On fast Fourier transform-based decoding of Reed-Solomon codes

Yunghsiang S. Han*

The State Key Laboratory of ISN,
Xidian University,
Xi'an, China
and
School of Electrical Engineering and Intelligentization,
Dongguan University of Technology,
Dongguan, China
Email: yunghsiangh@gmail.com
*Corresponding author

Chao Chen

The State Key Laboratory of ISN,
Xidian University,
Xi'an, China
Email: cchen@xidian.edu.cn

Sian-Jheng Lin

School of Information Science and Technology,
University of Science and Technology of China,
Hefei, China
Email: sjhenglin@gmail.com

Baoming Bai

The State Key Laboratory of ISN,
Xidian University,
Xi'an, China
Email: bmbai@mail.xidian.edu.cn

Abstract: Reed-Solomon (RS) codes are a popular class of codes that have been implemented in many practical systems. Recently, a fast approach to the error decoding of RS codes based on fast Fourier transform (FFT) was invented. In this work, we derive the key equation based on the Lagrange polynomial and then present erasure-and-error decoding of an (n, k) RS code. This decoding algorithm can simultaneously correct up to v errors and f erasures when $2v + f < n - k + 1$. The decoding complexity is with only $O(n \log n + (n - k) \log^2(n - k))$.

Keywords: coding; decoding; Reed-Solomon codes; fast Fourier transform; FFT.

Reference to this paper should be made as follows: Han, Y.S., Chen, C., Lin, S-J. and Bai, B. (xxxx) 'On fast Fourier transform-based decoding of Reed-Solomon codes', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Yunghsiang S. Han received his BSc and MSc in Electrical Engineering from the National Tsing Hua University, Taiwan, in 1984 and 1986, respectively, and PhD from the School of Computer and Information Science, Syracuse University, NY, in 1993. He is currently with the School of Electrical Engineering and Intelligentization at Dongguan University of Technology, China. He is also the Huashan Chair Professor at the Xidian University, and the Chair Professor at the National Taipei University. His research interests are in error-control coding, wireless networks, and security.

Chao Chen received his PhD from the Xidian University, Xi'an, China, in 2010. He is currently with the State Key Laboratory of Integrated Services Networks, School of Telecommunication Engineering, Xidian University. His research interests include information theory and channel coding.

Sian-Jheng Lin received his BSc, MSc and PhD in Computer Science from the National Chiao Tung University, Hsinchu, Taiwan, in 2004, 2006, and 2010, respectively. From 2010 to 2014, he was a Postdoc with the Research Center for Information Technology Innovation, Academia

Sinica. From 2014 to 2016, he was a Postdoc with the Electrical Engineering Department at the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. He was a part-time Lecturer at the Yuanpei University from 2007 to 2008, and at the Hsuan Chuang University from 2008 to 2010. He is currently a project researcher with the School of Information Science and Technology at the University of Science and Technology of China (USTC), Hefei, China.

Baoming Bai received his BS from the Northwest Telecommunications Engineering Institute, China, in 1987, and MS and PhD in Communication Engineering from the 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), 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 the McMaster University, Ontario, Canada. He 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.

1 Introduction

Reed-Solomon (RS) codes were invented in 1960 (Reed and Solomon, 1960). An (n,k) RS code is encoded over a finite field $\mathbf{F}_q$ for the length n equal to $q-1$ or q and information dimension k. RS codes are maximum distance separable (MDS), where the minimum Hamming distances of them are $n-k+1$. (n,k) RS codes can correct up to $\lfloor (n-k)/2 \rfloor$ erroneous symbols. The systematic (n,k) RS code adds $n-k$ redundancy symbols with the k information (message) symbols to form a codeword. RS codes have been applied to many important applications, including space exploration, barcodes, storage devices, digital broadcast system, and data transmission technologies. Recently, RS codes are adapted to construct other distributed storage codes such as regenerating codes (Rashmi et al., 2011). Since the real-world applications of RS codes involve erasure and error correction functions over characteristic-2 finite fields, the decoding complexity of RS codes has attracted a lot of attentions (Chen and Yan, 2008; Truong et al., 2006; Justesen, 1976; Gao, 2002).

Recently some fast approaches to the decoding of RS codes based on fast Fourier transform (FFT) or fast polynomial arithmetic techniques were invented (Justesen, 1976; Gao, 2002; Dianat et al., 2006; Motazedi and Dianat, 2018; Lin et al., 2014, 2016a, 2016b). Among these approaches, Lin et al. (2014, 2016a) showed a novel method to perform FFT over finite fields. The authors invented a new polynomial basis that is constructed by subspace polynomials over $\mathbf{F}_{2^m}$ to perform FFT. For a polynomial of degree less than n presented in this new basis, the n-point evaluations on this polynomial can be performed in $O(n\log n)$ field operations. That is, it is the first FFT on the coefficients of a polynomial with time complexity $O(n\log n)$ over all finite fields. Based on the FFT, encoding and erasure decoding algorithms for (n,k) RS codes were proposed with time complexity $O(n\log n)$. Later, an error-correction RS decoding algorithm with decoding complexity $O(n\log(n-k)+(n-k)\log^2(n-k))$ was provided in Lin et al. (2016b).

