_2$  of length 4 as the truncated normalized cyclic convolution:

$$\begin{aligned} \mathbf{r}_2 &= \begin{pmatrix} \alpha^0 \\ \alpha^5 \\ \alpha^1 \\ \alpha^6 \end{pmatrix} f_3 \\ &= \begin{pmatrix} 1 & 0 \\ \alpha^5 & 0 \\ 0 & 1 \\ 0 & \alpha^5 \end{pmatrix} \begin{pmatrix} 1 \\ \alpha^1 \end{pmatrix} f_3 = \begin{pmatrix} f_3 \\ \alpha^5 f_3 \\ \alpha^1 f_3 \\ \alpha^6 f_3 \end{pmatrix}; \end{aligned}$$

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^{12} & \alpha^9 & \alpha^3 & \alpha^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^9 & \alpha^3 & \alpha^6 & \alpha^{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^3 & \alpha^6 & \alpha^{12} & \alpha^9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^{12} & \alpha^9 & \alpha^3 & \alpha^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^9 & \alpha^3 & \alpha^6 & \alpha^{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^3 & \alpha^6 & \alpha^{12} & \alpha^9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^5 & \alpha^{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{12} & \alpha^9 & \alpha^3 & \alpha^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^9 & \alpha^3 & \alpha^6 & \alpha^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^3 & \alpha^6 & \alpha^{12} & \alpha^9 & 0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_8 \\ f_3 \\ f_6 \\ f_{12} \\ f_9 \\ f_5 \\ f_{10} \\ f_7 \\ f_{14} \\ f_{13} \\ f_{11} \end{pmatrix} \tag{10}$$



3) Compute the column vector  $r_3$  of length 2 as the truncated normalized cyclic convolution:

$$r_3 = \begin{pmatrix} \alpha^0 \\ \alpha^5 \end{pmatrix} f_5 = \begin{pmatrix} f_5 \\ \alpha^5 f_5 \end{pmatrix}.$$

Note that the calculation of the second and the third convolutions is the trivial case since we have only one nonzero input component.

Finally, the second step of the algorithm for the truncated cyclotomic DFT calculation is multiplying the binary matrix

$$\tilde{\mathbf{Post}} \text{ by the vector } \mathbf{r} = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} \text{ over } GF(2^4).$$

Consider the computational complexity of the current Example.

The first step is the calculation of the four truncated normalized cyclic convolutions. It requires  $0 + 4 + 3 + 1 = 8$  multiplications and  $0 + 6 + 0 + 0 = 6$  additions.

The second step is the multiplication of the binary matrix  $\tilde{\mathbf{Post}}$  by the vector  $\mathbf{r}$  over  $GF(2^4)$ . If we use a heuristic algorithm [33] then the complexity is 34 additions. Other methods for multiplication of a binary matrix by a vector have about the same additive complexity.

The full computational complexity of Example is shown in Table 6.

Consider another nontrivial Example of convolution computation.

*Example 5:* Consider the calculation of the polynomial roots over  $GF(2^8)$  when the degree of polynomial  $t = \text{deg}f(x) = 33$ . This Example is interesting because according to Table 5 the input components in the truncated normalized cyclic convolutions cease being consecutive. In general, the calculations are the same as in the  $t = 32$  case except for one calculation of the truncated normalized cyclic convolution. Consider this nontrivial case in detail.

The finite field  $GF(2^8)$  is defined by an element  $\alpha$ , which is a root of the primitive polynomial  $x^8 + x^4 + x^3 + x^2 + 1$ .

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^0 & \alpha^0 & \alpha^0 & 0 & 0 & 0 \\ 0 & \alpha^5 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 \\ 0 & \alpha^1 & \alpha^2 & \alpha^4 & 0 & 0 & 0 \\ 0 & \alpha^6 & \alpha^{12} & \alpha^9 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \alpha^0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^5 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^6 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & \alpha^0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^5 & 0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_5 \end{pmatrix}. \tag{11}$$

$$r_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha^{85} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha^{85} & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^{85} & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{85} & 1 \end{pmatrix} \begin{pmatrix} 11 & 00 & 00 & 00 \\ 01 & 00 & 00 & 00 \\ \hline 00 & 11 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ \hline 00 & 00 & 11 & 00 \\ 00 & 00 & 01 & 00 \\ \hline 00 & 00 & 00 & 11 \\ 00 & 00 & 00 & 01 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline \alpha^{17} & 0 & 0 & 0 \\ 0 & \alpha^{34} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & \alpha^{17} & 0 \\ 0 & 0 & 0 & \alpha^{34} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline \alpha^7 & 0 & 0 \\ 0 & \alpha^{14} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_9 \\ f_{18} \\ f_{33} \end{pmatrix} \tag{12}$$

