Two Fast Erasure Decoding Algorithms for Reed–Solomon Codes Based on LCH-FFT

Chao Chen, Sian-Jheng Lin, Nianqi Tang, Yunghsiang S. Han, Suihua Cai, Leilei Yu, Zhongwei Li, Baoming Bai, and Bo Bai

***Abstract*—Based on a recently proposed fast Fourier transform by Lin, Chung, and Han, this paper presents two fast erasure decoding algorithms for Reed–Solomon (RS) codes over binary extension fields of length N and dimension K. The first algorithm applies to low-rate RS codes (i.e., $\frac{K}{N} \leq 0.5$) and achieves a complexity of $O(N \log K)$. The second algorithm applies to high-rate RS codes (i.e., $\frac{K}{N} \geq 0.5$) and achieves a complexity of $O(N \log(N-K))$. Compared to recent state-of-the-art algorithms, both proposed algorithms achieve the best complexity, resulting in significant throughput improvements in Single Instruction Multiple Data (SIMD) based simulations. Besides yielding new fast algorithms for RS codes, this paper also presents a new interpolation formula, as well as related results, which may be of independent interest.**

***Index Terms*—Reed–Solomon codes, fast erasure decoding, Lin–Chung–Han FFT, complexity, SIMD.**

I. Introduction

REED–SOLOMON (RS) codes [1] form an optimal class of erasure codes, which have been widely used in various communication and storage systems. By "optimal", it is meant that for an RS code of length N and dimension K, any K out of the N coded symbols can be used to recover the K data symbols. The standard encoding of RS codes is based on the generator matrix in Vandermonde or Cauchy form [2], [3], [4]. Decoding is generally performed in two steps: a preliminary step that depends only on the erasure positions, and a main step that utilizes the values of the received symbols. In many applications, such as packet-based networks and disk storage, codewords are transmitted in groups, and each group consists of a large number of codewords that share the same set of erasure positions. Therefore, the preliminary step can be performed once per group, and the main step practically dominates the decoding complexity.

This work was supported by the China National Key R&D Program under Grant 2021YFA1000500. Part of this work was presented at the 2023 IEEE International Symposium on Information Theory [42]. (*Corresponding author: Chao Chen.*)

Chao Chen and Baoming Bai are with the State Key Lab of ISN, Xidian University, Xi'an, China (e-mail: cchen@xidian.edu.cn, bmbai@mail.xidian.edu.cn).

Sian-Jheng Lin is with School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China (e-mail: sjlin@ustc.edu.cn)

Nianqi Tang, Yunghsiang S. Han, and Leilei Yu are with Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China (e-mail: 724973040@qq.com, yunghsiangh@gmail.com, yuleilei@ustc.edu).

Suihua Cai is with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China (e-mail: caish23@mail.sysu.edu.cn).

Zhongwei Li is with School of Software, Dalian University of Technology, Dalian, China (e-mail: zhongweili@dlut.edu.cn).

Bo Bai is with the Theory Laboratory, Central Research Institute, 2012 Labs, Huawei Technologies Company Ltd., Hong Kong, SAR (e-mail: baibo8@huawei.com).

In the 1970s, Pollard [5] first studied fast Fourier transform (FFT) over finite fields and Justesen [6] subsequently exploited the FFT for designing fast decoding algorithms for RS codes. A typical example is the Cooley–Tukey-like FFT over Fermat fields (i.e., prime fields of the form $\mathbb{F}_{2^r+1}$), which is commonly referred to as the Fermat Number Transform (FNT). There has been extensive research on designing FNT-based algorithms for RS codes, refer to [7], [8], [9], [10], [13] for error decoding and [11], [12], [13], [14], [15], [16] for erasure decoding. A major drawback of RS codes over $\mathbb{F}_{2^r+1}$ is that storing a (2^r+1)-ary number needs $(r+1)$ bits, which is wasteful and ineffective. By contrast, RS codes over binary extension fields are more favorable. For a long time, only semi-fast Fourier transform algorithms were developed for binary extension fields [17], [18], [19], [20]. A semi-fast algorithm significantly reduces the number of multiplications compared with the natural form of computation, but does not reduce (and may even increase) the number of additions [17]. A typical semi-fast algorithm is called the cyclotomic FFT (CFFT) [19], [20], [21]. In [22] and [23], the CFFT-based RS error decoding was investigated. So far, however, there has been little discussion on applying semi-fast Fourier transform algorithms to the erasure decoding of RS codes.

Recently, Lin, Chung, and Han [29] presented a new FFT (LCH-FFT, for short) over binary extension fields based on Wang and Zhu [25], Cantor [26], and Gao and Mateer [27]. For the first time, the $O(2^n \log 2^n)$ complexity is achieved for a 2^n-point FFT over binary extension fields. Like the Cooley–Tukey FFT over complex fields, the LCH-FFT can be represented by a butterfly diagram (refer to [29, Figure 1]), which effectively illustrates the flow of data through the algorithm.

The LCH-FFT has been effectively applied to RS codes for the design of fast algorithms. For low-rate RS codes (i.e., rate $\frac{K}{N} \leq 0.5$) with K being a power of 2, a systematic encoding algorithm with complexity $O(N \log K)$ was presented in [30]. For high-rate RS codes (i.e., rate $\frac{K}{N} \geq 0.5$) with $N-K$ being a power of 2, a systematic encoding algorithm with complexity $O(N \log(N-K))$ was presented in [31]. An erasure decoding algorithm with complexity $O(N \log N)$ was presented in [30], which applies to both low-rate and high-rate RS codes. An error decoding algorithm for high-rate RS codes was presented in [31], which achieves a complexity of $O(N \log(N-K) + (N-K) \log^2(N-K))$. In [33] and [34], Welch–Berlekamp-type algorithms were designed to solve the key equation derived in [31]. In [32], the "partial FFT" algorithm was presented to adapt to general code parameters.

Besides the LCH-FFT-based algorithm [30], other state-

of-the-art algorithms for erasure decoding of RS codes over binary extension fields include the Didier's algorithm [35] and the Reed–Muller-transform (RMT)-based algorithm [36]. Among these algorithms, the LCH-FFT-based algorithm [30] achieves the best previously known complexity, reaching $O(N \log N)$. Refer to Section VI for a detailed comparison of complexity.

Focusing on RS codes over binary extension fields, this paper is devoted to developing new fast erasure decoding algorithms based on LCH-FFT. By exploring new properties of LCH-FFT, we design even faster algorithms with reduced complexity. The main contributions of the paper can be summarized as follows.

- For low-rate RS codes (i.e., $\frac{K}{N} \leq 0.5$), we present a fast erasure decoding algorithm, which achieves a complexity of $O(N \log K)$. By improving on the existing LCH-FFT-based algorithm [30], the new algorithm achieves the best complexity for low-rate RS codes.
- For high-rate RS codes (i.e., $\frac{K}{N} \geq 0.5$), we present a fast erasure decoding algorithm, which achieves a complexity of $O(N \log(N-K))$. To achieve erasure decoding, a key equation, as well as a Forney-like formula, is derived. The key to low complexity is to incorporate the LCH-FFT into the computation. The resulting algorithm achieves the best complexity for high-rate RS codes.
- As the essential foundation for the proposed algorithms, a new interpolation formula is presented. Two corollaries are derived: one is essential for building the first algorithm, and the other provides a new proof of an important property of LCH-FFT, which is crucial for constructing the second algorithm. Moreover, these results may also be of independent interest.
- To demonstrate the superiority of the proposed algorithms, we compare them with recent state-of-the-art algorithms in terms of complexity. The proposed algorithms are shown to attain the best complexity. To test real performance, we further perform simulations based on the Single Instruction Multiple Data (SIMD) technique. Simulation results show that the proposed algorithms achieve significant throughput improvements compared to the state-of-the-art algorithms.

The rest of the paper is organized as follows. Section II provides the preliminaries. Section III presents a new interpolation formula as well as related results. Section IV derives a fast erasure decoding algorithm for low-rate RS codes. Section V derives a fast erasure decoding algorithm for high-rate RS codes. Section VI compares the algorithm complexity and presents the simulation results. Section VII concludes the paper.

II. Preliminaries

A. LCH-FFT

Let $\mathbb{F}_{2^m}$ denote a binary extension field, which can be seen as an m-dimensional vector space over $\mathbb{F}_2$. Let $\{v_i\}_{i=0}^{m-1}$ denote a basis of the vector space. For an integer $0 \leq i \leq 2^m-1$, the binary expansion of i is $i = i_0 + i_1 2 + i_2 2^2 + \cdots + i_{m-1}2^{m-1}$, where $i_j = 0$ or 1 for $0 \leq j \leq m-1$. Let $\{\omega_i\}_{i=0}^{2^m-1}$ be the elements of $\mathbb{F}_{2^m}$, specified as

$$\omega_i = i_0 v_0 + i_1 v_1 + \cdots + i_{m-1} v_{m-1}. \tag{1}$$

Note that $\omega_0 = 0$ is the additive identity of $\mathbb{F}_{2^m}$. Let V_n be an n-dimensional subspace, $0 \leq n \leq m$, given by

$$\begin{aligned} V_n &= \{i_0 v_0 + i_1 v_1 + \cdots + i_{n-1} v_{n-1} : i_j \in \mathbb{F}_2\} \\ &= \{\omega_0, \omega_1, \ldots, \omega_{2^n-1}\}. \end{aligned} \tag{2}$$

The (vanishing) subspace polynomial corresponding to V_n is defined as [28]

$$s_n(x) = \prod_{a \in V_n} (x - a). \tag{3}$$

Clearly, the degree of $s_n(x)$ is equal to $\deg(s_n(x)) = 2^n$. The subspace polynomial $s_n(x)$ is a linearized polynomial such that $s_n(a+b) = s_n(a) + s_n(b)$ for any $a, b \in \mathbb{F}_{2^m}$. According to [30, Lemma 5], the formal derivative of $s_n(x)$ is a constant, given by

$$s_n'(x) = \prod_{a \in V_n \setminus \{0\}} a. \tag{4}$$

Let $\mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ denote the ring of polynomials over $\mathbb{F}_{2^m}$ modulo $x^{2^m} - x$. In [29], Lin, Chung, and Han introduced a new polynomial basis for $\mathbb{F}_{2^m}[x]/(x^{2^m} - x)$, now commonly referred to as the LCH basis in the literature. It is denoted as

$$\bar{\mathbb{X}} = \left\{ \bar{X}_0(x), \bar{X}_1(x), \ldots, \bar{X}_{2^m-1}(x) \right\}. \tag{5}$$

Recall that $(i_0, i_1, \ldots, i_{m-1})$ is the binary representation of i, $0 \leq i \leq 2^m - 1$. Each $\bar{X}_i(x)$ of $\bar{\mathbb{X}}$ is defined as

$$\bar{X}_i(x) = \frac{X_i(x)}{p_i}, \tag{6}$$

where

$$X_i(x) = \prod_{j=0}^{m-1} (s_j(x))^{i_j}, \tag{7}$$

and

$$p_i = \prod_{j=0}^{m-1} (s_j(v_j))^{i_j}. \tag{8}$$

The degree of $\bar{X}_i(x)$ is equal to $\deg(\bar{X}_i(x)) = \deg(X_i(x)) = \sum_{j=0}^{m-1} i_j \deg(s_j(x)) = \sum_{j=0}^{m-1} i_j 2^j = i$. In the LCH basis, a polynomial $f(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ can be represented as

$$f(x) = \sum_{i=0}^{2^m-1} f_i \bar{X}_i(x). \tag{9}$$

The vector $\mathbf{f} = (f_0, f_1, \ldots, f_{2^m-1})$ is the coefficient vector of $f(x)$ with respect to the LCH basis. When it is clear from the context that the LCH basis is being considered, we may interchangeably use $f(x)$ and $\mathbf{f}$.