The original key equation derived in Lin et al. (2016b) was based on Chinese remainder theorem. In this work, we re-derive this key equation based on the Lagrange polynomial which is easier than the original derivation. We also further generalise the error decoding of RS codes presented in Lin et al. (2016b) to an erasure-and-error decoding. This decoding algorithm can simultaneously correct up to v errors and f erasures when $2v+f<n-k+1$, which is the best one can do. The decoding complexity is only $O(n\log n+(n-k)\log^2(n-k))$. This decoding complexity, to the best of the authors' knowledge, is the least so far.

The rest of this work is presented as follows. Section 2 reviews the FFT and the encoding algorithm of RS codes. Section 3 derives the key equation and Section 4 proposes the erasure-and-error decoding algorithm of RS codes. Section 5 concludes this work.

2 Brief review of the FFT and encoding of RS codes based on the FFT

Let $\mathbf{F}_{2^m}$ denote the finite field with size 2^m. Assume that $\mathbf{B}=(b_0,b_1,...,b_{m-1})$ is a basis of $\mathbf{F}_{2^m}$. A subspace V_k of $\mathbf{F}_{2^m}$ with dimension k is given as

$$V_k=\{i_0\cdot b_0+i_1\cdot b_1+...+i_{k-1}\cdot b_{k-1}|, \\ \forall i_j\in\{0,1\}\}, \tag{1}$$

where $\mathbf{B}_k=(b_0,b_1,...,b_{k-1})$ is a basis of subspace V_k, and $k\le m$. Let $\{\omega_0,\omega_1,...,\omega_{2^m-1}\}$ denote the 2^m elements of $\mathbf{F}_{2^m}$ and they are represented as

$$\omega_i=i_0\cdot b_0+i_1\cdot b_1+...+i_{m-1}\cdot b_{m-1},$$

where

$$i=i_0+i_1\cdot 2+...+i_{m-1}\cdot 2^{m-1},\ \forall i_j\in\{0,1\} \tag{2}$$

is the binary representation of i. The subspace polynomial (Ore, 1933; Cantor, 1989) of V_k with degree 2^k is defined as

$$f_k(x) = \prod_{\omega \in V_k} (x - \omega). \tag{3}$$

Next we present the basis designed in Lin et al. (2016b) based on the subspace polynomials. Let $\mathbb{X} = \{X_0(x), X_1(x), ..., X_{2^m-1}(x)\}$ denote a basis of polynomial ring $\mathbf{F}_{2^m}[x]/(x^{2^m} - x)$, where

$$X_i(x) = \prod_{j=0}^{m-1} (f_j(x))^{i_j}, \tag{4}$$

and $i_j \in \{0, 1\}$ is given in equation (2) that is the binary representation of i. Note that $\deg(X_i(x)) = i$, and thus the basis $\mathbb{X}$ can represent all polynomials in $\mathbf{F}_{2^m}[x]/(x^{2^m} - x)$.

A polynomial $\bar{P}_\ell(x)$ of degree $\ell - 1$ represented in the basis $\mathbb{X}$ is

$$\bar{P}_\ell(x) = \sum_{i=0}^{\ell-1} \bar{a}_i X_i(x), \tag{5}$$

where $\bar{a}_i \in \mathbf{F}_{2^m}$. Let $\bar{\mathbf{P}}_\ell = (\bar{a}_0, \bar{a}_1, ..., \bar{a}_{\ell-1})$ indicate the vector of the coefficients of $\bar{P}_\ell(x)$. Throughout this paper, in order to avoid confusion, we sometimes put the 'bar' on the top of a polynomial when it is represented in the basis $\mathbb{X}$. For any vector $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$, its polynomial representation is denoted as $v(x) = \sum_{i=0}^{n-1} v_i x^i$.

Let $g(V)$ denote a sequence of evaluation values of polynomial $g(x)$, where V is a set of elements from $\mathbf{F}_{2^m}$. Lin et al. (2016b) presented a recursive algorithm in $O(2^k \log 2^k)$ to compute $\bar{P}_{2^k}(V_k + \beta)$, where

$$V_k + \beta = \{\omega + \beta, \omega \in V_k\} \text{ for any } \beta \in \mathbf{F}_{2^m}.$$

The algorithm is an FFT on the coefficients of $g(x)$. We present it in Algorithm 1. Note that Algorithm 1 is performed under the basis $\bar{\mathbb{X}} = \{\bar{X}_0(x), \bar{X}_1(x), ..., \bar{X}_{2^m-1}(x)\}$, where $\bar{X}_i(x) = X_i(x)/p_i$ and $p_i = \prod_{j=0}^{m-1} (f_j(b_j))^{i_j}$, $0 \le i \le 2^m - 1$. p_i are pre-calculated constants to normalise the basis such that the $\text{FFT}_{\bar{\mathbb{X}}}$ has multiplication complexity constant 1/2. Hence, when we call $\text{FFT}_{\mathbb{X}}$ in this paper, we assume that the normalisation is performed before and after the $\text{FFT}_{\bar{\mathbb{X}}}$. It is the same for the inverse of FFT, $\text{IFFT}_{\mathbb{X}}$. The normalisation takes $O(2^k)$ and will be ignored since 2^k-point $\text{FFT}_{\bar{\mathbb{X}}}$ and $\text{IFFT}_{\bar{\mathbb{X}}}$ take $O(2^k \log 2^k)$.