Let  $\Gamma(8) = (\gamma, \gamma^2, \gamma^4, \gamma^8, \gamma^{16}, \gamma^{32}, \gamma^{64}, \gamma^{128})$  be the normal basis for the field  $GF(2^8)$  over  $GF(2)$ , where  $\gamma = \alpha^5$ . The set  $C_9 = \{9, 18, 36, 72, 144, 33, 66, 132\}$  is the cyclotomic coset.

There are only two consecutive input components and one more input component for this case

$$\begin{pmatrix} f_9 \\ f_{18} \\ 0 \\ 0 \\ 0 \\ f_{33} \\ 0 \\ 0 \end{pmatrix}.$$

Using the matrix decomposition

$$\Lambda_8 = \mathcal{P}_8 \mathcal{S}_8,$$

where  $\mathcal{P}_8$  is the binary matrix of postadditions for the calculation of the normalized cyclic convolution of length  $m = 8$ , we compute the column vector  $r_5$  of length 8 as the truncated normalized cyclic convolution (12), as shown at the bottom of the previous page, which is based on the cyclotomic coset  $C_9$ .

Consider the computational complexity of the current Example. The calculation of this nontrivial convolution requires 10 multiplications and 10 additions only. According to Table 4 this is two additions more than for the convolution with two consecutive input components, and one multiplication and two additions less than for the convolution with three consecutive input components. This Example requires only two more additions in comparison to the case when  $t = 32$ .

## VI. COMPLEXITY OF THE TRUNCATED CYCLOTOMIC DFT COMPUTATION

### A. COMPLEXITY OF THE TRUNCATED NORMALIZED CYCLIC CONVOLUTIONS COMPUTATION

#### 1) COMPLEXITY OF THE MODIFICATION OF THE NORMALIZED CYCLIC CONVOLUTIONS COMPUTATION

From Theorem 2 it follows that normalized cyclic convolutions of length  $m$  computation consists of calculation of two normalized cyclic convolutions of length  $m/2$ , componentwise multiplication by a vector of length  $m/2$ , and two additions of vectors of length  $m/2$ . Note that the order of operations differs from the order of operations by [21, Th. 1], and besides the last operation will be the multiplication by the binary matrix  $\mathbf{B}$ . This binary matrix  $\mathbf{B}$  (or the product of several binary matrices at recursive computations) is absorbed into the binary matrix  $\mathbf{Post}$ .

Consider the method of obtaining the multiplicative and the additive complexity. The method is a modification and improvement of the result [21].

The recursive formulae for the number of multiplications and additions of the normalized cyclic convolution

calculation of length  $e2^i$  over  $GF(2^m)$  follow from Theorem 2

$$\begin{cases} \text{Mult}(e2^i) = 2\text{Mult}(e2^{i-1}) + e2^{i-1}; \\ \text{Add}(e2^i) = 2\text{Add}(e2^{i-1}) + e2^i; \end{cases} \quad i \geq 1,$$

where number  $e$  is an odd integer,  $e \geq 1$ , number  $i$  is an integer,  $i \geq 0$  for the convolution length and  $i \geq 1$  for the recursive formulae,  $h_m$  and  $h_a$  are the number of multiplications and additions for the convolution of odd length  $e$ , respectively, the initial conditions are  $\text{Mult}(e) = h_m$  and  $\text{Add}(e) = h_a$ .

These recursions are satisfied by

$$\begin{cases} \text{Mult}(e2^i) = (ie + 2 h_m) 2^{i-1}; \\ \text{Add}(e2^i) = (ie + h_a) 2^i; \end{cases} \quad i \geq 0.$$

The matrix of postadditions  $\mathcal{P}_{m_k}$ , where  $m_k = e2^i$  for some integer  $i$ , is absorbed into the binary matrix  $\mathbf{Post}$ . The multiplication by this matrix of postadditions  $\mathcal{P}_{m_k}$  is not included in the complexity of the normalized cyclic convolution computation.