Algorithm 1: $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, n, \beta)$ (or $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(f(x), n, \beta)$) [31]

Input: $\mathbf{f} = (f_0, f_1, \ldots, f_{2^n-1})$ as the coefficient vector of $f(x) = \sum_{i=0}^{2^n-1} f_i \bar{X}_i(x)$
Output: $\mathbf{F} = (f(\omega_0 + \beta), f(\omega_1 + \beta), \ldots, f(\omega_{2^n-1} + \beta))$
if $n = 0$ **then**
  return f_0
end if
for $i = 0, 1, \ldots, 2^{n-1} - 1$ **do**
  $a_i^{(0)} = f_i + \frac{s_{n-1}(\beta)}{s_{n-1}(v_{n-1})} f_{i+2^{n-1}}$
  $a_i^{(1)} = a_i^{(0)} + f_{i+2^{n-1}}$
end for
$\mathbf{A}_0 = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{a}^{(0)}, n-1, \beta)$, where $\mathbf{a}^{(0)} = (a_0^{(0)}, a_1^{(0)}, \ldots, a_{2^{n-1}-1}^{(0)})$
$\mathbf{A}_1 = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{a}^{(1)}, n-1, v_{n-1} + \beta)$, where $\mathbf{a}^{(1)} = (a_0^{(1)}, a_1^{(1)}, \ldots, a_{2^{n-1}-1}^{(1)})$
return $\mathbf{F} = (\mathbf{A}_0, \mathbf{A}_1)$

Algorithm 2: $\mathbf{f} = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}, n, \beta)$ (or $f(x) = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}, n, \beta)$) [31]

Input: $\mathbf{F} = (F_0, F_1, \ldots, F_{2^n-1})$
Output: $\mathbf{f}$ such that $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, n, \beta)$.
if $n = 0$ **then**
  return F_0
end if
$\mathbf{a}^{(0)} = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{A}_0, n-1, \beta)$, where $\mathbf{A}_0 = (F_0, F_1, \ldots, F_{2^{n-1}-1})$
$\mathbf{a}^{(1)} = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{A}_1, n-1, v_{n-1} + \beta)$, where $\mathbf{A}_1 = (F_{2^{n-1}}, F_{2^{n-1}+1}, \ldots, F_{2^n-1})$
for $i = 0, 1, \ldots, 2^{n-1} - 1$ **do**
  $f_{i+2^{n-1}} = a_i^{(0)} + a_i^{(1)}$
  $f_i = a_i^{(0)} + \frac{s_{n-1}(\beta)}{s_{n-1}(v_{n-1})} f_{i+2^{n-1}}$
end for
return $\mathbf{f} = (f_0, f_1, \ldots, f_{2^n-1})$

Similar to [31, Eq. (75)], it can be proved that for $0 \le k \le n \le m$,

$$s_n(x) = p_{2^n-2^k} \bar{X}_{2^n-2^k}(x) s_k(x) + \sum_{i=k}^{n-1} p_{2^n-2^i} \bar{X}_{2^n-2^i}(x) s_i(v_i). \tag{10}$$

Let $f(x) = \sum_{i=0}^{2^n-1} f_i \bar{X}_i(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ be a polynomial of degree less than 2^n. Let β be any fixed element of $\mathbb{F}_{2^m}$. Let $V_n + \beta = \{a + \beta : a \in V_n\}$ be the point set. The LCH-FFT is a fast algorithm for evaluating the polynomial $f(x)$ at the point set $V_n + \beta$. We denote the algorithm by

$$\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, n, \beta), \quad \text{or} \quad \mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(f(x), n, \beta), \tag{11}$$

which takes $\mathbf{f} = (f_0, f_1, \ldots, f_{2^n-1})$ as input and computes $\mathbf{F} = (f(\omega_0 + \beta), f(\omega_1 + \beta), \ldots, f(\omega_{2^n-1} + \beta))$ as output. Given a vector $\mathbf{F}$ over $\mathbb{F}_{2^m}$ of length 2^n, there is a unique vector $\mathbf{f}$ over $\mathbb{F}_{2^m}$ of length 2^n such that $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, n, \beta)$. The inverse FFT algorithm that performs the polynomial interpolation is denoted by

$$\mathbf{f} = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}, n, \beta), \quad \text{or} \quad f(x) = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}, n, \beta). \tag{12}$$

Algorithm 1 and Algorithm 2 present the descriptions of $\mathrm{FFT}_{\bar{\mathbb{X}}}$ and $\mathrm{IFFT}_{\bar{\mathbb{X}}}$, respectively. Let $N = 2^n$ denote the FFT length. Take $\mathrm{FFT}_{\bar{\mathbb{X}}}$ as an example. Let $A(N)$ be the number of additions and $M(N)$ the number of multiplications. The recursive structure of the algorithm implies that $A(N) = 2A(N/2) + N$ and $M(N) = 2M(N/2) + N/2$. Therefore, $A(N) = N \log N$ and $M(N) = \frac{1}{2} N \log N$. (In this paper, the base of the logarithm is assumed to be 2.) Similarly, $\mathrm{IFFT}_{\bar{\mathbb{X}}}$ requires the same number of operations over $\mathbb{F}_{2^m}$ as $\mathrm{FFT}_{\bar{\mathbb{X}}}$.

Let k be an integer, $0 \le k \le n$. Let a length-2^n vector $\mathbf{f} = (f_0, f_1, \ldots, f_{2^n-1})$ be denoted in segmented form as

$$\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{2^{n-k}-1}), \tag{13}$$

where each sub-vector $\mathbf{f}_i = (f_{i2^k}, f_{i2^k+1}, \ldots, f_{i2^k+2^k-1})$, $0 \le i \le 2^{n-k} - 1$, is of length 2^k. Similarly, let the vector $\mathbf{F} = (f(\omega_0 + \beta), f(\omega_1 + \beta), \ldots, f(\omega_{2^n-1} + \beta))$ be denoted as

$$\mathbf{F} = (\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{2^{n-k}-1}), \tag{14}$$

where each sub-vector $\mathbf{F}_i = (f(\omega_{i2^k} + \beta), f(\omega_{i2^k+1} + \beta), \ldots, f(\omega_{i2^k+2^k-1} + \beta))$, $0 \le i \le 2^{n-k} - 1$, is of length 2^k.

The following lemma provides two important properties of LCH-FFT, which are essential for fast encoding and decoding of RS codes.

Lemma 1: Let $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, n, \beta)$, where $\mathbf{f}$ and $\mathbf{F}$ are segmented as in (13) and (14).

1) If $(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{2^{n-k}-1}) = (\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0})$, then [30]
$$\mathbf{F}_i = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}_0, k, \omega_{i2^k} + \beta), \quad 0 \le i \le 2^{n-k} - 1. \tag{15}$$
2) The following equality holds [31]:
$$\mathbf{f}_{2^{n-k}-1} = \sum_{i=0}^{2^{n-k}-1} \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}_i, k, \omega_{i2^k} + \beta). \tag{16}$$

B. RS Encoding based on LCH-FFT

Let $\mathrm{RS}(N, K)$ denote an RS code over $\mathbb{F}_{2^m}$ of length N and dimension K. A low-rate RS code refers to an $\mathrm{RS}(N, K)$ code with $\frac{K}{N} \le 0.5$, and a high-rate RS code refers to an $\mathrm{RS}(N, K)$ code with $\frac{K}{N} \ge 0.5$.

In this paper, we consider full-length RS codes with $N = 2^m$. An $\mathrm{RS}(2^m, K)$ code $\mathcal{C}$ can be defined as

$$\mathcal{C} = \Big\{ \mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(f(x), m, \omega_0) : f(x) = \sum_{i=0}^{2^m-1} f_i \bar{X}_i(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x) \text{ such that } \deg(f(x)) < K \Big\}. \tag{17}$$

For a low-rate RS code, we assume that $N = 2^m$ and $K = 2^k$ for some $0 \le k < m$. The systematic encoding can be performed as follows [30]. Let $\mathbf{f} = (f_0, f_1, \ldots, f_{2^m-1})$ and $\mathbf{F} =$

$\mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, m, \omega_0) = (f(\omega_0), f(\omega_1), \ldots, f(\omega_{2^m-1}))$ be denoted as $\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{2^{m-k}-1})$ and $\mathbf{F} = (\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{2^{m-k}-1})$, where $\mathbf{f}_i = (f_{i2^k}, f_{i2^k+1}, \ldots, f_{i2^k+2^k-1})$ and $\mathbf{F}_i = (f(\omega_{i2^k}), f(\omega_{i2^k+1}), \ldots, f(\omega_{i2^k+2^k-1}))$, $0 \le i \le 2^{m-k}-1$. Since $\deg(f(x)) < K = 2^k$, we have $(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{2^{m-k}-1}) = (\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0})$. Let $\mathbf{F}_0$ be the given message vector and let $(\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{2^{m-k}-1})$ be the parity vector to be computed. According to Property 1 of Lemma 1, it follows from (15) that

$$\mathbf{f}_0 = \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}_0, k, \omega_0). \tag{18}$$

Using (15) again, the parity vector can be computed section by section, given by

$$\mathbf{F}_i = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}_0, k, \omega_{i2^k}), \quad 1 \le i \le 2^{m-k}-1. \tag{19}$$

The systematic encoding requires $2^k \log 2^k + (2^{m-k}-1) \times 2^k \log 2^k = N \log K$ additions and $\frac{1}{2}2^k \log 2^k + (2^{m-k}-1) \times \frac{1}{2}2^k \log 2^k = \frac{1}{2}N \log K$ multiplications.

For a high-rate RS code, we assume that $N = 2^m$ and $N - K = 2^t$ for some $0 \le t < m$. The systematic encoding can be performed as follows [31]. Let $\mathbf{f} = (f_0, f_1, \ldots, f_{2^m-1})$ and $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(\mathbf{f}, m, \omega_0) = (f(\omega_0), f(\omega_1), \ldots, f(\omega_{2^m-1}))$ be denoted as $\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{2^{m-t}-1})$ and $\mathbf{F} = (\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{2^{m-t}-1})$, where $\mathbf{f}_i = (f_{i2^t}, f_{i2^t+1}, \ldots, f_{i2^t+2^t-1})$ and $\mathbf{F}_i = (f(\omega_{i2^t}), f(\omega_{i2^t+1}), \ldots, f(\omega_{i2^t+2^t-1}))$, $0 \le i \le 2^{m-t}-1$. Let $(\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{2^{m-t}-1})$ be the given message vector and let $\mathbf{F}_0$ be the parity vector to be computed. Since $\deg(f(x)) < K$, we have $\mathbf{f}_{2^{m-t}-1} = \mathbf{0}$. According to Property 2 of Lemma 1, we have

$$\mathbf{0} = \sum_{i=0}^{2^{m-t}-1} \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}_i, t, \omega_{i2^t}). \tag{20}$$

Applying $\mathrm{FFT}_{\bar{\mathbb{X}}}(\cdot, t, \omega_0)$ on (20), the parity vector can be computed by

$$\mathbf{F}_0 = \mathrm{FFT}_{\bar{\mathbb{X}}}\Big(\sum_{i=1}^{2^{m-t}-1} \mathrm{IFFT}_{\bar{\mathbb{X}}}(\mathbf{F}_i, t, \omega_{i2^t}), t, \omega_0 \Big). \tag{21}$$

The systematic encoding requires $(2^{m-t}-1) \times 2^t \log 2^t + (2^{m-t}-2) \times 2^t + 2^t \log 2^t = N \log(N-K) + (2K-N)$ additions and $(2^{m-t}-1) \times \frac{1}{2}2^t \log 2^t + \frac{1}{2}2^t \log 2^t = \frac{1}{2}N \log(N-K)$ multiplications.

As a concluding remark of this subsection, we would like to emphasize that the restriction due to the constraint on N and K is mild. In fact, many code parameters of interest are within the scope of our consideration. What is more, we may use techniques such as puncturing, shortening, and partial FFT [32] to adapt to general code parameters.

C. RS Erasure Decoding based on LCH-FFT

In [30], an erasure decoding algorithm based on LCH-FFT was proposed, which applies to both low-rate and high-rate RS codes. The derivation is as follows.

Let $\mathbf{F} = \mathrm{FFT}_{\bar{\mathbb{X}}}(f(x), m, \omega_0)$ be the transmitted codeword of length $N = 2^m$, where $f(x) = \sum_{i=0}^{2^m-1} f_i \bar{X}_i(x)$ with $\deg(f(x)) < K$. Since any K out of the 2^m received symbols are sufficient for erasure decoding, for ease of description, we assume that the received word has $2^m - K$ erasures. The erasure locator polynomial is

$$\Lambda(x) = \prod_{\omega \in \mathcal{E}} (x - \omega), \tag{22}$$

where $\mathcal{E} \subset \{\omega_0, \omega_1, \ldots, \omega_{2^m-1}\}$ is the set of erasure locators such that $|\mathcal{E}| = 2^m - K$. Define the polynomial

$$\hat{f}(x) = f(x)\Lambda(x). \tag{23}$$

Since $\deg(f(x)) < K$ and $\deg(\Lambda(x)) = 2^m - K$, we have $\deg(\hat{f}(x)) < 2^m$. The formal derivative of $\hat{f}(x)$ is

$$\hat{f}'(x) = f'(x)\Lambda(x) + f(x)\Lambda'(x). \tag{24}$$

Substituting $x = \omega \in \mathcal{E}$ into (24), since $\Lambda(\omega) = 0$, the erasures can be computed by

$$f(\omega) = \frac{\hat{f}'(\omega)}{\Lambda'(\omega)}, \quad \omega \in \mathcal{E}. \tag{25}$$

Invoking the LCH-FFT, the erasure decoding of RS codes can be performed as follows. First, compute

$$\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\} \quad \text{and} \quad \{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}. \tag{26}$$

Then, based on (23), compute $(\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1}))$ by

$$\hat{f}(\omega) = \begin{cases} 0, & \text{if } \omega \in \mathcal{E}, \\ f(\omega)\Lambda(\omega), & \text{if } \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}, \end{cases} \tag{27}$$

where $\{f(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ are the received symbols. Next, compute $\hat{f}(x) = \sum_{i=0}^{2^m-1} \hat{f}_i \bar{X}_i(x)$ by

$$\begin{aligned} &(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{2^m-1}) \\ &= \mathrm{IFFT}_{\bar{\mathbb{X}}}((\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1})), m, \omega_0). \end{aligned} \tag{28}$$

Then compute the formal derivative $\hat{f}'(x) = \sum_{i=0}^{2^m-1} \hat{f}'_i \bar{X}_i(x)$. Next, compute $(\hat{f}'(\omega_0), \hat{f}'(\omega_1), \ldots, \hat{f}'(\omega_{2^m-1}))$ by

$$\begin{aligned} &(\hat{f}'(\omega_0), \hat{f}'(\omega_1), \ldots, \hat{f}'(\omega_{2^m-1})) \\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((\hat{f}'_0, \hat{f}'_1, \ldots, \hat{f}'_{2^m-1}), m, \omega_0), \end{aligned} \tag{29}$$

Finally, compute the erasures by (25).