An efficient encoder has been presented in Lin et al. (2016b) with complexity $O(n \log(n-k))$. Now consider an $(n = 2^m, k)$ RS code over $\mathbf{F}_{2^m}$ with $2^t = n - k$. Assume the information polynomial $\bar{u}(x)$ is presented in the basis $\mathbb{X}$ such that $\bar{u}(x) = \sum_{i=0}^{k-1} u_i X_i(x)$. Let

$$\mathbf{u} = (u_0, u_1, ..., u_{k-1}, \overbrace{\omega_0, \omega_0, ..., \omega_0}^{2^t}), \tag{6}$$

with 2^t ω_0s as the coefficients in the high degrees. Then the codeword $\mathbf{c}$ can be computed via $\text{FFT}_{\mathbb{X}}$ as follows. First, $\mathbf{c}$ is divided into a number of sub-vectors

$$\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, ..., \mathbf{c}_{n/2^t-1}), \tag{7}$$

where each $\mathbf{c}_i$ has 2^t elements and is defined as

$$\mathbf{c}_i = (c_{i \cdot 2^t}, c_{1+i \cdot 2^t}, ..., c_{2^t-1+i \cdot 2^t}) \ i = 0, 1, ..., n/2^t - 1.$$

We want $\mathbf{c}$ to have the message symbols in $\{\mathbf{c}_i\}_{i=1}^{n/2^t-1}$ and all parity symbols in $\mathbf{c}_0$. The parity $\mathbf{c}_0$ can be computed via

$$\begin{aligned} \mathbf{c}'_0 &= \text{IFFT}_{\mathbb{X}}(\mathbf{c}_1, t, \omega_{1 \cdot 2^t}) + \text{IFFT}_{\mathbb{X}}(\mathbf{c}_2, t, \omega_{2 \cdot 2^t}) + ... \\ &\quad + \text{IFFT}_{\mathbb{X}}(\mathbf{c}_{n/2^t-1}, t, \omega_k), \\ \mathbf{c}_0 &= \text{FFT}_{\mathbb{X}}(\mathbf{c}'_0, t, \omega_0). \end{aligned} \tag{8}$$

Algorithm 1 FFT in the basis $\bar{\mathbb{X}}$

Input: $\text{FFT}_{\bar{\mathbb{X}}}(\bar{\mathbf{P}}_{2^k}, k, \beta)$: $\bar{\mathbf{P}}_{2^k} = (\bar{a}_0, \bar{a}_1, ..., \bar{a}_{2^k-1})$, and $\beta \in \mathbf{F}_{2^m}$
Output: 2^k evaluations $\underline{\mathbf{P}}_{2^k} = (\underline{a}_0, \underline{a}_1, ..., \underline{a}_{2^k-1})$, where each $\underline{a}_i = \bar{P}_{2^k}(\omega_i + \beta)$

1 **if** $k = 0$ **then**
2 $\quad$ **return** $\bar{a}_0$
3 **for** $i = 0, ..., 2^{k-1} - 1$ **do**
4 $\quad$ $g_i^{(0)} \leftarrow \bar{a}_i + \frac{f_{k-1}(\beta)}{f_{k-1}(b_{k-1})} \bar{a}_{i+2^{k-1}}$
$\quad$ $g_i^{(1)} \leftarrow g_i^{(0)} + \bar{a}_{i+2^{k-1}}$
5 Call $D_0 \leftarrow \text{FFT}_{\bar{\mathbb{X}}}(\bar{\mathbf{P}}^{(0)}_{2^{k-1}}, k-1, \beta)$, where $\bar{\mathbf{P}}^{(0)}_{2^{k-1}} = (g_0^{(0)}, ..., g_{2^{k-1}-1}^{(0)})$ and $D_0 = (\underline{a}_0, ..., \underline{a}_{2^{k-1}-1})$
6 Call $D_1 \leftarrow \text{FFT}_{\bar{\mathbb{X}}}(\bar{\mathbf{P}}^{(1)}_{2^{k-1}}, k-1, v_{k-1} + \beta)$, where $\bar{\mathbf{P}}^{(1)}_{2^{k-1}} = (g_0^{(1)}, ..., g_{2^{k-1}-1}^{(1)})$ and $D_1 = (\underline{a}_{2^{k-1}}, ..., \underline{a}_{2^k-1})$
7 **return** $\underline{\mathbf{P}}_{2^k} = (\underline{a}_0, \underline{a}_1, ..., \underline{a}_{2^k-1})$

Source: Lin et al. (2016b)

3 Syndrome and key equation of the RS codes

In this section we discuss the syndromes of the traditional RS codes proposed by Reed and Solomon (1960) that perform encoding and decoding in monomial basis, 1, x, x^2, ..., x^{2^m-1}, and the RS codes proposed in Lin et al. (2016b) that perform encoding and decoding in $\mathbb{X}$.