#### 2) TRUNCATED NORMALIZED CYCLIC CONVOLUTIONS

Consecutive input components.

Recall the definition of the truncated normalized cyclic convolution  $\Lambda_{\text{left}} f[c_k]_{\text{top}}$ . If the column vector  $f[c_k]_{\text{top}}$  consists of nonzero components only, and the column vector  $f[c_k]_{\text{bottom}}$  is an all-zero vector, then this convolution is called the truncated normalized cyclic convolution with consecutive components.

For calculation of the truncated normalized cyclic convolution of length  $m_k$  with one component there is a simple algorithm which requires  $m_k - 1$  multiplications. From the normal basis property that the sum of all its components equals 1, it follows that the multiplicative complexity is reduced by one multiplication. This result cannot be improved. For calculation of the truncated normalized cyclic convolution of length  $m_k$  with any  $s$  components  $s(m_k - 1)$  multiplications are required. Of course, this result is not optimal.

The computational complexity of the truncated normalized cyclic convolution for lengths  $m = 2, 4, 8$  is shown in Table 2, Table 3, and Table 4, respectively. All input components in these convolutions are consecutive.

**TABLE 2. The computational complexity of the truncated normalized cyclic convolution for length  $m = 2$ .**

number of consecutive components	Mult	Add
1	1	0
2	1	2

#### 3) CONSECUTIVE INPUT COMPONENTS IN THE FINITE FIELDS

The minimal degrees  $t$  of polynomials for which the input components in the truncated normalized cyclic convolutions cease being consecutive are shown in Table 5. The length  $n = 2^m - 1$  is the length of the DFT over  $GF(2^m)$ .

**TABLE 3. The computational complexity of the truncated normalized cyclic convolution for length  $m = 4$ .**

number of consecutive components	Mult	Add
1	3	0
2	4	4
3	4	6
4	4	8

**TABLE 4. The computational complexity of the truncated normalized cyclic convolution for length  $m = 8$ .**

number of consecutive components	Mult	Add
1	7	0
2	10	8
3	11	12
4	12	16
5	12	18
6	12	20
7	12	22
8	12	24

**TABLE 5. The minimal degrees of polynomials for which the input components in the truncated normalized cyclic convolutions cease being consecutive.**

$n$	$t$	$C_k$
7	5	$C_3$
15	9	$C_3$
31	9	$C_5$
63	17	$C_5$
127	17	$C_9$
255	33	$C_9$
511	33	$C_{17}$
1023	65	$C_{17}$
2047	65	$C_{33}$
4095	129	$C_{33}$

For all error-locator polynomials whose degree is less than  $t$ , all input components in the truncated normalized cyclic convolutions are consecutive. The cyclotomic coset  $C_k$  is the cyclotomic coset modulo  $n$  over  $GF(2)$  for which  $t \in C_k$ .

4) CONDITIONS FOR CONSECUTIVE INPUT COMPONENTS

Let us write the conditions for all the input components to be consecutive.

For each cyclotomic coset  $C_k$  modulo  $n$  over  $GF(2)$  according to Definition 4 there is a set  $\widehat{C}_k = \{c_k, c_k 2, c_k 2^2, \dots, c_k 2^{m_k-1}\}$  of integers. Note that the set  $\widehat{C}_k$  lies in ordinary integer arithmetic.

*Lemma 2:* If for the set  $\widehat{C}_k, k = 1, 2, \dots, l$ ,

$$\exists s_{\min} : \begin{cases} s_{\min} = \min \{s : c_k 2^s > n\} \\ t \equiv c_k 2^{s_{\min}} \pmod n \end{cases},$$

where  $t$  is a degree of the error-locator polynomial, then input components do not have to be consecutive.