For clarity, Algorithm 3 describes the erasure decoding algorithm for RS codes [30]. The complexity is analyzed as follows. The preliminary step corresponds to Line 1, which can be achieved with complexity $O(N \log N)$ by using the fast Walsh–Hadamard transform [30]. The computation depends only on the erasure positions. The main step corresponds to Lines 2–6. Line 2 requires K multiplications. Line 3 requires $N \log N$ additions and $\frac{1}{2}N \log N$ multiplications. Line 4 computes the formal derivative of a polynomial of degree less than N in the LCH basis, which requires N multiplications and around $\frac{1}{2}N \log N$ additions [30]. Line 5 requires $N \log N$ additions and $\frac{1}{2}N \log N$ multiplications. Line 6 requires $(N - K)$ multiplications. In summary, the main step that dominates the decoding complexity achieves complexity $O(N \log N)$.

Algorithm 3: A Fast Erasure Decoding Algorithm for RS Codes [30]

Input: The erasure locator set $\mathcal{E}$ and the received symbols $\{f(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$

Output: The erasures $\{f(\omega) : \omega \in \mathcal{E}\}$.

1: Compute $\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ and $\{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}$.

2: Compute $(\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1}))$ by

$$\hat{f}(\omega) = \begin{cases} 0, & \text{if } \omega \in \mathcal{E}, \\ f(\omega)\Lambda(\omega), & \text{if } \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}. \end{cases}$$

3: Compute $\hat{f}(x) = \sum_{i=0}^{2^m-1} \hat{f}_i \bar{X}_i(x)$ by

$$(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{2^m-1}) = \mathrm{IFFT}_{\bar{\mathbb{X}}}((\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1})), m, \omega_0).$$

4: Compute the formal derivative $\hat{f}'(x) = \sum_{i=0}^{2^m-1} \hat{f}'_i \bar{X}_i(x)$.

5: Compute $(\hat{f}'(\omega_0), \hat{f}'(\omega_1), \ldots, \hat{f}'(\omega_{2^m-1}))$ by

$$(\hat{f}'(\omega_0), \hat{f}'(\omega_1), \ldots, \hat{f}'(\omega_{2^m-1})) = \mathrm{FFT}_{\bar{\mathbb{X}}}((\hat{f}'_0, \hat{f}'_1, \ldots, \hat{f}'_{2^m-1}), m, \omega_0).$$

6: Compute the erasures by

$$f(\omega) = (\Lambda'(\omega))^{-1} \hat{f}'(\omega), \quad \omega \in \mathcal{E}.$$

III. A New Interpolation Formula and Related Results

In this section, we present a new interpolation formula (Theorem 1) and two related results (Corollary 1 and Corollary 2). The significance is twofold. First, Corollary 1 is essential for building the fast algorithm for low-rate RS codes in Section IV. Second, Corollary 2 leads to a new proof of Property 2 of Lemma 1, which is essential for building the fast algorithm for high-rate RS codes in Section V. For clarity, proofs are largely deferred to the appendices.

Let $f(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ be a polynomial of degree less than 2^n. Let k be any fixed integer, $0 \le k \le n$. Define 2^{n-k} polynomials, denoted $f^{(i)}(x)$, $0 \le i \le 2^{n-k} - 1$, as follows. Let β be any fixed element of $\mathbb{F}_{2^m}$. Let $f^{(i)}(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ be a polynomial of degree less than 2^k specified by

$$f^{(i)}(\omega_{i2^k+j} + \beta) = f(\omega_{i2^k+j} + \beta), \quad 0 \le j \le 2^k - 1. \tag{30}$$

We have the following interpolation formula for representing $f(x)$.

Theorem 1:

$$f(x) = \frac{\prod_{a \in V_k \setminus \{0\}} a}{\prod_{a \in V_n \setminus \{0\}} a} \sum_{i=0}^{2^{n-k}-1} f^{(i)}(x) T^{(i)}(x), \tag{31}$$

where

$$T^{(i)}(x) = \frac{s_n(x) - s_n(\omega_{i2^k} + \beta)}{s_k(x) - s_k(\omega_{i2^k} + \beta)}. \tag{32}$$

If $k = 0$, then $f^{(i)}(x)$ is a constant, equal to $f(\omega_i + \beta)$, and (31) becomes

$$f(x) = \Big(\prod_{a \in V_n \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{s_n(x) - s_n(\omega_i + \beta)}{x - (\omega_i + \beta)}. \tag{33}$$

Proof: Please refer to Appendix A. ∎

We notice that the interpolation formula (33) can be interpreted as a special form of Lagrange interpolation. Specifically, let $x_i = \omega_i + \beta$ for $0 \le i \le 2^n - 1$. Then

$$\begin{aligned} f(x) &= \sum_{i=0}^{2^n-1} f(x_i) \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \\ &= \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{\prod_{j \neq i} (x - (\omega_j + \beta))}{\prod_{j \neq i} ((\omega_i + \beta) - (\omega_j + \beta))} \\ &= \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{\prod_{j \neq i} (x - (\omega_j + \beta))}{\prod_{j \neq i} (\omega_i - \omega_j)} \\ &= \Big(\prod_{a \in V_n \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{\prod_{j=0}^{2^n-1} (x - (\omega_j + \beta))}{x - (\omega_i + \beta)} \\ &= \Big(\prod_{a \in V_n \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{s_n(x) - s_n(\beta)}{x - (\omega_i + \beta)} \\ &= \Big(\prod_{a \in V_n \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{s_n(x) - s_n(\omega_i + \beta)}{x - (\omega_i + \beta)}, \end{aligned} \tag{34}$$

where the fourth equality is due to that $\prod_{j \neq i} (\omega_i - \omega_j) = \prod_{a \in V_n \setminus \{0\}} a$ for any $0 \le i \le 2^n - 1$, and the last equality is due to that $s_n(\omega_i) = 0$ for all $0 \le i \le 2^n - 1$. It is well known that Lagrange interpolation is a special case of the Chinese remainder theorem. Given that (33) is both a special case of (31) and a special form of Lagrange interpolation, there might be a certain relationship, which we do not know yet, between the proposed interpolation formula (31) and the Chinese remainder theorem.

Define the polynomial

$$g(x) = f'(x) \mod s_k(x - \beta), \tag{35}$$

where $f'(x)$ is the formal derivative of $f(x)$.

We have the following result for representing $g(x)$.

Corollary 1:

$$\begin{aligned} g(x) =& \big(f^{(0)}(x)\big)' + \Big(\sum_{a \in V_n \setminus V_k} \frac{1}{a} \Big) f^{(0)}(x) \\ &+ \sum_{i=1}^{2^{n-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} f^{(i)}(x). \end{aligned} \tag{36}$$

Proof: Please refer to Appendix B. ∎

Next, we consider dividing $f(x)$ by $\bar{X}_{2^n-2^k}(x)$, written

$$f(x) = h(x) \bar{X}_{2^n-2^k}(x) + r(x), \tag{37}$$

where $r(x) = 0$ or $\deg(r(x)) < 2^n - 2^k$.

We have the following result for representing $h(x)$.

Corollary 2:

$$h(x) = \Big(\prod_{a \in V_k \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^n-1} f(\omega_i + \beta) \frac{s_k(x) - s_k(\omega_i + \beta)}{x - (\omega_i + \beta)} \tag{38}$$

$$= \sum_{i=0}^{2^{n-k}-1} f^{(i)}(x). \tag{39}$$

If $f(x)$ is represented in the LCH basis as $f(x) = \sum_{i=0}^{2^n-1} f_i \bar{X}_i(x)$, then

$$h(x) = \sum_{i=0}^{2^k-1} f_{2^n - 2^k + i} \bar{X}_i(x). \tag{40}$$

Proof: Please refer to Appendix C. ∎

As a consequence of Corollary 2, we give a simple direct proof of Property 2 of Lemma 1, in contrast to the original proof by induction [31].

Proof of Property 2: It is seen from (40) that the coefficient vector of $h(x)$ with respect to the LCH basis is $\mathbf{f}_{2^{n-k}-1} = (f_{2^n-2^k}, f_{2^n-2^k+1}, \ldots, f_{2^n-1})$, the right-most subvector of $\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{2^{n-k}-1})$ in (13). Based on (30) and the definition of IFFT, the coefficient vector of $f^{(i)}(x)$ with respect to the LCH basis is $\mathrm{IFFT}_{\mathbb{X}}(\mathbf{F}_i, k, \omega_{i2^k} + \beta)$, where $\mathbf{F}_i = (f(\omega_{i2^k} + \beta), f(\omega_{i2^k+1} + \beta), \ldots, f(\omega_{i2^k+2^k-1} + \beta))$. Since $h(x) = \sum_{i=0}^{2^{n-k}-1} f^{(i)}(x)$ in (39), we have $\mathbf{f}_{2^{n-k}-1} = \sum_{i=0}^{2^{n-k}-1} \mathrm{IFFT}_{\mathbb{X}}(\mathbf{F}_i, k, \omega_{i2^k} + \beta)$, as is given in (16) as Property 2 of Lemma 1. ∎

IV. A Fast Erasure Decoding Algorithm for Low-Rate RS Codes

In this section, we present a reduced-complexity algorithm for low-rate RS codes. Note that, for a low-rate RS(N, K) code, K is much smaller than N. If we can reduce the decoding complexity of some steps of the decoding process from $O(N \log N)$ to $O(N \log K)$, the overall decoding complexity will be reduced. Based on Corollary 1, a length-N LCH-FFT of Algorithm 3 can be replaced by $\frac{N}{K}$ length-K LCH-FFTs, thus reducing some decoding steps' complexity from $O(N \log N)$ to $O(N \log K)$. The derivation is as follows.

In deriving Algorithm 3, we have defined a polynomial $\hat{f}(x) = f(x)\Lambda(x)$ such that $\deg(\hat{f}(x)) \le 2^m - 1$ (refer to (23)). We now apply Theorem 1 and Corollary 1 on this polynomial by taking $n = m$ and $\beta = 0$. For this purpose, let $\hat{f}^{(i)}(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$, $0 \le i \le 2^{m-k} - 1$, be 2^{m-k} polynomials of degree less than 2^k, specified by

$$\hat{f}^{(i)}(\omega_{i2^k+j}) = \hat{f}(\omega_{i2^k+j}), \quad 0 \le j \le 2^k - 1. \tag{41}$$

Let

$$g(x) = \hat{f}'(x) \mod s_k(x), \tag{42}$$

where $\hat{f}'(x)$ is the formal derivative of $\hat{f}(x)$. When $n = m$, we have $V_m = \mathbb{F}_{2^m}$, $\prod_{a \in \mathbb{F}_{2^m} \setminus \{0\}} a = 1$, $s_m(x) = x^{2^m} - x$, and $s_m(a) = 0$ for all $a \in \mathbb{F}_{2^m}$. Therefore, based on Theorem 1,

$$\hat{f}(x) = \Big(\prod_{a \in V_k \setminus \{0\}} a \Big) \sum_{i=0}^{2^{m-k}-1} \hat{f}^{(i)}(x) \frac{x^{2^m} - x}{s_k(x - \omega_{i2^k})}. \tag{43}$$

Based on Corollary 1,

$$g(x) = \big(\hat{f}^{(0)}(x)\big)' + \Big(\sum_{a \in \mathbb{F}_{2^m} \setminus V_k} \frac{1}{a} \Big) \hat{f}^{(0)}(x) + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}^{(i)}(x). \tag{44}$$

Substituting $x = \omega \in V_k$ into (43), we have

$$\begin{aligned} &\hat{f}(\omega) \\ &= \Big(\prod_{a \in V_k \setminus \{0\}} a \Big) \hat{f}^{(0)}(x) \frac{x^{2^m} - x}{s_k(x)} \Big|_{x=\omega} \\ &\quad + \Big(\prod_{a \in V_k \setminus \{0\}} a \Big) \sum_{i \ne 0} \hat{f}^{(i)}(x) \frac{x^{2^m} - x}{s_k(x - \omega_{i2^k})} \Big|_{x=\omega} \\ &= \Big(\prod_{a \in V_k \setminus \{0\}} a \Big) \frac{(\hat{f}^{(0)}(x))'(x^{2^m} - x) + \hat{f}^{(0)}(x)(x^{2^m} - x)'}{s_k'(x)} \Big|_{x=\omega} \\ &\quad + \Big(\prod_{a \in V_k \setminus \{0\}} a \Big) \sum_{i \ne 0} \hat{f}^{(i)}(\omega) \frac{0}{s_k(\omega_{i2^k})} \\ &= \hat{f}^{(0)}(\omega), \quad \omega \in V_k. \end{aligned} \tag{45}$$

Since $\Lambda(\omega) = 0$ for $\omega \in \mathcal{E}$, we have

$$\hat{f}(\omega) = f(\omega)\Lambda(\omega) = 0, \quad \omega \in \mathcal{E}. \tag{46}$$

Extracting the first and the third terms of $g(x)$ in (44), define the polynomial

$$\hat{g}(x) = (\hat{f}^{(0)}(x))' + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}^{(i)}(x). \tag{47}$$

We have the following result.