Let the $(n = 2^m, k)$ RS code over finite field $\mathbf{F}_{2^m}$ be $\mathbf{C}$ and the codeword corresponding to the information sequence $u_0, u_1, ..., u_{k-1}$ be

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (\bar{u}(\omega_0), ..., \bar{u}(\omega_{n-1})),$$

where $\bar{u}(x) = \sum_{i=0}^{k-1} u_i X_i(x)$ and $n - k = 2^t$ for some integer t. After we expand $X_i(x)$ into monomial basis representation and substitute them into $\bar{u}(x)$, we have

$$\bar{u}(x) = \sum_{i=0}^{k-1} u_i X_i(x) = \sum_{i=0}^{k-1} \hat{u}_i x^i.$$

Let the $(n = 2^m, k)$ RS code over finite field $\mathbf{F}_{2^m}$ given in Reed and Solomon (1960) be $\hat{\mathbf{C}}$ such that the codeword corresponding to the information sequence $\hat{u}_0, \hat{u}_1, ..., \hat{u}_{k-1}$ be

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (\hat{u}(\omega_0), ..., \hat{u}(\omega_{n-1})),$$

where $\hat{u}(x) = \sum_{i=0}^{k-1} \hat{u}_i x^i$. Hence, the codeword in $\mathbf{C}$ corresponding to the information sequence $u_0, u_1, ..., u_{k-1}$ is the same as that in $\hat{\mathbf{C}}$ corresponding to the information sequence $\hat{u}_0, \hat{u}_1, ..., \hat{u}_{k-1}$. We then have the following lemma.

Lemma 1: Let the $(n = 2^m, k)$ RS code over finite field $\mathbf{F}_{2^m}$ be $\mathbf{C}$ and the codeword corresponding to the information sequence $u_0, u_1, ..., u_{k-1}$ be

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (\bar{u}(\omega_0), ..., \bar{u}(\omega_{n-1})),$$

where $\bar{u}(x) = \sum_{i=0}^{k-1} u_i X_i(x)$. Let the $(n = 2^m, k)$ RS code over finite field $\mathbf{F}_{2^m}$ given in Reed and Solomon (1960) be $\hat{\mathbf{C}}$ such that the codeword corresponding to the information sequence $\hat{u}_0, \hat{u}_1, ..., \hat{u}_{k-1}$ be

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (u(\omega_0), ..., (\omega_{n-1})),$$

where $\hat{u}(x) = \sum_{i=0}^{k-1} \hat{u}_i x^i$. Then $\mathbf{C} = \hat{\mathbf{C}}$.

By the above lemma, one can see that Lin et al. (2016b) proposed a clever way to encode the RS codes that employs the FFT proposed in Lin et al. (2016a). Assume that the transmitted codeword is

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (\bar{u}(\omega_0), ..., \bar{u}(\omega_{n-1})).$$

Since $\mathbf{C} = \hat{\mathbf{C}}$, by Moon (2005), we have

Lemma 2: For any $\mathbf{c} \in \mathbf{C}$, $\alpha, \alpha^2, ..., \alpha^{2p}$ are the roots of the codeword polynomial $c(x) = \sum_{i=0}^{n-1} c_i x^i$, where p is the error correction capability and $2p = n - k$.

According to Lemma 2, any RS code encoded by the FFT in $\mathbb{X}$ can still be decoded by traditional method such as Berlekamp-Massey algorithm based on syndrome calculated by $c(\alpha^i)$ for $1 \le i \le 2p$. However, when we represent $c(x)$ in $\mathbb{X}$, i.e., $c(x) = \sum_{i=0}^{n-1} c_i X_i(x)$, $\alpha, \alpha^2, ..., \alpha^{2p}$ are not necessary roots of $c(x)$. Hence, we cannot calculate syndrome in $\mathbb{X}$ as in monomial basis.

Next we present a way to calculate syndrome in $\mathbb{X}$ and derive the key equation by Lagrange polynomial instead of Chinese remainder theorem used in Lin et al. (2016b). Let the received vector $\mathbf{y} = (y_0, y_1, ..., y_{n-1})$ have v errors, where the errors are at $i_1, i_2, ..., i_v$. Assume that $v \le p$. Let

$$E_v = \{\omega_{i_\ell} \in \mathbf{F}_{2^m} | 1 \le \ell \le v\} \tag{9}$$

denote the set of ω_i corresponding to locations of errors. The error-locator polynomial is defined as

$$\lambda(x) = \prod_{\omega_i \in E_v} (x - \omega_i). \tag{10}$$

Let $\bar{r}(x)$ be the unique polynomial with degree at most $n - 1$ such that, for $0 \le i \le n - 1$, $\bar{r}(\omega_i) = y_i$. Note that $\bar{r}(x)$ can be obtained by applying inverse FFT (IFFT) to $\mathbf{y}$. If there are no errors, $\bar{r}(x)$ is the information polynomial with degree no more than $k - 1$. We then divide $\bar{r}(x)$ by $X_k(x)$ into

$$\bar{r}(x) = \bar{r}_L(x) + X_k(x)\bar{s}(x), \tag{11}$$

where $\bar{r}_L(x)$ is the remainder. Note that $\bar{s}(x)$ can be treated as a syndrome polynomial since $\bar{s}(x) = 0$ when there are no errors.