*Proof:* From  $c_k 2^{s_{\min}} > n$  and due to increasing the integer numbers of the ordered set  $\widehat{C}_k$ , we obtain  $c_k 2^{s_{\min}-1} < n$ . Further, we have

$$\begin{aligned} c_k 2^{s_{\min}-1} &= c_k 2^{s_{\min}} - c_k 2^{s_{\min}-1} < n; \\ c_k 2^{s_{\min}} - n &< c_k 2^{s_{\min}-1}; \\ (c_k 2^{s_{\min}} \pmod n) &< c_k 2^{s_{\min}-1}. \end{aligned}$$

Number  $c_k 2^{s_{\min}}$  in the cyclotomic coset  $C_k$  modulo  $n$  over  $GF(2)$  is less than number  $c_k 2^{s_{\min}-1}$  in the same cyclotomic coset. Thus in a cyclotomic coset the input components do not have to be consecutive. An exception (all input components are consecutive) is possible when in a cyclotomic coset the input components cease being consecutive several times. ■

*Lemma 3 (sufficient condition):* If all numbers  $i, i \in \{0, 1, \dots, t\}$ , where  $t$  is a degree of the error-locator polynomial, can be represented as

$$i = c_k 2^s, \quad c_k \in C_k \subset \mathcal{C}, \quad s \in \{0, 1, \dots, m_k - 1\},$$

then for all cyclotomic cosets  $C_k \subset \mathcal{C}$  all input components are consecutive.

*Proof:* Obviously, the proof follows from Lemma 2. ■

If the truncated normalized cyclic convolution has no consecutive input components, then we can either construct an algorithm for this case of calculation or complete the ‘‘phantom’’ input components of the convolution until the input components become consecutive.

**B. COMPLEXITY OF THE FIRST STEP FOR THE TRUNCATED CYCLOTOMIC DFT COMPUTATION ALGORITHM**

We construct complexity estimates  $\text{Mult}_1(t, m)$  and  $\text{Add}_1(t, m)$  for the first step of the truncated cyclotomic DFT computation algorithm.

*Lemma 4 (Upper bounds for computational complexity):* Let  $t$  be a degree of the error-locator polynomial over  $GF(2^m), n = 2^m - 1, t \leq (n - 1)/2$ , then the upper bounds for the computational complexity of the truncated cyclotomic DFT computation algorithm are

$$\begin{aligned} \text{Mult}_1(t, m) &\leq \begin{cases} \frac{t}{4} \text{Mult}(m) + \frac{t}{4}(m - 1), & \text{if } 4 \mid t, \\ \frac{t+1}{4} \text{Mult}(m) + \frac{t+1}{4}(m - 1), & \text{if } 4 \mid (t + 1), \\ \frac{t+2}{4} \text{Mult}(m) + \frac{t-2}{4}(m - 1), & \text{if } 4 \mid (t + 2), \\ \frac{t-1}{4} \text{Mult}(m) + \frac{t+3}{4}(m - 1), & \text{if } 4 \mid (t + 3), \end{cases} \end{aligned}$$

and

$$\text{Add}_1(t, m) \leq \begin{cases} \frac{t}{4} \text{Add}(m), & \text{if } 4 \mid t, \\ \frac{t+1}{4} \text{Add}(m), & \text{if } 4 \mid (t + 1), \\ \frac{t+2}{4} \text{Add}(m), & \text{if } 4 \mid (t + 2), \\ \frac{t-1}{4} \text{Add}(m), & \text{if } 4 \mid (t + 3), \end{cases}$$

where  $\text{Mult}(m)$  and  $\text{Add}(m)$  are the number of multiplications and additions of the normalized cyclic convolution calculation of length  $m$  over  $GF(2^m)$ , respectively.

*Proof:* The cyclotomic cosets  $C_k$  and their generators  $c_k$  have been chosen according to Subsection V-B.

TABLE 6. The complexity of the truncated cyclotomic DFT computation over  $GF(2^4)$ .

$t$	Chien search		Affine decomposition method [17]			Truncated cyclotomic DFT algorithm [32]	Novel method	
	Mult	Add	Mult	Add	Exp	Mult	Mult	Add
1	15	15	31	42	30	3	3	16
2	30	30	31	42	30	5	4	20
3	45	45	31	42	30	8	7	28
4	60	60	31	42	30	8	7	30
5	75	75	62	84	30	9	8	40
6	90	90	62	84	30	11	9	44
7	105	105	62	84	30	14	12	56
8	120	120	62	84	30	14	12	58
9	135	135	62	84	30	14	12	62
10	150	150	93	126	30	14	12	64
11	165	165	93	126	30	16	13	68
12	180	180	93	126	30	16	13	68
13	195	195	93	126	30	16	13	72
14	210	210	93	126	30	16	13	72