Theorem 2: For low-rate RS codes, the erasures in the message part can be computed by

$$f(\omega) = \frac{\hat{g}(\omega)}{\Lambda'(\omega)}, \quad \omega \in \mathcal{E} \cap V_k. \tag{48}$$

Proof: If $\omega \in \mathcal{E} \cap V_k$, then

$$\begin{aligned} \hat{f}'(\omega) &= g(\omega) \\ &= \big(\hat{f}^{(0)}(\omega)\big)' + \Big(\sum_{a \in \mathbb{F}_{2^m} \setminus V_k} \frac{1}{a} \Big) \hat{f}^{(0)}(\omega) \\ &\quad + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}^{(i)}(\omega) \\ &= \big(\hat{f}^{(0)}(\omega)\big)' + \Big(\sum_{a \in \mathbb{F}_{2^m} \setminus V_k} \frac{1}{a} \Big) \hat{f}(\omega) \\ &\quad + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}^{(i)}(\omega) \\ &= \big(\hat{f}^{(0)}(\omega)\big)' + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}^{(i)}(\omega) \\ &= \hat{g}(\omega), \quad \omega \in \mathcal{E} \cap V_k, \end{aligned} \tag{49}$$

where the first equality is due to (42), the second equality is due to (44), the third equality is due to (45), the fourth equality is due to (46), and the last equality is due to (47).

According to the systematic encoding of low-rate RS codes, $V_k = \{\omega_0, \omega_1, \cdots, \omega_{2^k-1}\}$ is index set for the message symbols. By substituting (49) into (25), we obtain (48). ■

Invoking the LCH-FFT, the erasure decoding of low-rate RS codes can be performed as follows. First, compute $\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ and $\{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}$. Then compute $(\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1}))$ by (27). These computations are the same as Algorithm 3, but from this point the two algorithms diverge. Next, compute $\hat{f}^{(i)}(x) = \sum_{j=0}^{2^k-1} \hat{f}_j^{(i)} \bar{X}_j(x)$, $0 \le i \le 2^{m-k}-1$, by

$$\begin{aligned}&(\hat{f}_0^{(i)}, \hat{f}_1^{(i)}, \ldots, \hat{f}_{2^k-1}^{(i)})\\ &= \mathrm{IFFT}_{\bar{\mathbb{X}}}((\hat{f}^{(i)}(\omega_{i2^k}), \ldots, \hat{f}^{(i)}(\omega_{i2^k+2^k-1})), k, \omega_{i2^k})\\ &= \mathrm{IFFT}_{\bar{\mathbb{X}}}((\hat{f}(\omega_{i2^k}), \hat{f}(\omega_{i2^k+1}), \ldots, \hat{f}(\omega_{i2^k+2^k-1})), k, \omega_{i2^k}),\end{aligned} \tag{50}$$

where the second equality follows from (41). Then compute the formal derivative $(\hat{f}^{(0)}(x))' = \sum_{i=0}^{2^k-1} \hat{f}_i^{\prime(0)} \bar{X}_i(x)$. Next, based on (47), compute $\hat{g}(x) = \sum_{j=0}^{2^k-1} \hat{g}_j \bar{X}_j(x)$ by

$$\hat{g}_j = \hat{f}_j^{\prime(0)} + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}_j^{(i)}, \quad 0 \le j \le 2^k - 1.$$

Then compute $(\hat{g}(\omega_0), \hat{g}(\omega_1), \ldots, \hat{g}(\omega_{2^k-1}))$ by

$$\begin{aligned}&(\hat{g}(\omega_0), \hat{g}(\omega_1), \ldots, \hat{g}(\omega_{2^k-1}))\\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2^k-1}), k, \omega_0).\end{aligned} \tag{51}$$

Finally, compute the erasures in the message part by (48).

For clarity, Algorithm 4 describes the erasure decoding algorithm for low-rate RS codes. The complexity is analyzed as follows. The preliminary step corresponding to Line 1 is the same as Algorithm 3, requiring complexity $O(N \log N)$ [30]. The main step corresponds to Lines 2–7. Line 2 requires K multiplications. Line 3 requires $N \log K$ additions and $\frac{1}{2} N \log K$ multiplications. Line 4 computes the formal derivative of a polynomial of degree less than K in the LCH basis, which requires K multiplications and around $\frac{1}{2} K \log K$ additions [30]. Line 5 requires $(N-K)$ multiplications and $(N-K)$ additions. Line 6 requires $K \log K$ additions and $\frac{1}{2} K \log K$ multiplications. Line 7 requires at most K multiplications. In summary, the main step achieves complexity $O(N \log K)$.

Considering that Algorithm 4 is built on Algorithm 3, it is worth making a comparison of the two algorithms. First, Algorithm 3 applies to both low-rate and high-rate RS codes, while Algorithm 4 applies only to low-rate RS codes. Second, for the main step, Algorithm 3 achieves complexity $O(N \log N)$ while Algorithm 4 achieves complexity $O(N \log K)$. Finally, Algorithm 3 recovers the erasures in both the message and parity parts while Algorithm 4 recovers only the erasures in the message part. To recover the erasures in the parity part when using Algorithm 4, the systematic encoding with complexity $O(N \log K)$ can be further performed without essentially affecting the decoding complexity.

Algorithm 4: A Fast Erasure Decoding Algorithm for Low-Rate RS Codes

Input: The erasure locator set $\mathcal{E}$ and the received symbols $\{f(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$

Output: The erasures in the message part $\{f(\omega) : \omega \in \mathcal{E} \cap V_k\}$.

1: Compute $\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ and $\{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}$.

2: Compute $(\hat{f}(\omega_0), \hat{f}(\omega_1), \ldots, \hat{f}(\omega_{2^m-1}))$ by

$$\hat{f}(\omega) = \begin{cases} 0, & \text{if } \omega \in \mathcal{E}, \\ f(\omega)\Lambda(\omega), & \text{if } \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}. \end{cases}$$

3: Compute $\hat{f}^{(i)}(x) = \sum_{j=0}^{2^k-1} \hat{f}_j^{(i)} \bar{X}_j(x), 0 \le i \le 2^{m-k}-1$, by

$$\begin{aligned}&(\hat{f}_0^{(i)}, \hat{f}_1^{(i)}, \ldots, \hat{f}_{2^k-1}^{(i)})\\ &= \mathrm{IFFT}_{\bar{\mathbb{X}}}((\hat{f}(\omega_{i2^k}), \ldots, \hat{f}(\omega_{i2^k+2^k-1})), k, \omega_{i2^k}).\end{aligned}$$

4: Compute the formal derivative $(\hat{f}^{(0)}(x))' = \sum_{i=0}^{2^k-1} \hat{f}_i^{\prime(0)} \bar{X}_i(x)$.

5: Compute $\hat{g}(x) = \sum_{j=0}^{2^k-1} \hat{g}_j \bar{X}_j(x)$ by

$$\hat{g}_j = \hat{f}_j^{\prime(0)} + \sum_{i=1}^{2^{m-k}-1} \frac{\prod_{a \in V_k \setminus \{0\}} a}{s_k(\omega_{i2^k})} \hat{f}_j^{(i)}, 0 \le j \le 2^k - 1.$$

6: Compute $(\hat{g}(\omega_0), \hat{g}(\omega_1), \ldots, \hat{g}(\omega_{2^k-1}))$ by

$$\begin{aligned}&(\hat{g}(\omega_0), \hat{g}(\omega_1), \ldots, \hat{g}(\omega_{2^k-1}))\\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2^k-1}), k, \omega_0).\end{aligned}$$

7: Compute the erasures in the message part by

$$f(\omega) = (\Lambda'(\omega))^{-1} \hat{g}(\omega), \quad \omega \in \mathcal{E} \cap V_k.$$

V. A Fast Erasure Decoding Algorithm for High-Rate RS Codes

In this section, we present a fast erasure decoding algorithm for high-rate RS codes. Note that, for a high-rate RS(N, K) code, $N-K$ is much smaller than N. If we can reduce the complexity of some decoding steps from $O(N \log N)$ to $O(N \log(N-K))$, the overall decoding complexity can be reduced. Based on Property 2 of Lemma 1, the syndromes are computed by using LCH-FFT. Then, a key equation and a Forney-like formula are presented. Their complexities are of $O(N \log(N-K))$. The derivation is as follows.

Let $(f(\omega_0), f(\omega_1), \ldots, f(\omega_{2^m-1}))$ be the transmitted codeword, where $f(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ is a polynomial of degree less than K. Let $\mathcal{E} \subset \{\omega_0, \omega_1, \ldots, \omega_{2^m-1}\}$ be the set of erasure locators with $|\mathcal{E}| = 2^m - K = 2^t$. The erasure locator polynomial is $\Lambda(x) = \prod_{\omega \in \mathcal{E}} (x - \omega)$.

Define a polynomial $\tilde{f}(x) \in \mathbb{F}_{2^m}[x]/(x^{2^m} - x)$ of degree less than 2^m by

$$\tilde{f}(\omega) = \begin{cases} 0, & \text{if } \omega \in \mathcal{E}, \\ f(\omega), & \text{if } \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}, \end{cases} \tag{52}$$

where $\{f(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ are the received symbols. Since

$\Lambda(\omega) = 0$ for $\omega \in \mathcal{E}$, we have

$$\tilde{f}(\omega)\Lambda(\omega) = f(\omega)\Lambda(\omega), \quad \omega \in \mathbb{F}_{2^m}, \tag{53}$$

which implies that

$$\tilde{f}(x)\Lambda(x) = f(x)\Lambda(x) \mod s_m(x). \tag{54}$$

Therefore, there exists some $q(x) \in \mathbb{F}_{2^m}[x]$ such that

$$f(x)\Lambda(x) = \tilde{f}(x)\Lambda(x) + q(x)s_m(x), \tag{55}$$

where $\deg(q(x)) < \deg(\Lambda(x)) = 2^t$.

Dividing $\tilde{f}(x)$ by $\bar{X}_{2^m-2^t}(x)$, we obtain

$$\tilde{f}(x) = h(x)\bar{X}_{2^m-2^t}(x) + r(x), \tag{56}$$

where $\deg(h(x)) < 2^t$, $r(x) = 0$ or $\deg(r(x)) < 2^m - 2^t$. Taking $n = m$ and $k = t$ in (10) and substituting the resulting $s_m(x)$ and (56) into (55), we obtain

$$\begin{aligned} &f(x)\Lambda(x) + r(x)\Lambda(x) + \Big(\sum_{i=t}^{m-1} p_{2^m-2^i}\bar{X}_{2^m-2^i}(x)s_i(v_i)\Big)q(x) \\ &= \bar{X}_{2^m-2^t}(x)\big(h(x)\Lambda(x) + p_{2^m-2^t}q(x)s_t(x)\big). \end{aligned} \tag{57}$$

Extracting the second term on the right-hand side of (57), we obtain the key equation

$$z(x) = h(x)\Lambda(x) + p_{2^m-2^t}q(x)s_t(x), \tag{58}$$

where $\deg(z(x)) < \deg(\Lambda(x))$.