Let $\mathbf{y} = \mathbf{e} + \mathbf{c}$, where $\mathbf{e}$ is the error pattern vector. Then

$$\bar{r}(x) = e(x) + \bar{u}(x), \tag{12}$$

where $e(x) = \text{IFFT}(\mathbf{e})$ is the error polynomial represented by the monomial basis. Next we derive $e(x)$. Assume that $\mathbf{e} = (e_0, e_1, ..., e_{n-1})$. Hence, $e_i = 0$ and ω_i is the root of $e(x)$ for $\omega_i \notin E_v$. By Lagrange polynomial, we have

$$e(x) = \sum_{i=0}^{n-1} \frac{e_i}{A_i} \frac{f_m(x)}{(x - \omega_i)} = \sum_{\omega_i \in E_v} \frac{e_i}{A_i} \frac{f_m(x)}{(x - \omega_i)}, \tag{13}$$

where $e_i = e(\omega_i)$, $A_i = \prod_{j=0, j\neq i}^{n-1} (\omega_i - \omega_j) = -1 = 1$, and $f_m(x) = \prod_{j=0}^{2^m-1} (x - \omega_j)$. We then represent $f_m(x) = \hat{a}_m(x) + X_k(x) f_t(x)$, where the degree of $\hat{a}_m(x)$ is at most k (Lin et al., 2016b). Multiplying equation (13) by $\lambda(x)$, it becomes

$$\begin{aligned} e(x)\lambda(x) &= \sum_{\omega_i \in E_v} e_i \frac{\lambda(x)}{(x - \omega_i)} (\hat{a}_m(x) + X_k(x) f_t(x)) \\ &= p(x) + X_k(x) f_t(x) q(x), \end{aligned} \tag{14}$$

where $p(x) = \sum_{\omega_i \in E_v} e_i \frac{\lambda(x)}{(x - \omega_i)} \hat{a}_m(x)$ and $q(x) = \sum_{\omega_i \in E_v} e_i \frac{\lambda(x)}{(x - \omega_i)}$. The degree of $p(x)$ is less than $k + v$ and the degree of $q(x)$ is less than v.

Multiplying equation (12) by $\lambda(x)$ on both sides, we have

$$\begin{aligned} &\bar{r}_L(x)\lambda(x) + \lambda(x) X_k(x) \bar{s}(x) \\ &= p(x) + X_k(x) f_t(x) q(x) \\ &+ \lambda(x)\bar{u}(x). \end{aligned} \tag{15}$$

By dividing $X_k(x)$ and taking only quotient on both sides, we have

$$z(x) = \bar{s}(x)\lambda(x) + f_t(x) q(x), \tag{16}$$

where $\deg(z(x)) < v \le (n - k)/2$. Equation (16) can be treated as the key equation to find the error locator polynomial $\lambda(x)$ and $q(x)$. Next we show how to calculate the error values. The derivation is slightly different from that given in Lin et al. (2016b).

According to equation (14),

$$q(x) = \sum_{\omega_i \in E_v} e_i \frac{\lambda(x)}{x - \omega_i} = \sum_{\omega_i \in E_v} e_i \lambda_i(x), \tag{17}$$

where $\lambda_i(x) = \frac{\lambda(x)}{x - \omega_i}$. Note that $\lambda'(x) = \sum_{\omega_i \in E_v} \lambda_i(x)$ and $\lambda'(\omega_i) = \lambda_i(\omega_i)$ for $\omega_i \in E_v$. Hence,

$$\begin{aligned} &q(\omega_j) = \sum_{\omega_i \in E_v} e_i \lambda_i(\omega_j) = e_j \lambda'(w = \omega_j), \\ &\forall \omega_j \in E_v, \end{aligned} \tag{18}$$

which is equivalent to

$$e_j = e(\omega_j) = \frac{q(\omega_j)}{\lambda'(\omega_j)},\ \forall \omega_j \in E_v. \tag{19}$$

If **C** is a systematic code, $u_0, u_1, ..., u_{k-1}$ can be obtained by **c**. If it is not, $\bar{u}(x)$ can be obtained by applying IFFT to **c**. Note that equation (19) is similar to Forney's formula except the error-evaluator polynomial becomes $q(x)$.

4 Erasure-and-error decoding of RS codes

In this section, we present an erasure-and-error decoding algorithm for $(n = 2^m, k)$ RS codes over finite field $\mathbf{F}_{2^m}$, where $n - k = 2^t$ for some integer t. A new key equation based on equation (16) will be presented.

4.1 Key equation

Assume the transmitted codeword is

$$\mathbf{c} = (c_0, c_1, ..., c_{n-1}) = (\bar{u}(\omega_0), ..., \bar{u}(\omega_{n-1})).$$

Let the received vector $\mathbf{y} = (y_0, y_1, ..., y_{n-1})$ have v errors and f erasures, where the errors are at $i_1, i_2, ..., i_v$ and the erasures at $j_1, j_2, ..., j_f$. Note that $\{i_1, i_2, ..., i_v\} \cap \{j_1, j_2, ..., j_f\} = \emptyset$. Assume that $2v + f < n - k + 1$. Let

$$E_v = \{\omega_{i_\ell} \in \mathbf{F}_{2^m} | 1 \le \ell \le v\} \tag{20}$$

and

$$E_f = \{\omega_{j_\ell} \in \mathbf{F}_{2^m} | 1 \le \ell \le f\} \tag{21}$$

denote the set of ω_i corresponding to locations of errors and erasures, respectively. The error-locator polynomial is defined as

$$\lambda(x) = \prod_{\omega_i \in E_v} (x - \omega_i) \tag{22}$$

and the erasure-locator polynomial defined as

$$\gamma(x) = \prod_{\omega_i \in E_f} (x - \omega_i). \tag{23}$$

Note that, at the beginning of the decoding procedure, the erasure-locator polynomial $\gamma(x)$ can be computed since all erased positions are known.