The union  $\mathcal{C}$  of the cyclotomic cosets has the form  $\mathcal{C} = C_0 \cup C_1 \cup C_3 \cup C_5 \cup \dots \cup C_t$ , that is, all indices of the cyclotomic cosets  $C_k \in \mathcal{C}$  are odd (excepting the trivial cyclotomic coset  $C_0 = \{c_0\} = \{0\}$ ). Note that some of the cyclotomic cosets may not be on this list because they have already been included into the set  $\mathcal{C}$ . To calculate the roots of the error-locator polynomial of degree  $t$ , we divide the set  $\mathcal{C}$  into three parts. If index of the cyclotomic coset does not exceed  $t/2$ , then such cyclotomic coset contains two or more components, which are necessary for calculating the truncated normalized cyclic convolution. Obviously, the computational complexity of such truncated normalized cyclic convolution does not exceed the computational complexity of the normalized cyclic convolution. If index of the cyclotomic coset is more than  $t/2$ , then almost all such cyclotomic cosets contain only one component. Sometimes in these cyclotomic cosets the number of components can be greater than one, if some cyclotomic cosets with indices greater than  $t/2$  are absent in the set  $\mathcal{C}$ , but this does not affect the computational complexity (if  $d$  cyclotomic cosets are absent, then no more than  $d$  components are added into other cyclotomic cosets).

Division of the set  $\mathcal{C}$  into three parts (the trivial cyclotomic coset; the cyclotomic cosets containing several components; the cyclotomic cosets containing one component):

- 1) If  $4 \mid t$  then  
 $C_0; C_1, C_3, \dots, C_{t/2-1}; C_{t/2+1}, \dots, C_{t-1};$
- 2) If  $4 \mid (t + 1)$  then  
 $C_0; C_1, C_3, \dots, C_{(t-1)/2}; C_{(t+3)/2}, \dots, C_t;$
- 3) If  $4 \mid (t + 2)$  then  
 $C_0; C_1, C_3, \dots, C_{t/2}; C_{t/2+2}, \dots, C_{t-1};$

- 4) If  $4 \mid (t + 3)$  then  
 $C_0; C_1, C_3, \dots, C_{(t-3)/2}; C_{(t+1)/2}, \dots, C_t.$

This completes the proof of Lemma 4. ■

Note that the additive complexity of the first step of the algorithm is considerably less than the additive complexity of the second step.

**C. COMPLEXITY ANALYSIS**

The computational complexity depends on two parameters: the DFT length  $n = 2^m - 1$  and the degree of the error-locator polynomial  $t$ . We assume that  $t < n/2$ .

The complexity of the first step of the algorithm for the truncated cyclotomic DFT calculation, using the complexity of the normalized cyclic convolutions computation (Subsubsection VI-A1) and upper bounds for computational complexity (Lemma 4), can be estimated as  $Mult_1(t, m) = \frac{t}{4}Mult(m) + \frac{t}{4}m = \frac{t}{4}(\frac{1}{2}m \log_2 m + m) = O(\frac{1}{8}t \log_2 n \log_2 \log_2 n)$  and  $Add_1(t, m) = \frac{t}{4}Add(m) = \frac{t}{4}m \log_2 m = O(\frac{1}{4}t \log_2 n \log_2 \log_2 n)$ , if  $m$  is a power of two.

For  $n = 15$  and  $n = 255$  the estimate for the number of multiplications is  $2t$  and  $5t$ , respectively. From comparison with Table 6 and Table 7 it follows that the last estimate is quite accurate.

If  $t > n/2$ , then the multiplicative complexity is close to the multiplicative complexity of the full DFT calculation  $O(n(\log_2 n)^{\log_2(3/2)})$  [23].

**D. ADDITIVE COMPLEXITY**

The second step of the truncated cyclotomic DFT calculation algorithm consists in multiplying the binary matrix by the vector over  $GF(2^m)$  (see (9)).