The erasures can be computed as follows. The formal derivative of (55) is

$$\begin{aligned} &f'(x)\Lambda(x) + f(x)\Lambda'(x) \\ &= \tilde{f}'(x)\Lambda(x) + \tilde{f}(x)\Lambda'(x) + q'(x)s_m(x) + q(x), \end{aligned} \tag{59}$$

Substituting $x = \omega \in \mathcal{E}$ into (59), we have

$$f(\omega)\Lambda'(\omega) = q(\omega). \tag{60}$$

Therefore, the erasures can be computed by the Forney-like formula

$$f(\omega) = \frac{q(\omega)}{\Lambda'(\omega)}, \quad \omega \in \mathcal{E}. \tag{61}$$

The following provides an alternative method for computing the erasures by distinguishing between the message symbols and the parity symbols. According to the systematic encoding of high-rate RS codes, $V_t = \{\omega_0, \omega_1, \ldots, \omega_{2^t-1}\}$ is the index set for the parity symbols and $\mathbb{F}_{2^m} \setminus V_t = \{\omega_{2^t}, \omega_{2^t+1}, \ldots, \omega_{2^m-1}\}$ is the index set for the message symbols. Based on (58), since $\Lambda(\omega) = 0$ for $\omega \in \mathcal{E}$ and $s_t(\omega) \neq 0$ for $\omega \in \mathbb{F}_{2^m} \setminus V_t$, we have

$$q(\omega) = \frac{z(\omega)}{p_{2^m-2^t}s_t(\omega)}, \quad \omega \in \mathcal{E} \cap (\mathbb{F}_{2^m} \setminus V_t). \tag{62}$$

Substituting (62) into (61), the erasures in the message part can be computed by

$$f(\omega) = \frac{z(\omega)}{p_{2^m-2^t}s_t(\omega)\Lambda'(\omega)}, \quad \omega \in \mathcal{E} \cap (\mathbb{F}_{2^m} \setminus V_t). \tag{63}$$

The formal derivative of (58) is

$$\begin{aligned} z'(x) =& h'(x)\Lambda(x) + h(x)\Lambda'(x) + p_{2^m-2^t}q'(x)s_t(x) \\ &+ p_{2^m-2^t}q(x)s_t'(x). \end{aligned} \tag{64}$$

Note that $s_t'(x) = \prod_{a \in V_t \setminus \{0\}} a$. Based on (64), since $\Lambda(\omega) = 0$ for $\omega \in \mathcal{E}$ and $s_t(\omega) = 0$ for $\omega \in V_t$, we have

$$q(\omega) = \frac{z'(\omega) - h(\omega)\Lambda'(\omega)}{p_{2^m-2^t}\prod_{a \in V_t \setminus \{0\}} a}, \quad \omega \in \mathcal{E} \cap V_t. \tag{65}$$

Substituting (65) into (61), the erasures in the parity part can be computed by

$$f(\omega) = \frac{z'(\omega) - h(\omega)\Lambda'(\omega)}{p_{2^m-2^t}(\prod_{a \in V_t \setminus \{0\}} a)\Lambda'(\omega)}, \quad \omega \in \mathcal{E} \cap V_t. \tag{66}$$

Using the LCH-FFT, erasure decoding of high-rate RS codes can be performed as follows. First, compute $\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ and $\{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}$. Then compute $(\tilde{f}(\omega_0), \tilde{f}(\omega_1), \ldots, \tilde{f}(\omega_{2^m-1}))$ by (52). Next, based on Corollary 2 and Property 2 of Lemma 1, compute $h(x) = \sum_{i=0}^{2^t-1} h_i\bar{X}_i(x)$ by

$$\begin{aligned} &(h_0, h_1, \ldots, h_{2^t-1}) \\ &= \sum_{i=0}^{2^{m-t}-1} \mathrm{IFFT}_{\bar{\mathbb{X}}}((\tilde{f}(\omega_{i2^t}), \ldots, \tilde{f}(\omega_{i2^t+2^t-1})), t, \omega_{i2^t}). \end{aligned} \tag{67}$$

Then compute $(h(\omega_0), h(\omega_1), \ldots, h(\omega_{2^t-1}))$ by

$$\begin{aligned} &(h(\omega_0), h(\omega_1), \ldots, h(\omega_{2^t-1})) \\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((h_0, h_1, \ldots, h_{2^t-1}), t, \omega_0). \end{aligned} \tag{68}$$

Next, based on (58), compute $(z(\omega_0), z(\omega_1), \ldots, z(\omega_{2^t-1}))$ by

$$z(\omega_i) = h(\omega_i)\Lambda(\omega_i), \quad 0 \leq i \leq 2^t - 1. \tag{69}$$

Then compute $z(x) = \sum_{i=0}^{2^t-1} z_i\bar{X}_i(x)$ by

$$\begin{aligned} &(z_0, z_1, \ldots, z_{2^t-1}) \\ &= \mathrm{IFFT}_{\bar{\mathbb{X}}}((z(\omega_0), z(\omega_1), \ldots, z(\omega_{2^t-1})), t, \omega_0). \end{aligned} \tag{70}$$

and its formal derivative $z'(x) = \sum_{i=0}^{2^t-1} z_i'\bar{X}_i(x)$. Next, compute the subvectors

$$\begin{aligned} &(z(\omega_{i2^t}), z(\omega_{i2^t+1}), \ldots, z(\omega_{i2^t+2^t-1})) \\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((z_0, z_1, \ldots, z_{2^t-1}), t, \omega_{i2^t}), \quad 1 \leq i \leq 2^{m-t} - 1, \end{aligned} \tag{71}$$

and

$$\begin{aligned} &(z'(\omega_0), z'(\omega_1), \ldots, z'(\omega_{2^t-1})) \\ &= \mathrm{FFT}_{\bar{\mathbb{X}}}((z_0', z_1', \ldots, z_{2^t-1}'), t, \omega_0). \end{aligned} \tag{72}$$

Finally, compute the erasures in the message symbols by (63) and the erasures in the parity symbols by (66).

For clarity, Algorithm 5 describes the erasure decoding algorithm for high-rate RS codes. The complexity is analyzed as follows. The preliminary step corresponding to Line 1 requires complexity $O(N \log N)$ [30]. The main step corresponds to Lines 2–11. The assignment operations in Line 2 are not taken into account in assessing the complexity. Line 3

Algorithm 5: A Fast Erasure Decoding Algorithm for High-Rate RS Codes

Input: The erasure locator set $\mathcal{E}$ and the received symbols $\{f(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$

Output: The erasures $\{f(\omega) : \omega \in \mathcal{E}\}$.

1: Compute $\{\Lambda(\omega) : \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}\}$ and $\{(\Lambda'(\omega))^{-1} : \omega \in \mathcal{E}\}$.

2: Compute $(\tilde{f}(\omega_0), \tilde{f}(\omega_1), \ldots, \tilde{f}(\omega_{2^m-1}))$ by

$$\tilde{f}(\omega) = \begin{cases} 0, & \text{if} \quad \omega \in \mathcal{E}, \\ f(\omega), & \text{if} \quad \omega \in \mathbb{F}_{2^m} \setminus \mathcal{E}. \end{cases}$$

3: Compute $h(x) = \sum_{i=0}^{2^t-1} h_i \bar{X}_i(x)$ by

$$(h_0, h_1, \ldots, h_{2^t-1}) = \sum_{i=0}^{2^{m-t}-1} \mathrm{IFFT}_{\bar{\mathbb{X}}}((\tilde{f}(\omega_{i2^t}), \ldots, \tilde{f}(\omega_{i2^t+2^t-1})), t, \omega_{i2^t}).$$

4: Compute $(h(\omega_0), h(\omega_1), \ldots, h(\omega_{2^t-1}))$ by

$$(h(\omega_0), h(\omega_1), \ldots, h(\omega_{2^t-1})) = \mathrm{FFT}_{\bar{\mathbb{X}}}((h_0, h_1, \ldots, h_{2^t-1}), t, \omega_0).$$

5: Compute $(z(\omega_0), z(\omega_1), \ldots, z(\omega_{2^t-1}))$ by

$$z(\omega_i) = h(\omega_i)\Lambda(\omega_i), \quad 0 \le i \le 2^t - 1.$$

6: Compute $z(x) = \sum_{i=0}^{2^t-1} z_i \bar{X}_i(x)$ by

$$(z_0, z_1, \ldots, z_{2^t-1}) = \mathrm{IFFT}_{\bar{\mathbb{X}}}((z(\omega_0), z(\omega_1), \ldots, z(\omega_{2^t-1})), t, \omega_0).$$

7: Compute the formal derivative $z'(x) = \sum_{i=0}^{2^t-1} z'_i \bar{X}_i(x)$.

8: Compute $(z(\omega_{i2^t}), z(\omega_{i2^t+1}), \ldots, z(\omega_{i2^t+2^t-1}))$, $1 \le i \le 2^{m-t} - 1$, by

$$(z(\omega_{i2^t}), z(\omega_{i2^t+1}), \ldots, z(\omega_{i2^t+2^t-1})) = \mathrm{FFT}_{\bar{\mathbb{X}}}((z_0, z_1, \ldots, z_{2^t-1}), t, \omega_{i2^t}).$$

9: Compute $(z'(\omega_0), z'(\omega_1), \ldots, z'(\omega_{2^t-1}))$ by

$$(z'(\omega_0), z'(\omega_1), \ldots, z'(\omega_{2^t-1})) = \mathrm{FFT}_{\bar{\mathbb{X}}}((z'_0, z'_1, \ldots, z'_{2^t-1}), t, \omega_0).$$

10: Compute the erasures in the message part by

$$f(\omega) = \frac{z(\omega)}{p_{2^m-2^t} s_t(\omega) \Lambda'(\omega)}, \quad \omega \in \mathcal{E} \cap (\mathbb{F}_{2^m} \setminus V_t).$$

11: Compute the erasures in the parity part by

$$f(\omega) = \frac{z'(\omega) - h(\omega)\Lambda'(\omega)}{p_{2^m-2^t}(\prod_{a \in V_t \setminus \{0\}} a)\Lambda'(\omega)}, \quad \omega \in \mathcal{E} \cap V_t.$$

requires $N \log(N-K) + K$ additions and $\frac{1}{2} N \log(N-K)$ multiplications. Line 4 requires $(N-K)\log(N-K)$ additions and $\frac{1}{2}(N-K)\log(N-K)$ multiplications. Line 5 requires $(N-K)$ multiplications. Line 6 requires $(N-K)\log(N-K)$ additions and $\frac{1}{2}(N-K)\log(N-K)$ multiplications. Line 7 requires $(N-K)$ multiplications and around $\frac{1}{2}(N-K)\log(N-K)$ additions [30]. Line 8 requires $K\log(N-K)$ additions and $\frac{1}{2}K\log(N-K)$ multiplications. Line 9 requires $(N-K)\log(N-K)$ additions and $\frac{1}{2}(N-K)\log(N-K)$ multiplications. Line 10 requires at most $2(N-K)$ multiplications. Line 11 requires at most $(N-K)$ additions and $3(N-K)$ multiplications. In summary, the main step achieves complexity $O(N\log(N-K))$. If it is not necessary to recover the erasures in the parity symbols, Lines 7, 9, 11 can be removed from the algorithm.

VI. Complexity Comparison and Simulations

Table I compares the complexity of six algorithms for RS erasure decoding. We provide more explanations about these algorithms as follows.

1) The standard algorithm [2]. A codeword is obtained by multiplying a length-K message vector with the $K \times N$ generator matrix (Vandermonde or Cauchy). The preliminary step involves inverting a $K \times K$ submatrix using the Gaussian elimination, which requires a complexity of $O(K^3)$. In [9], a method for computing the inverse matrix (called wide-sense systematic generator matrix therein) was proposed, which is similar to the Lagrange interpolation and requires $O(K^2)$ complexity. The main step recovers the message vector by multiplying the length-K vector composed of received symbols with the inverse matrix obtained in the preliminary step, thus requiring $O(K^2)$ complexity.
2) The Didier's algorithm [35]. The algorithm is derived based on the fast Walsh–Hadamard transform (FWHT). The preliminary step essentially computes the evaluations of the erasure locator polynomial and achieves $O(N \log N)$ complexity by exploiting the FWHT. The main step performs a special form of polynomial evaluation and achieves $O(N \log^2 N)$ complexity by exploiting the FWHT.
3) The LCH-FFT-based algorithm [30]. We have described the algorithm in Algorithm 3. It applies to both low-rate and high-rate RS codes, as compared to our proposed two algorithms. Based on [35], the preliminary step achieves $O(N \log N)$ complexity, see [30, Appendix A] for detail. The main step achieves $O(N \log N)$ complexity by exploiting the LCH-FFT.
4) The Reed–Muller (RM) transform based algorithm [36]. The RS code is defined by the Vandermonde parity-check matrix. The preliminary step solves a Vandermonde system of $(N-K)$ linear equations using a lower-upper (LU) decomposition approach, which requires a complexity of $O((N-K)^2)$. For the complexity of the main step, there is no explicit expression for the term $\mathcal{E}(N,K)$. However, numerical results show that by fixing $(N-K)$, the ratio $\frac{\mathcal{E}(N,K)}{K}$ gradually approaches 0 as N continues to increase [36, Fig. 4]. This indicates that the algorithm is especially suitable for very high-rate RS codes, i.e., $(N-K)$ is small relative to N.
5) Our proposed Algorithm 4. The algorithm applies to low-rate RS codes. It improves on the existing LCH-FFT-based algorithm [30] in the main step. The prelim-

TABLE I
COMPLEXITY COMPARISON OF ERASURE DECODING ALGORITHMS FOR RS CODES OVER BINARY EXTENSION FIELDS.