Let $\mathbf{y} = \mathbf{e} + \mathbf{w} + \mathbf{c}$, where **e** is the error pattern vector and **w** is the erasure pattern vector. Then

$$\bar{r}(x) = e(x) + w(x) + \bar{u}(x), \tag{24}$$

where $e(x) = \text{IFFT}(\mathbf{e})$ is the error polynomial and $w(x) = \text{IFFT}(\mathbf{w})$ is the erasure polynomial, both represented by the monomial basis. Next we derive $e(x)$ and $w(x)$. Assume that $\mathbf{e} = (e_0, e_1, ..., e_{n-1})$ and $\mathbf{w} = (w_0, w_1, ..., w_{n-1})$. Hence, $e_i = 0$ and ω_i is a root of $e(x)$ for $\omega_i \notin E_v$. Also $w_i = 0$ and ω_i is a root of $w(x)$ for $\omega_i \notin E_f$. By Lagrange polynomial, we have

$$e(x) = \sum_{i=0}^{n-1} \frac{e_i}{A_i} \frac{f_m(x)}{(x - \omega_i)} = \sum_{\omega_i \in E_v} \frac{e_i}{A_i} \frac{f_m(x)}{(x - \omega_i)}, \tag{25}$$

where $e_i = e(\omega_i)$, $A_i = \prod_{j=0, j\neq i}^{n-1} (\omega_i - \omega_j) = -1 = 1$, and $f_m(x) = \prod_{j=0}^{2^m-1} (x - \omega_j)$. Similarly, by Lagrange polynomial, we have

$$w(x) = \sum_{\omega_i \in E_f} w_i \frac{f_m(x)}{(x - \omega_i)}, \tag{26}$$

where $w_i = w(\omega_i)$. Then we have

$$e(x) + w(x) = \left(\sum_{\omega_i \in E_v} \frac{e_i}{(x - \omega_i)} + \sum_{\omega_i \in E_f} \frac{w_i}{(x - \omega_i)} \right) f_m(x). \tag{27}$$

We represent $f_m(x) = \hat{a}_m(x) + X_k(x) f_t(x)$, where the degree of $\hat{a}_m(x)$ is at most k. Multiplying equation (27) by $\lambda(x)\gamma(x)$, it becomes

$$\begin{aligned} &(e(x) + w(x))\lambda(x)\gamma(x) \\ &= \left(\sum_{\omega_i \in E_v} e_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} + \sum_{\omega_i \in E_f} w_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} \right) \\ &\times (\hat{a}_m(x) + X_k(x) f_t(x)) \qquad (28) \\ &= p(x) + X_k(x) f_t(x) q(x), \qquad (29) \end{aligned}$$

where

$$p(x) = \left(\sum_{\omega_i \in E_v} e_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} + \sum_{\omega_i \in E_f} w_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} \right) \hat{a}_m(x)$$

and

$$q(x) = \left(\sum_{\omega_i \in E_v} e_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} + \sum_{\omega_i \in E_f} w_i \frac{\lambda(x)\gamma(x)}{(x - \omega_i)} \right).$$

The degree of $p(x)$ is less than $k + v + f$ and the degree of $q(x)$ is less than $v + f$.

Multiplying equation (24) by $\lambda(x)\gamma(x)$ on both sides, we have

$$\begin{aligned} &\bar{r}_L(x)\lambda(x)\gamma(x) + \lambda(x)\gamma(x) X_k(x) \bar{s}(x) \\ &= p(x) + X_k(x) f_t(x) q(x) + \lambda(x)\gamma(x)\bar{u}(x). \end{aligned} \tag{30}$$

By dividing $X_k(x)$ and taking only quotient on both sides of equation (30), we have

$$\bar{z}(x) = (\bar{s}(x)\gamma(x))\lambda(x) + q(x) f_t(x). \tag{31}$$

Note that $\deg(\bar{z}(x)) < v + f \le \frac{n-k+f}{2}$. Let

$$\bar{s}(x)\gamma(x) = \bar{q}^g(x) f_t(x) + \bar{s}^g(x), \tag{32}$$

where $\deg(\bar{s}^g(x)) < 2^t$. Then $\bar{s}^g(x)$ can be treated as a generalised syndrome polynomial. We now have a generalised key equation (GKE) as

$$\bar{z}(x) = \bar{s}^g(x)\lambda(x) + \tilde{q}(x) f_t(x), \tag{33}$$

where $\tilde{q}(x) = \bar{q}^g(x)\lambda(x) + q(x)$.

4.2 Decoding steps

We summarise the steps of erasure-and-error decoding as follows:

1 Calculate the generalised syndrome polynomial $\bar{s}^g(x)$.

2 Find the error-locator polynomial $\lambda(x)$ from GKE [equation (33)] by half-GCD algorithm given in Lin et al. (2016b).