**TABLE 7.** The complexity of the truncated cyclotomic DFT computation over  $GF(2^8)$ .

$t$	Chien search		Affine decomposition method [17]			Truncated cyclotomic DFT algorithm [32]	Novel method	
	Mult	Add	Mult	Add	Exp	Mult	Mult	Add
1	255	255	287	534	510	7	7	255
2	510	510	287	534	510	14	10	255
3	765	765	287	534	510	21	17	559
4	1020	1020	287	534	510	26	18	563
5	1275	1275	574	1068	510	33	25	858
6	1530	1530	574	1068	510	40	28	866
7	1785	1785	574	1068	510	47	35	1263
8	2040	2040	574	1068	510	47	36	1267
9	2295	2295	574	1068	510	54	43	1612
10	2550	2550	861	1602	510	61	46	1620
11	2805	2805	861	1602	510	68	53	1961
12	3060	3060	861	1602	510	73	54	1965
13	3315	3315	861	1602	510	80	61	2080
14	3570	3570	861	1602	510	87	64	2088
15	3825	3825	1148	2136	510	94	71	2242
16	4080	4080	1148	2136	510	94	71	2250
17	4335	4335	1148	2136	510	97	74	2276
24	6120	6120	1435	2670	510	137	103	3119
32	8160	8160	2009	3738	510	184	138	4289

There are several methods for multiplying the binary matrix by the vector. The best known method is the modification of “four Russians” algorithm (V. L. Arlazarov, E. A. Dinits, M. A. Kronrod, and I. A. Faradzhev) [1, Algorithm 6.2] for multiplication of Boolean matrices, with complexity less than  $2n^2/\log_2 n$  additions over  $GF(2^m)$ . Other heuristic methods were reported in [3], [4], [10], [11], [33], [34], [40], [41].

#### E. COMPUTATIONAL COMPLEXITY COMPARISON FOR SEVERAL METHODS OF FINDING ROOTS OF POLYNOMIALS

The complexity of the truncated cyclotomic DFT computation over  $GF(2^4)$  and  $GF(2^8)$  is shown in Table 6 and Table 7, respectively. The methods are classical Chien search [13], the method of affine decomposition (the Fedorenko–Trifonov method) [17], direct truncation of the cyclotomic DFT algorithm [32] (unpublished), and the novel method.

The methods for comparison were chosen for the following reasons: the Chien search [13] seems to be the most popular method, the method of affine decomposition (the Fedorenko–Trifonov method [17]) is the best among the published, direct truncation of the cyclotomic DFT algorithm [32] (unpublished) seems to be the best method known to the author.

The number  $t$  is called the degree of the error-locator polynomial. The computational complexity of methods is indicated in terms of number of multiplications (Mult), additions (Add), and exponentiations (Exp) in the finite field  $GF(2^m)$ . Recall that the exponentiation means calculation  $(\alpha_j)^3$  and  $(\alpha_j)^5$  for all nonzero elements of the finite field

$GF(2^m)$ . For multiplying the binary matrix by the vector over  $GF(2^m)$  in the second step of the novel method the heuristic algorithm [33] was used. The number of additions for direct truncation of the cyclotomic DFT algorithm and for the novel method almost coincides.

#### VII. CONCLUSION

A novel method for finding roots of polynomials interconnected with the method of affine decomposition and the cyclotomic DFT algorithm was considered. Because of adapting the truncated cyclotomic DFT and the truncated normalized cyclic convolutions the novel method has the smallest multiplicative complexity. For this method according to Table 6 and Table 7 the multiplicative complexity is up to ten times less than with the method of affine decomposition (the Fedorenko–Trifonov method) [17], although the additive complexity is approximately 10–15% more. Moreover, the number of multiplications of the novel method is up to 23–33% less at approximately the same the number of additions, compared with the direct truncation of the cyclotomic DFT algorithm [32] (unpublished). According to the Tables, the full computational complexity of the novel method is less than that of other methods. Finally, for small values of degree  $t$  the novel method has not only the smallest multiplicative complexity, but the full computational complexity of this method is also less than with other methods.

#### ACKNOWLEDGMENT

The author would like to thank Peter Trifonov for his help in the optimization of matrix calculations.