Algorithm	Decoding Complexity	
	Preliminary step	Main step
Standard [2]	$O(K^3)$	$O(K^2)$
Didier [35]	$O(N \log N)$	$O(N \log^2 N)$
LCH-FFT-based [30]	$O(N \log N)$	$O(N \log N)$
RM-transform-based [36]	$O((N-K)^2)$	$O(N \log(N-K) + (N-K)^2 + \mathcal{E}(N,K))$
Proposed Algorithm 4 (Low-rate)	$O(N \log N)$	$O(N \log K)$
Proposed Algorithm 5 (High-rate)	$O(N \log N)$	$O(N \log(N-K))$

inary step keeps the same, thus requiring $O(N \log N)$ complexity. The main step achieves $O(N \log K)$ complexity by exploiting the LCH-FFT.

6) Our proposed Algorithm 5. The algorithm applies to high-rate RS codes. The preliminary step is the same as that of the existing LCH-FFT-based algorithm [30], thus requiring $O(N \log N)$ complexity. The main step achieves $O(N \log(N-K))$ complexity by exploiting the LCH-FFT.

From Table I, we have the following observations: 1) The best previously known complexity is $O(N \log N)$, which is achieved by the LCH-FFT-based algorithm [30] that applies to both low-rate and high-rate RS codes; 2) Our proposed two algorithms, which apply respectively to low-rate and high-rate RS codes, achieve even better complexity in the main step, reaching $O(N \log(\min\{K, N-K\}))$. Thus, we can conclude that the proposed algorithms achieve the best complexity so far.

To assess the real-world performance, we have performed extensive simulations based on the libraries that are programmed with the Single Instruction Multiple Data (SIMD) [37]. These libraries include ISA-L [38] and Jerasure [39] for the standard algorithm, Leopard-RS [40] for the LCH-FFT-based algorithm, RMT-RS [36] for the RM-transform-based algorithm, and XD-RS [41] we provide for our proposed algorithms. We simulated RS(N, K) codes over $\mathbb{F}_{2^8}$ with $N = 256$ and $K = 8, 16, 32, 64, 128, 192, 224, 240, 248$. All simulations were performed on Windows 10 platform with Intel Core i7-9700 with 32 GB RAM. The performance metric for decoding an RS(N, K) code over $\mathbb{F}_{2^m}$ is

$$Throughput = \frac{K \times m \times Num\_codeword \times Num\_group}{10^6 \times 8 \times Time} \quad (MB/s),$$

where Num_group is the number of simulated groups, $Num\_codeword$ is the number of codewords in each group, $Time$ is the total run time (in seconds). Note that all the codewords in each group share the same set of erasure locators and the preliminary step for these codewords is performed only once. In all simulations, we set $Num\_group = 10^6$ and $Num\_codeword = 1024$.

Considering that each library may support one or both of two different versions of SIMD, namely SSE and AVX2, for fair comparison, we present the simulation results in the following manner. Fig. 1 presents the throughput performance for the algorithms with libraries supporting SSE. It is seen that our proposed algorithms achieve significant throughput improvements compared to the standard algorithm and the RM-transform-based algorithm. Fig. 2 presents the throughput performance for the algorithms with libraries supporting AVX-2. It is seen that our proposed algorithms achieve significant throughput improvements compared to the standard algorithm and the LCH-FFT-based algorithm.

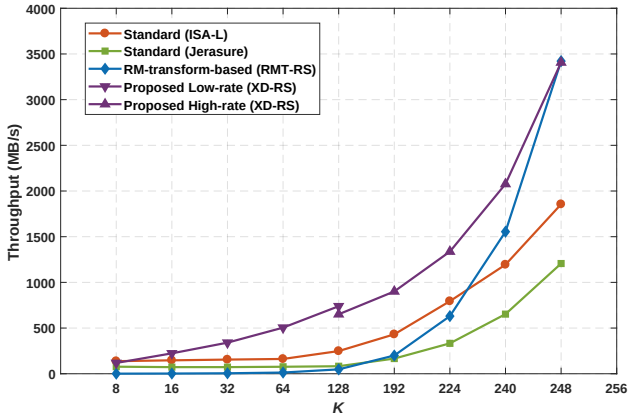

Fig. 1. Throughput performance for RS$(256, K)$ codes over $\mathbb{F}_{2^8}$ for the algorithms with libraries supporting SSE.

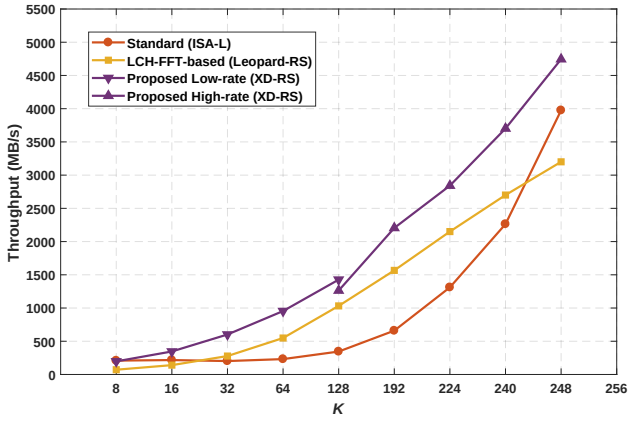

Fig. 2. Throughput performance for RS$(256, K)$ codes over $\mathbb{F}_{2^8}$ for the algorithms with libraries supporting AVX2.

VII. CONCLUSION

We utilize the LCH-FFT to design two fast erasure decoding algorithms, which apply, respectively, to low-rate and high-rate RS codes over binary extension fields. Both algorithms achieve

the best complexity so far, reaching $O(N\log(\min\{K, N-K\}))$. Our program library, namely XD-RS, for the proposed algorithms outperforms the libraries available for state-of-the-art algorithms in terms of throughput. As the essential foundation for the proposed algorithms, a new interpolation formula is derived, which may be of interest in its own right.

In [43], Hartmann and Rudolph presented an optimum symbol-by-symbol decoding rule for linear codes based on the dual codes, which makes them firmly believe in the coding-complexity Folk Theorem: "The complexity of any function defined on a linear code is comparable to the complexity of (essentially) that same function defined on the dual code." Noticing that the dual code of an RS(N, K) code is an RS$(N, N-K)$ code, our proposed two decoding algorithms, with complexity $O(N\log(\min\{K, N-K\}))$, serve as another supporting example.

Appendix A. Proof of Theorem 1

Proof: For $0 \le i, \ell \le 2^{n-k}-1$, consider computing $T^{(i)}(\omega_{\ell 2^k+j}+\beta)$, $0 \le j \le 2^k-1$. We first note that

$$s_n(\omega_{\ell 2^k+j}) - s_n(\omega_{i2^k}) = 0, \tag{73}$$

and that

$$s_k(\omega_{\ell 2^k+j}) - s_k(\omega_{i2^k}) = 0 \quad \text{if and only if} \quad i = \ell. \tag{74}$$

Therefore, based on the definition of $T^{(i)}(x)$ in (32), if $i = \ell$, we have

$$\begin{aligned} T^{(\ell)}(\omega_{\ell 2^k+j}+\beta) &= \left.\frac{s_n(x)-s_n(\omega_{\ell 2^k}+\beta)}{s_k(x)-s_k(\omega_{\ell 2^k}+\beta)}\right|_{x=\omega_{\ell 2^k+j}+\beta} \\ &= \left.\frac{s_n'(x)}{s_k'(x)}\right|_{x=\omega_{\ell 2^k+j}+\beta} \\ &= \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a}, \end{aligned} \tag{75}$$

where the second equality follows from the L'Hospital's rule. If $i \ne \ell$, we have

$$\begin{aligned} T^{(i)}(\omega_{\ell 2^k+j}+\beta) &= \left.\frac{s_n(x)-s_n(\omega_{i2^k}+\beta)}{s_k(x)-s_k(\omega_{i2^k}+\beta)}\right|_{x=\omega_{\ell 2^k+j}+\beta} \\ &= \frac{0}{s_k(\omega_{\ell 2^k+j}-\omega_{i2^k})} \\ &= 0. \end{aligned} \tag{76}$$

Denote the right hand side of (31) by $\bar{f}(x)$. Taking $x = \omega_{\ell 2^k+j}+\beta$, we have

$$\begin{aligned} &\bar{f}(\omega_{\ell 2^k+j}+\beta) \\ &= \frac{\prod_{a\in V_k\setminus\{0\}} a}{\prod_{a\in V_n\setminus\{0\}} a} \sum_{i=0}^{2^{n-k}-1} f^{(i)}(x)T^{(i)}(x)\Big|_{x=\omega_{\ell 2^k+j}+\beta} \\ &= \frac{\prod_{a\in V_k\setminus\{0\}} a}{\prod_{a\in V_n\setminus\{0\}} a} f^{(\ell)}(x)T^{(\ell)}(x)\Big|_{x=\omega_{\ell 2^k+j}+\beta} \\ &\quad + \frac{\prod_{a\in V_k\setminus\{0\}} a}{\prod_{a\in V_n\setminus\{0\}} a} \sum_{i\ne\ell} f^{(i)}(x)T^{(i)}(x)\Big|_{x=\omega_{\ell 2^k+j}+\beta} \\ &= f^{(\ell)}(\omega_{\ell 2^k+j}+\beta) \\ &= f(\omega_{\ell 2^k+j}+\beta), \quad 0 \le \ell \le 2^{n-k}-1,\ 0 \le j \le 2^k-1, \end{aligned} \tag{77}$$

where the third equality is due to (75) and (76). Rewriting (77), we have $\bar{f}(\omega_i+\beta) = f(\omega_i+\beta)$ for $0 \le i \le 2^n-1$. Considering that both $\bar{f}(x)$ and $f(x)$ have a degree less than 2^n, we have $\bar{f}(x) = f(x)$.

If $k = 0$, it follows from $s_0(x) = x$ that $s_0'(x) = 1$. In this case, (31) becomes (33). ∎

Appendix B. Proof of Corollary 1

Proof: Based on Theorem 1, the formal derivative of $f(x)$ is given by

$$\begin{aligned} &f'(x) = \\ &\frac{\prod_{a\in V_k\setminus\{0\}} a}{\prod_{a\in V_n\setminus\{0\}} a} \sum_{i=0}^{2^{n-k}-1} \Big(T^{(i)}(x)(f^{(i)}(x))' + (T^{(i)}(x))'f^{(i)}(x)\Big), \end{aligned} \tag{78}$$

where

$$\begin{aligned} &(T^{(i)}(x))' = \\ &\frac{\prod_{a\in V_n\setminus\{0\}} a}{s_k(x-(\omega_{i2^k}+\beta))} + \frac{\big(\prod_{a\in V_k\setminus\{0\}} a\big)s_n(x-(\omega_{i2^k}+\beta))}{s_k^2(x-(\omega_{i2^k}+\beta))}. \end{aligned} \tag{79}$$

Substituting (78) into (35), we have

$$\begin{aligned} g(x) =& \frac{\prod_{a\in V_k\setminus\{0\}} a}{\prod_{a\in V_n\setminus\{0\}} a} \sum_{i=0}^{2^{n-k}-1} \Big(T^{(i)}(x)(f^{(i)}(x))' \\ &+ (T^{(i)}(x))'f^{(i)}(x)\Big) \mod s_k(x-\beta). \end{aligned} \tag{80}$$

We first compute the term $T^{(i)}(x)(f^{(i)}(x))' \mod s_k(x-\beta)$ in (80). There are two cases.

1) $i \ne 0$.
   Then for any $\omega \in V_k+\beta$, we have
   $$T^{(i)}(\omega) = 0, \tag{81}$$
   which implies
   $$T^{(i)}(x) = 0 \mod s_k(x-\beta). \tag{82}$$
   Therefore,
   $$T^{(i)}(x)(f^{(i)}(x))' = 0 \mod s_k(x-\beta). \tag{83}$$

2) $i = 0$.
Then for any $\omega \in V_k + \beta$, we have

$$T^{(0)}(\omega) = \frac{s_n(x-\beta)}{s_k(x-\beta)}\Big|_{x=\omega} = \frac{s_n'(x)}{s_k'(x)}\Big|_{x=\omega} = \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a}, \quad (84)$$

which implies that

$$T^{(0)}(x) = \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a} \mod s_k(x-\beta). \quad (85)$$

Therefore,

$$T^{(0)}(x)(f^{(0)}(x))' = \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a}(f^{(0)}(x))' \mod s_k(x-\beta). \quad (86)$$

We next compute the term $(T^{(i)}(x))' f^{(i)}(x) \mod s_k(x-\beta)$ in (80). There are two cases.