3 Calculate the error locations E_v.

4 Determine the error values and the erasure values.

The detail of each step is given below.

Step 1

To calculate $\bar{s}^g(x)$ we first need to determine $\bar{s}(x)$. By equation (11), $\bar{s}(x)$ is the higher degree portion of $\bar{r}(x)$. $\bar{r}(x)$ can be obtained by performing $\text{IFFT}_{\mathbb{X}}$ on the received vector $\mathbf{y}$ and $\bar{s}(x)$ can be obtained by the encoding scheme given in equation (8) in Section 2, where set $\bar{c}(x) = \bar{r}(x)$ and $\bar{c}_0(x)$ is the desired $\bar{s}(x)$. This calculation takes $O(n \log n)$.

Note that $\bar{s}(x)$ is obtained in the basis $\mathbb{X}$ and $\gamma(x)$ is presented in the monomial basis. In order to perform multiplication of $\bar{s}(x)$ and $\gamma(x)$ to obtain $\bar{s}^g(x)$ by $\text{FFT}_{\mathbb{X}}$, we first need to find the values that $\gamma(x)$ is evaluated in ω_i for $\omega_i \in \mathbf{F}_{2^m}$, $0 \le i \le 2^t - 1$. This can be done by applying the fast Walsh-Hadamard transform (FWHT) proposed in Didier (2009). For completeness, we present this transform in Appendix. This evaluation takes time complexity $O(n \log n)$.

When there are 2^m points evaluated for $\gamma(x)$ and $\bar{s}(x)$ in the field, we can compute $\gamma(x)\bar{s}(x) \mod f_m(x)$ by $\text{FFT}_{\mathbb{X}}$ and $\text{IFFT}_{\mathbb{X}}$ as $\gamma(x)\bar{s}(x) \mod f_m(x) = \text{IFFT}_{\mathbb{X}}(\text{FFT}_{\mathbb{X}}(\gamma(x)) \otimes \text{FFT}_{\mathbb{X}}(\bar{s}(x)))$, where $\otimes$ is the component-wise multiplication (von zur Gathen and Gerhard, 2013). Since we only need $\bar{s}^g(x) = \gamma(x)\bar{s}(x) \mod f_t(x)$ in the GKE, next we prove that the multiplication can be performed in $\mathbf{F}_{2^m}$ with 2^t points evaluated.

Lemma 3: Let $\bar{s}^g(x) = \bar{s}(x)\gamma(x) \mod f_t(x)$. Then

$$\bar{s}^g(x) = \text{IFFT}_{\mathbb{X}}(\text{FFT}_{\mathbb{X}}(\gamma(x)) \otimes \text{FFT}_{\mathbb{X}}(\bar{s}(x))),$$

where $\text{FFT}_{\mathbb{X}}$ and $\text{IFFT}_{\mathbb{X}}$ are taken in the finite field $\mathbf{F}_{2^m}$ and there are only 2^t evaluated points involving in the transform.

Proof: It is well-known that $\gamma(x)\bar{s}(x) \mod f_m(x) = \text{IFFT}_{\mathbb{X}}(\text{FFT}_{\mathbb{X}}(\gamma(x)) \otimes \text{FFT}_{\mathbb{X}}(\bar{s}(x)))$, where $\text{FFT}_{\mathbb{X}}$ and $\text{IFFT}_{\mathbb{X}}$ are taken in the finite field $\mathbf{F}_{2^m}$ (von zur Gathen and Gerhard, 2013). Since $f_t(\omega_i) = 0$ for $0 \le \omega_i \le 2^t - 1$, we have $\bar{s}^g(\omega_i) = \bar{s}(\omega_i)\gamma(\omega_i)$ for $0 \le \omega_i \le 2^t - 1$ and they are the first 2^t components in $\text{FFT}_{\mathbb{X}}(\gamma(x)) \otimes \text{FFT}_{\mathbb{X}}(\bar{s}(x))$. Since $\deg \bar{s}^g(x) < 2^t$, it is uniquely defined by $\bar{s}^g(\omega_i)$ for $0 \le \omega_i \le 2^t - 1$. Hence, by $\text{IFFT}_{\mathbb{X}}$, we can recover $\bar{s}^g(x)$ as $\bar{s}^g(x) = \text{IFFT}_{\mathbb{X}}(\text{FFT}_{\mathbb{X}}(\gamma(x)) \otimes \text{FFT}_{\mathbb{X}}(\bar{s}(x)))$. Furthermore, $2^t = n - k < n = 2^m$ such that all operations mentioned above can be performed in $\mathbf{F}_{2^m}$.

The computation complexity of the $\text{IFFT}_{\mathbb{X}}$ and the $\text{IFFT}_{\mathbb{X}}$ are $O((n-k)\log(n-k))$ due to the fact that $\deg(\bar{s}(x))$, $\deg(\gamma(x))$ and $\deg(\bar{s}^g(x))$ are no more than $n - k$. Hence, the overall complexity to obtain $\bar{s}^g(x)$ in this step is $O(n \log n)$.