## REFERENCES

- [1] A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA, USA: Addison-Wesley, 1976.
- [2] M. Alekhovich, "Linear diophantine equations over polynomials and soft decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2257–2265, Jul. 2005.
- [3] S. Bellini, M. Ferrari, and A. Tomasoni, "On the structure of cyclotomic Fourier transforms and their applications to Reed–Solomon codes," *IEEE Trans. Commun.*, vol. 59, no. 8, pp. 2110–2118, Aug. 2011.
- [4] S. Bellini, M. Ferrari, and A. Tomasoni, "On the reduction of additive complexity of cyclotomic FFTs," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1465–1468, Jun. 2012.
- [5] E. R. Berlekamp, *Algebraic Coding Theory*. New York, NY, USA: McGraw-Hill, 1968.
- [6] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA, USA: Addison-Wesley, 1983.
- [7] Q. Cai, T.-C. Lin, Y. Wu, W. Yu, and T.-K. Truong, "General method for prime-point cyclic convolution over the real field," 2019, *arXiv:1905.03398*. [Online]. Available: <http://arxiv.org/abs/1905.03398>
- [8] C.-L. Chen, "Formulas for the solutions of quadratic equations over  $GF(2^m)$ ," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 5, pp. 792–794, Sep. 1982.
- [9] N. Chen and Z. Yan, "Complexity analysis of Reed–Solomon decoding over  $GF(2^m)$  without using syndromes," *EURASIP J. Wireless Commun. Netw.*, vol. 2008, no. 1, Dec. 2008, Art. no. 843634.
- [10] N. Chen and Z. Yan, "Cyclotomic FFTs with reduced additive complexities based on a novel common subexpression elimination algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1010–1020, Mar. 2009.
- [11] N. Chen and Z. Yan, "Reduced-complexity Reed–Solomon decoders based on cyclotomic FFTs," *IEEE Signal Process. Lett.*, vol. 16, no. 4, pp. 279–282, Apr. 2009.
- [12] Y.-H. Chen, C.-W. Chen, J. Chang, T.-K. Truong, and G.-H. Liaw, "A fast algorithm for polynomial evaluation in Reed–Solomon codec," *J. Chin. Inst. Eng.*, vol. 38, no. 6, pp. 770–779, Aug. 2015.
- [13] R. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 4, pp. 357–363, Oct. 1964.
- [14] R. Chien, B. Cunningham, and I. Oldham, "Hybrid methods for finding roots of a polynomial—With application to BCH decoding (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 2, pp. 329–335, Mar. 1969.
- [15] E. Costa, S. Fedorenko, E. Krouk, M. Lott, E. Schulz, and P. Trifonov, "Method and device for a communication system for finding roots of an error locator polynomial," European Patent 1 367 727 A1, Mar. 12, 2003.
- [16] M. Elia, J. Rosenthal, and D. Schipani, "Polynomial evaluation over finite fields: New algorithms and complexity bounds," *Applicable Algebra Eng. Commun. Comput.*, vol. 23, nos. 3–4, pp. 129–141, Nov. 2012.
- [17] S. V. Fedorenko and P. V. Trifonov, "Finding roots of polynomials over finite fields," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1709–1711, Nov. 2002.
- [18] S. Fedorenko, P. Trifonov, and E. Costa, "Improved hybrid algorithm for finding roots of error-locator polynomials," *Eur. Trans. Telecommun.*, vol. 14, no. 5, pp. 411–416, 2003.
- [19] S. V. Fedorenko, "A method for computation of the discrete Fourier transform over a finite field," *Problems Inf. Transmiss.*, vol. 42, no. 2, pp. 139–151, Jun. 2006.
- [20] S. V. Fedorenko, "The discrete Fourier transform over a finite field with reduced multiplicative complexity," in *Proc. IEEE Int. Symp. Inf. Theory Proc.*, Saint-Petersburg, Russia, Jul. 2011, pp. 1200–1204.
- [21] S. V. Fedorenko, "Normalized cyclic convolution: The case of even length," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5307–5317, Oct. 2015.
- [22] S. V. Fedorenko, "Improving the Goertzel–Blahut algorithm," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 824–827, Jun. 2016.
- [23] S. V. Fedorenko, "Duhamel/Hollmann-like discrete Fourier transform algorithm with the smallest multiplicative complexity over a finite field," *IEEE Trans. Signal Process.*, vol. 68, pp. 4813–4823, Aug. 2020.
- [24] S. Gao and T. Mateer, "Additive fast Fourier transforms over finite fields," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6265–6272, Dec. 2010.
- [25] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [26] R. Lidl and H. Niederreiter, *Finite Fields*. Reading, MA, USA: Addison-Wesley, 1983.