1) $i \neq 0$.
Then for any $\omega \in V_k + \beta$, based on (79), we have

$$(T^{(i)}(\omega))' = \frac{\prod_{a\in V_n\setminus\{0\}} a}{s_k(\omega_{i2^k})}, \quad (87)$$

which implies

$$(T^{(i)}(x))' = \frac{\prod_{a\in V_n\setminus\{0\}} a}{s_k(\omega_{i2^k})} \mod s_k(x-\beta). \quad (88)$$

Therefore,

$$(T^{(i)}(x))' f^{(i)}(x) = \frac{\prod_{a\in V_n\setminus\{0\}} a}{s_k(\omega_{i2^k})} f^{(i)}(x) \mod s_k(x-\beta). \quad (89)$$

2) $i = 0$.
Then for any $\omega \in V_k + \beta$, based on (79), we compute (90), as shown at the bottom of this paper. Note that in (90), the last equality follows from

$$\prod_{a\in V_n\setminus V_k}(x-\beta-a)\Big|_{x=\omega} = \frac{s_n(x-\beta)}{s_k(x-\beta)}\Big|_{x=\omega} = \frac{s_n'(x-\beta)}{s_k'(x-\beta)}\Big|_{x=\omega} = \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a}, \quad (91)$$

and

$$\sum_{a\in V_n\setminus V_k}\frac{1}{x-\beta-a}\Big|_{x=\omega} = \sum_{a\in V_n\setminus V_k}\frac{1}{\omega-\beta-a} = \sum_{a\in V_n\setminus V_k}\frac{1}{a}. \quad (92)$$

Note that (92) is from the fact that for any $\omega \in V_k + \beta$,

$$\{\omega-\beta-a : a \in V_n \setminus V_k\} = \{a : a \in V_n \setminus V_k\}. \quad (93)$$

$$\begin{aligned}
(T^{(0)}(\omega))' &= \frac{\big(\prod_{a\in V_n\setminus\{0\}} a\big)s_k(x-\beta) + \big(\prod_{a\in V_k\setminus\{0\}} a\big)s_n(x-\beta)}{s_k^2(x-\beta)}\Big|_{x=\omega} \\
&= \frac{\big(\prod_{a\in V_n\setminus\{0\}} a\big)s_k(x-\beta) + \big(\prod_{a\in V_k\setminus\{0\}} a\big)s_n(x-\beta)}{s_k^2(x-\beta)}\Big|_{x=\omega} \\
&= \frac{\big(\prod_{a\in V_n\setminus\{0\}} a\big)s_k(x-\beta) + \big(\prod_{a\in V_k\setminus\{0\}} a\big)s_k(x-\beta)\big(\prod_{a\in V_n\setminus V_k}(x-\beta-a)\big)}{s_k^2(x-\beta)}\Big|_{x=\omega} \\
&= \frac{\big(\prod_{a\in V_n\setminus\{0\}} a\big) + \big(\prod_{a\in V_k\setminus\{0\}} a\big)\big(\prod_{a\in V_n\setminus V_k}(x-\beta-a)\big)}{s_k(x-\beta)}\Big|_{x=\omega} \\
&= \frac{\Big(\big(\prod_{a\in V_n\setminus\{0\}} a\big) + \big(\prod_{a\in V_k\setminus\{0\}} a\big)\big(\prod_{a\in V_n\setminus V_k}(x-\beta-a)\big)\Big)'}{s_k'(x-\beta)}\Big|_{x=\omega} \\
&= \sum_{a\in V_n\setminus V_k}\prod_{b\in V_n\setminus V_k, b\neq a}(x-\beta-b)\Big|_{x=\omega} \\
&= \sum_{a\in V_n\setminus V_k}\Big(\frac{1}{x-\beta-a}\prod_{b\in V_n\setminus V_k}(x-\beta-b)\Big)\Big|_{x=\omega} \\
&= \Big(\prod_{a\in V_n\setminus V_k}(x-\beta-a)\Big)\Big|_{x=\omega}\Big(\sum_{a\in V_n\setminus V_k}\frac{1}{x-\beta-a}\Big)\Big|_{x=\omega} \\
&= \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a}\sum_{a\in V_n\setminus V_k}\frac{1}{a}
\end{aligned} \quad (90)$$

It is implied by (90) that

$$(T^{(0)}(x))' = \frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a} \sum_{a\in V_n\setminus V_k} \frac{1}{a} \mod s_k(x-\beta). \tag{94}$$

Therefore,

$$\begin{aligned}&(T^{(0)}(x))' f^{(0)}(x)\\ &= \Big(\frac{\prod_{a\in V_n\setminus\{0\}} a}{\prod_{a\in V_k\setminus\{0\}} a} \sum_{a\in V_n\setminus V_k} \frac{1}{a}\Big) f^{(0)}(x) \mod s_k(x-\beta).\end{aligned} \tag{95}$$

Now we are able to prove (36). By substituting (83), (86), (89), and (95) into (80), we obtain (36). ∎

Appendix C. Proof of Corollary 2

Proof: Before we prove Corollary 2, we first give a lemma that will be used in the proof.

Lemma 2: For $0 \le k \le n \le m$,

$$\Big(\prod_{a\in V_k\setminus\{0\}} a\Big) p_{2^n-2^k} = \prod_{a\in V_n\setminus\{0\}} a. \tag{96}$$

Proof: Since $2^n - 2^k = 2^k + 2^{k+1} + \cdots + 2^{n-1}$, it follows from the definition of p_i in (8) that

$$\begin{aligned}&p_{2^n-2^k}\\ &= s_k(v_k)s_{k+1}(v_{k+1})\cdots s_{n-1}(v_{n-1})\\ &= \Big(\prod_{a\in V_k}(v_k-a)\Big)\cdots\Big(\prod_{a\in V_{n-1}}(v_{n-1}-a)\Big)\\ &= \Big(\prod_{a\in V_{k+1}\setminus V_k} a\Big)\Big(\prod_{a\in V_{k+2}\setminus V_{k+1}} a\Big)\cdots\Big(\prod_{a\in V_n\setminus V_{n-1}} a\Big),\\ &= \prod_{a\in V_n\setminus V_k} a,\end{aligned} \tag{97}$$

where the third equality is due to $V_{i+1} = V_i \cup (V_i + v_i)$ where $V_i + v_i = \{a + v_i : a \in V_i\} = \{v_i - a : a \in V_i\}$. Therefore, we have

$$\begin{aligned}\Big(\prod_{a\in V_k\setminus\{0\}} a\Big) p_{2^n-2^k} &= \Big(\prod_{a\in V_k\setminus\{0\}} a\Big)\Big(\prod_{a\in V_n\setminus V_k} a\Big)\\ &= \prod_{a\in V_n\setminus\{0\}} a.\end{aligned} \tag{98}$$

∎

Now we prove Corollary 2. First, we prove (38). If $k = n$, then $h(x) = f(x)$ and (38) is exactly the interpolation formula (33), which has been proved in Theorem 1. Now suppose $k < n$. Rewrite $s_n(x)$ in (10) as

$$s_n(x) = p_{2^n-2^k}(s_k(x) - s_k(v_k))\bar{X}_{2^n-2^k}(x) + \eta_1(x), \tag{99}$$

where

$$\eta_1(x) = \sum_{i=k+1}^{n-1} p_{2^n-2^i}\bar{X}_{2^n-2^i}(x)s_i(v_i), \tag{100}$$

such that $\eta_1(x) = 0$ or $\deg(\eta_1(x)) < \deg(\bar{X}_{2^n-2^k}(x))$. On both sides of (99), first subtracting $s_n(\omega_i+\beta)$ and then divided by $x - (\omega_i + \beta)$, we obtain (101), as shown at the bottom of this page.

Let $\eta_2(x)$ be the second term of the right hand side of (101). Then we can rewrite (101) as

$$\begin{aligned}&\frac{s_n(x) - s_n(\omega_i+\beta)}{x-(\omega_i+\beta)}\\ &= \frac{p_{2^n-2^k}(s_k(x) - s_k(\omega_i+\beta))}{x-(\omega_i+\beta)}\bar{X}_{2^n-2^k}(x) + \eta_2(x),\end{aligned} \tag{102}$$

where $\eta_2(x) = 0$ or $\deg(\eta_2(x)) < \deg(\bar{X}_{2^n-2^k}(x))$. Substituting (102) into (33), we obtain (103), as shown at the bottom of this paper. Note that in (103), the second equality follows from Lemma 2. As can be seen from (103), $h(x)$ is given by (38).

$$\begin{aligned}&\frac{s_n(x) - s_n(\omega_i+\beta)}{x-(\omega_i+\beta)}\\ &= \frac{p_{2^n-2^k}(s_k(x) - s_k(v_k))\bar{X}_{2^n-2^k}(x) + \eta_1(x) - s_n(\omega_i+\beta)}{x-(\omega_i+\beta)}\\ &= \frac{p_{2^n-2^k}(s_k(x) - s_k(\omega_i+\beta))}{x-(\omega_i+\beta)}\bar{X}_{2^n-2^k}(x) + \frac{p_{2^n-2^k}(s_k(\omega_i+\beta) - s_k(v_k))\bar{X}_{2^n-2^k}(x) + \eta_1(x) - s_n(\omega_i+\beta)}{x-(\omega_i+\beta)}.\end{aligned} \tag{101}$$

$$\begin{aligned}f(x) &= \Big(\prod_{a\in V_n\setminus\{0\}} a\Big)^{-1} p_{2^n-2^k} \sum_{i=0}^{2^n-1} f(\omega_i+\beta)\frac{s_k(x) - s_k(\omega_i+\beta)}{x-(\omega_i+\beta)}\bar{X}_{2^n-2^k}(x) + \Big(\prod_{a\in V_n\setminus\{0\}} a\Big)^{-1}\sum_{i=0}^{2^n-1}\Big(f(\omega_i+\beta)\eta_2(x)\Big)\\ &= \underbrace{\Big(\big(\prod_{a\in V_k\setminus\{0\}} a\big)^{-1}\sum_{i=0}^{2^n-1} f(\omega_i+\beta)\frac{s_k(x) - s_k(\omega_i+\beta)}{x-(\omega_i+\beta)}\Big)}_{h(x)}\bar{X}_{2^n-2^k}(x) + \underbrace{\Big(\prod_{a\in V_n\setminus\{0\}} a\Big)^{-1}\sum_{i=0}^{2^n-1}\Big(f(\omega_i+\beta)\eta_2(x)\Big)}_{r(x)}.\end{aligned} \tag{103}$$

$$
\begin{aligned}
h(x) &= \Big(\prod_{a \in V_k \setminus \{0\}} a \Big)^{-1} \sum_{j=0}^{2^n-1} f(\omega_j + \beta) \frac{s_k(x) - s_k(\omega_j + \beta)}{x - (\omega_j + \beta)} \\
&= \Big(\prod_{a \in V_k \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^{n-k}-1} \sum_{j=i2^k}^{(i+1)2^k-1} f(\omega_j + \beta) \frac{s_k(x) - s_k(\omega_j + \beta)}{x - (\omega_j + \beta)} \\
&= \Big(\prod_{a \in V_k \setminus \{0\}} a \Big)^{-1} \sum_{i=0}^{2^{n-k}-1} \sum_{j=0}^{2^k-1} f(\omega_{i2^k+j} + \beta) \frac{s_k(x) - s_k(\omega_{i2^k+j} + \beta)}{x - (\omega_{i2^k+j} + \beta)} \\
&= \sum_{i=0}^{2^{n-k}-1} \Big(\prod_{a \in V_k \setminus \{0\}} a \Big)^{-1} \sum_{j=0}^{2^k-1} f^{(i)}(\omega_j + \omega_{i2^k} + \beta) \frac{s_k(x) - s_k(\omega_j + \omega_{i2^k} + \beta)}{x - (\omega_j + \omega_{i2^k} + \beta)} \\
&= \sum_{i=0}^{2^{n-k}-1} f^{(i)}(x). \qquad (104)
\end{aligned}
$$

Next, we prove (39). Based on (38), we have (104), as shown at the top of this page. Note that in (104), the fourth equality is due to $\omega_{i2^k+j} = \omega_{i2^k} + \omega_j$ for $0 \le j \le 2^k - 1$, and the last equality is due to Theorem 1.

Finally, we prove (40). The polynomial $f(x)$ can be written as

$$
\begin{aligned}
& f(x) \\
&= \sum_{i=0}^{2^n-1} f_i \bar{X}_i(x) \\
&= \sum_{i=0}^{2^n-2^k-1} f_i \bar{X}_i(x) + \sum_{i=2^n-2^k}^{2^n-1} f_i \bar{X}_i(x) \\
&= \sum_{i=0}^{2^n-2^k-1} f_i \bar{X}_i(x) + \sum_{i=0}^{2^k-1} f_{2^n-2^k+i} \bar{X}_{2^n-2^k+i}(x) \\
&= \underbrace{\sum_{i=0}^{2^n-2^k-1} f_i \bar{X}_i(x)}_{r(x)} + \underbrace{\Big(\sum_{i=0}^{2^k-1} f_{2^n-2^k+i} \bar{X}_i(x) \Big)}_{h(x)} \bar{X}_{2^n-2^k}(x).
\end{aligned} \qquad (105)
$$

Therefore, the representation of $h(x)$ in the LCH basis is given by (40). ∎

REFERENCES

[1] I. S. Reed and G. Solomon, "Polynominal codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[2] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *J. ACM*, vol. 36, no. 2, pp. 335–348, Apr. 1989.

[3] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Comput. Commun. Rev.*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[4] J. Bloemer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, "An XOR-based erasure-resilient coding scheme," ICSI Technical Report TR-95-048, International Computer Science Institute, Aug. 1995.

[5] J. M. Pollard, "The fast Fourier transform in a finite field," *Math. Comput.*, vol. 25, no. 114, pp. 365–374, Apr. 1971.

[6] J. Justesen, "On the complexity of decoding Reed–Solomon codes (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 22, no. 2, pp. 237–238, Mar. 1976.

[7] I. S. Reed, R. Scholtz, T.-K. Truong, and L. Welch, "The fast decoding of Reed–Solomon codes using Fermat theoretic transforms and continued fractions," *IEEE Trans. Inf. Theory*, vol. 24, no. 1, pp. 100–106, Jan. 1978.

[8] I. S. Reed, T.-K. Truong, and L. Welch, "The fast decoding of Reed–Solomon codes using Fermat transforms (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 24, no. 4, pp. 497–499, Jul. 1978.

[9] J. Hong and M. Vetterli, "Simple algorithms for BCH decoding," *IEEE Trans. Commun.*, vol. 43, no. 8, pp. 2324–2333, Aug. 1995.

[10] C. Chen, B. Bai, X. Ma, Y. S. Han, N. Tang, and X. Wang, "Efficient decoding of a class of Reed–Solomon codes over Fermat fields," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Athens, Greece, Jul. 2024, pp. 476–481.

[11] F. P. Preparata, "Holographic dispersal and recovery of information," *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1123–1124, Sep. 1989.

[12] J. Lacan and J. Fimes, "Systematic MDS erasure codes based on Vandermonde matrices," *IEEE Commun. Lett.*, vol. 8, no. 9, pp. 570–572, Sep. 2004.

[13] R. Dianat and F. Marvasti, "FFT-based fast Reed–Solomon codes with arbitrary block lengths and rates," *IEE Proc. Commun.*, vol. 152, no. 2, pp. 151–156, Apr. 2005.

[14] A. Soro and J. Lacan, "FNT-based Reed–Solomon erasure codes," in *Proc. 7th Annu. IEEE Consumer Commun. and Network. Conf.*, Piscataway, NJ, USA, Jan. 2010, pp. 466–470.

[15] S.-J. Lin and W.-H. Chung, "An efficient (n, k) information dispersal algorithm for high code rate system over Fermat fields," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2036–2039, Dec. 2012.

[16] S.-J. Lin and W.-H. Chung, "An efficient (n, k) information dispersal algorithm based on Fermat number transforms", *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 8, pp. 1371–1383, Aug. 2013.

[17] R. E. Blahut, *Fast Algorithms for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[18] D. V. Sarwate, "Semi-fast Fourier transforms over GF(2^m)," *IEEE Trans. Comput.*, vol. 27, no. 3, pp. 283–285, Mar. 1978.

[19] P. V. Trifonov and S. V. Fedorenko, "A method for fast computation of the Fourier transform over a finite field," *Probl. Inf. Transm.*, vol. 39, no. 3, pp. 231–238, Jul. 2003. [Online]. Available:http://dcn.infos.ru/petert/papers/fftEng.pdf

[20] S. V. Fedorenko, "The discrete Fourier transform over the binary finite field," *IEEE Access*, vol. 11, pp. 62771–62779, Jun. 2023.

[21] 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.

[22] 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.

[23] 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.

[24] D. J. J. Versfeld, J. N. Ridley, H. C. Ferreira, and A. S. J. Helberg, "On systematic generator matrices for Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2549–2550, Jun. 2010.

[25] Y. Wang and X. Zhu, "A fast algorithm for Fourier transform over finite fields and its VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 3, pp. 572–577, Apr. 1988.
[26] D. G. Cantor, "On arithmetical algorithms over finite fields," *J. Combin. Theory, ser. A*, vol. 50, no. 2, pp. 285–300, Mar. 1989.
[27] 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.
[28] O. Ore, "On a special class of polynomials," *Trans. Amer. Math. Soc.*, vol. 35, no. 11, pp. 559–584, Nov. 1933.
[29] S.-J. Lin, W.-H. Chung, and Y. S. Han, "Novel polynomial basis and its application to Reed–Solomon erasure codes," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci. (FOCS)*, Philadelphia, PA, USA, Oct. 2014, pp. 316–325.
[30] 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.
[31] 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.
[32] N. Tang and Y. Lin, "Fast encoding and decoding algorithms for arbitrary (n, k) Reed–Solomon codes over $\mathbb{F}_{2^m}$," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 716–719, Apr. 2020.
[33] N. Tang and Y. S. Han, "A new decoding method for Reed–Solomon codes based on FFT and modular approach," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 7790–7801, Oct. 2022.
[34] C. Chen, Y. S. Han, N. Tang, X. Ma, and B. Bai, "Parallel Welch–Berlekamp algorithm," *IEEE Trans. Inf. Theory*, vol. 71, no. 5, pp. 3473–3488, May 2025.
[35] F. Didier, "Efficient erasure decoding of Reed–Solomon codes," *Computing Research Repository* - CORR, vol. abs/0901.1886, 2009.
[36] L. Yu, S.-J. Lin, H. Hou, and Z. Li, "Reed–Solomon coding algorithms based on Reed–Muller transform for any number of parities," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2677–2688, Sep. 2023.
[37] C. J. Hughes, *Single-Instruction Multiple-Data Execution*. Berlin, Germany: Springer Nature, 2022.
[38] I. O. S. T. Center, "Intel(R) intelligent storage acceleration library," 2016. [Online]. Available: https://github.com/01org/isa-l
[39] J. S. Plank and K. M. Greenan, "Jerasure: A library in C facilitating erasure coding for storage applications Version 2.0," Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. UT-EECS-14-721, 2014.
[40] C. A. Taylor, "Leopard-RS," [Online.] Available: https://github.com/catid/leopard
[41] C. Chen, "XD-RS," [Online.] Available: https://github.com/fastecc/xdrs
[42] C. Chen, S.-J. Lin, Z. Li, S. Cai, Y. S. Han, and B. Bai, "Reduced-complexity erasure decoding of low-rate Reed–Solomon codes based on LCH-FFT," in *Proc. IEEE Int. Symp. Inf. Theory*, 2023, pp. 1015–1019.
[43] C. R. P. Hartmann and L. D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inf. Theory*, vol. 22, no. 5, pp. 514–517, Sep. 1976.

Chao Chen received the Ph.D. degree in communication and information systems from Xidian University, China, in 2010. From 2010 to 2014, he was an Engineer with China Academy of Space Technology (CAST), Xi'an. From 2014 to 2015, he was a Postdoctoral Fellow with Institute of Network Coding (INC), The Chinese University of Hong Kong. He is currently an associate Professor with the State Key Laboratory of Integrated Services Networks (ISN), Xidian University, China. His research interests include channel coding and source coding.

Sian-Jheng Lin (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2004, 2006, and 2010, respectively. From 2010 to 2014, he was a Post-Doctoral Researcher with the Research Center for Information Technology Innovation, Academia Sinica. From 2014 to 2016, he was a Post-Doctoral Researcher with the Electrical Engineering Department, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. From 2016 to 2021, he was a Researcher with the School of Information Science and Technology, University of Science and Technology of China (USTC), Hefei, China. He is currently an Independent Researcher in China. In recent years, his research focuses on the codes for storage systems and data compressions.

Nianqi Tang received the Ph.D. degree in communication and information systems from Xidian University, China, in 2019. From 2019 to 2023, he was a Senior Engineer with Huawei Technologies Co., Ltd. Since 2023, he has been an assistant professor with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China. His research interests include error control coding, network coding, and information theory.

Yunghsiang S. Han (Fellow, IEEE) was born in Taipei, Taiwan, in 1962. He received the B.Sc. and M.Sc. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1984 and 1986, respec tively, and the Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, USA, in 1993. From 1986 to 1988, he was a Lecturer at the Ming-Hsin Engineering College, Hsinchu. He was a Teaching Assistant from 1989 to 1992, and a Research Associate with the School of Computer and Information Science, Syracuse University, from 1992 to 1993. From 1993 to 1997, he was an Associate Professor with the Department of Electronic Engineering, Hua Fan College of Humanities and Technology, Taipei Hsien, Taiwan. He was with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan, from 1997 to 2004. He was promoted to a Professor in 1998. He was a Visiting Scholar with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI, USA, from June 2001 to October 2001; the SUPRIA Visiting Research Scholar with the Department of Electrical Engineering and Computer Science and the CASECenter, Syracuse University, from September 2002 to January 2004 and from July 2012 to June 2013; and a Visiting Scholar with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA, from August 2008 to June 2009. He was with the Graduate Institute of Communication Engineering, National Taipei University, Taipei, from August 2004 to July 2010. From August 2010 to January 2017, he was a Chair Professor with the Department of Electrical Engineering, National Taiwan University of Science and Technology. He has been a Chair Professor at the National Taipei University since February 2015. From February 2017 to February 2021, he was with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, China. He is currently with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China. His research interests include error-control coding, wireless networks, and security. He was a Winner of the 1994 Syracuse University Doctoral Prize. One of his papers won the prestigious 2013 ACM CCS Test-of-Time Award in Cybersecurity.

Suihua Cai (Member, IEEE) received the B.Sc. degree in information and computing science from China University of Geosciences, Wuhan, China, in 2011, and the M.S. degree in fundamental mathematics and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2016 and 2019, respectively. He is currently an Associate Professor with Sun Yat-sen University. His research interests include information theory and channel coding theory, and their applications to communication systems.

Leilei Yu received the Ph.D. in Cyberspace Security from the University of Science and Technology of China (USTC), Hefei, China. From 2021 to 2022, he was an Assistant Research Fellow in cybersecurity at Purple Mountain Laboratories (PML), Nanjing, China. Since 2023, he has been with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC), where he has successively served as Assistant Research Fellow and Associate Research Fellow. His research interests include channel coding, storage erasure codes, and high-performance computing.

Zhongwei Li received the Ph.D. degree in computer science from The University of Texas at Arlington, USA, in 2019. From 2021 to 2023, he was a Postdoctoral Researcher with Huawei Technologies Co., Ltd. Since 2025, he has been an Associate Professor with the School of Software Technology, Dalian University of Technology, China. His research interests include error control coding, applied cryptography, and AI security.

Baoming Bai (Senior Member, IEEE) received the B.S. degree from the Northwest Telecommunications Engineering Institute, China, in 1987, and the M.S. and Ph.D. degrees in communication engineering from Xidian University, China, in 1990 and 2000, respectively. From 2000 to 2003, he was a Senior Research Assistant at the Department of Electronic Engineering, City University of Hong Kong. Since April 2003, he has been with the State Key Laboratory of Integrated Services Networks (ISN), School of Telecom munication Engineering, Xidian University, China, where he is currently a Professor. In 2005, he was with the University of California, Davis, CA, USA, as a Visiting Scholar. In 2018, he spent one month as a Senior Visiting Fellow at McMaster University, Ontario, Canada. Dr. Bai co-authored the book Channel Coding for 5G (in Chinese, 2020). His research interests include information theory and channel coding, wireless communication, and quantum communication. He received the Best Paper Award from the CIC/IEEE China Communications, in 2018.

Bo Bai (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2010. He was a Research Associate with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, from 2010 to 2012. From July 2012 to January 2017, he was an Assistant Professor with the Department of Electronic Engineering, Tsinghua University. He has obtained the support from Backbone Talents Supporting Project of Tsinghua University. He is currently an Information Scientist and the Director of the Theory Laboratory, Central Research Institute, 2012 Labs, Huawei Technologies Company Ltd., Hong Kong. His research interests include classical and post-Shannon information theory, semantic communications, B5G/6G mobile networks, and graph informatics.