- [27] S.-J. Lin, T. Y. Al-Naffouri, and Y. S. Han, "FFT algorithm for binary extension finite fields and its application to Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5343–5358, Oct. 2016.
- [28] S.-J. Lin, T. Y. Al-Naffouri, Y. S. Han, and W.-H. Chung, "Novel polynomial basis with fast Fourier transform and its application to Reed–Solomon erasure codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6284–6299, Nov. 2016.
- [29] T.-C. Lin, T. K. Truong, and P. D. Chen, "A fast algorithm for the syndrome calculation in algebraic decoding of Reed–Solomon codes," *IEEE Trans. Commun.*, vol. 55, no. 12, pp. 2240–2244, Dec. 2007.
- [30] T.-C. Lin, P.-D. Chen, and T.-K. Truong, "Simplified procedure for decoding nonsystematic Reed–Solomon codes over  $GF(2^m)$  using euclid's algorithm and the fast Fourier transform," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1588–1592, Jun. 2009.
- [31] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed–Solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, Jan. 2000.
- [32] P. V. Trifonov and S. V. Fedorenko, "A method for fast computation of the Fourier transform over a finite field," *Problems Inf. Transmiss.*, vol. 39, no. 3, pp. 231–238, 2003.
- [33] P. Trifonov, "Matrix-vector multiplication via erasure decoding," in *Proc. 11th Int. Symp. Problems Redundancy Inf. Control Syst.*, St.Petersburg, Russia, Jul. 2007, pp. 104–108. [Online]. Available: <http://dcm.fik.spbstu.ru/~petert/papers/mo.pdf>
- [34] P. Trifonov, "On the additive complexity of the cyclotomic FFT algorithm," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Lausanne, Switzerland, Sep. 2012, pp. 537–541.
- [35] T.-K. Truong, J.-H. Jeng, and I. S. Reed, "Fast algorithm for computing the roots of error locator polynomials up to degree 11 in Reed–Solomon decoders," *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 779–783, May 2001.
- [36] T. Truong, P. Chen, L. Wang, Y. Chang, and I. Reed, "Fast, prime factor, discrete Fourier transform algorithms over  $(2)$  for  $8 \leq m \leq 10$ ," *Inf. Sci.*, vol. 176, no. 1, pp. 1–26, Jan. 2006.
- [37] T.-K. Truong, P. D. Chen, L. J. Wang, and T. C. Cheng, "Fast transform for decoding both errors and erasures of Reed–Solomon codes over  $GF(2^m)$  for  $8 \leq m \leq 10$ ," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 181–186, Feb. 2006.
- [38] B. L. van der Waerden, *Algebra*, vol. 1. New York, NY, USA: Springer-Verlag, 1991.
- [39] C. J. Williamson, "Apparatus and method for error correction," U.S. Patent 5905 740, May 18, 1999.
- [40] X. Wu, M. Wagh, N. Chen, Y. Wang, and Z. Yan, "Composite cyclotomic Fourier transforms with reduced complexities," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2136–2145, May 2011.
- [41] X. Wu, Y. Wang, and Z. Yan, "On algorithms and complexities of cyclotomic fast Fourier transforms over arbitrary finite fields," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1149–1158, Mar. 2012.
- [42] X. Wu, Z. Yan, and J. Lin, "Reduced-complexity decoders of long Reed–Solomon codes based on composite cyclotomic Fourier transforms," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3920–3925, Jul. 2012.



**SERGEI VALENTINOVICH FEDORENKO** was born in Saint Petersburg, Russia, in 1967. He received the Ph.D. degree in computer science and the D.Sc.Tech. degree from the Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, in 1994 and 2009, respectively.

He has been the Alexander von Humboldt Foundation Research Fellow with the Technische Universität Darmstadt, Darmstadt, Germany, since 1999. He is currently a Leading Research Fellow/Professor with the Department of Informatics, National Research University Higher School of Economics (HSE University), HSE Campus in Saint Petersburg, Russia. His research interests include error-correcting codes, decoding algorithms, discrete Fourier transform over finite fields, and fast algorithms.

Prof. Fedorenko was the Organizer in 2008, the Vice Chair from 2008 to 2009, and the Chair from 2010 to 2012 of the IEEE Russia, Russia (Siberia), and Russia (Northwest) Joint Sections Information Theory Society Chapter.

• • •