Step 2

This step is similar to the process to obtain $\lambda(x)$ given in Lin et al. (2016b). We briefly summarise it as follows. Fist we divide $f_t(x)$ by $\bar{s}^g(x)$ to obtain the remainder $\bar{r}_t(x)$ and the quotient $\bar{q}_t(x)$. Then apply the half-GCD algorithm (Algorithm 5) presented in Lin et al. (2016b) with inputs $\bar{s}^g(x)$, $\bar{r}_t(x)$, and t to obtain two matrices

$$\left(\begin{bmatrix} \bar{w}_0(x) \\ \bar{w}_1(x) \end{bmatrix}, \begin{bmatrix} \bar{u}_0(x) & \bar{v}_0(x) \\ \bar{u}_1(x) & \bar{v}_1(x) \end{bmatrix} \right). \tag{34}$$

By the above matrices,

$$\bar{w}_1(x) = \bar{u}_1(x)\bar{s}^g(x) + \bar{v}_1(x)\bar{r}_t(x) \tag{35}$$

which is equivalent to

$$\bar{w}_1(x) = (\bar{u}_1(x) - \bar{v}_1(x)q_t(x))\bar{s}^g(x) + \bar{v}_1(x)f_t(x). \tag{36}$$

The error-locator polynomial then becomes

$$\bar{\lambda}(x) = \bar{u}_1(x) - \bar{v}_1(x)\bar{q}_t(x),$$

where $\bar{\lambda}(x)$ is $\lambda(x)$ represented in the basis $\mathbb{X}$. The overall time complexity of this step is $O((n-k)\log^2(n-k))$.

Step 3

The error locations are the roots of $\bar{\lambda}(x)$. They can be found via $\text{FFT}_{\mathbb{X}}$. The overall time complexity in this step is $O(n \log(n-k))$.

Step 4

In this step, we need to determine the error values and the erasure values. According to equation (28),

$$\begin{aligned} q(x) &= \sum_{\omega_i \in E_v} e_i \frac{\lambda(x)\gamma(x)}{(x-\omega_i)} + \sum_{\omega_j \in E_f} w_j \frac{\lambda(x)\gamma(x)}{(x-\omega_j)} \\ &= \sum_{\omega_i \in E_v} e_i \lambda_i(x)\gamma(x) + \sum_{\omega_j \in E_f} w_j \lambda(x)\gamma_j(x), \end{aligned} \tag{37}$$

where $\lambda_i(x) = \frac{\lambda(x)}{x-\omega_i}$ and $\gamma_j(x) = \frac{\gamma(x)}{x-\omega_j}$. Note that $\lambda'(x) = \sum_{\omega_i \in E_v} \lambda_i(x)$ and $\lambda'(\omega_i) = \lambda_i(\omega_i)$ for $\omega_i \in E_v$. Hence,

$$\begin{aligned} q(\omega_\ell) &= \sum_{\omega_i \in E_v} e_i \lambda_i(\omega_\ell)\gamma(\omega_\ell) \\ &\quad + \sum_{\omega_j \in E_f} w_j \lambda(\omega_\ell)\gamma_j(\omega_\ell) \\ &= e_\ell \lambda'(\omega_\ell)\gamma(\omega_\ell), \ \forall \omega_\ell \in E_v, \end{aligned} \tag{38}$$

which is equivalent to

$$e_\ell = e(\omega_\ell) = \frac{q(\omega_\ell)}{\lambda'(\omega_\ell)\gamma(\omega_\ell)},\ \forall \omega_\ell \in E_v. \tag{39}$$

Similarly, $\gamma'(x) = \sum_{\omega_j \in E_f} \gamma_j(x)$ and $\gamma'(\omega_j) = \gamma_j(\omega_j)$ for $\omega_j \in E_f$. Hence,

$$\begin{aligned} q(\omega_\ell) &= \sum_{\omega_i \in E_v} e_i \lambda_i(\omega_\ell)\gamma(\omega_\ell) \\ &\quad + \sum_{\omega_j \in E_f} w_j \lambda(\omega_\ell)\gamma_j(\omega_\ell) \\ &= w_\ell \lambda(\omega_\ell)\gamma'(\omega_\ell),\ \forall \omega_\ell \in E_f, \end{aligned} \tag{40}$$

which is equivalent to

$$w_\ell = w(\omega_\ell) = \frac{q(\omega_\ell)}{\lambda(\omega_\ell)\gamma'(\omega_\ell)},\ \forall \omega_\ell \in E_f. \tag{41}$$

The above calculation requires determining $q(\omega_i)$, $\forall \omega_i \in E_v \cup E_f$. Next we give a way to calculate them. Recall that $q(x) = \tilde{q}(x) - \bar{q}^g(x)\lambda(x)$. By equation (35), $\tilde{q}(x) = \bar{v}_1(x)$ and, by equation (32), $f_t(x)\bar{q}^g(x) = \bar{s}(x)\gamma(x) - \bar{s}^g(x)$. When $\omega_i \in E_v$, $q(\omega_i) = \tilde{q}(\omega_i) = \bar{v}_1(\omega_i)$. When $\omega_i \in E_f$ and $f_t(\omega_i) \neq 0$, we have

$$q(\omega_i) = \tilde{q}(\omega_i) - \frac{\bar{s}(\omega)\lambda(\omega_i)\gamma(\omega_i) - \bar{s}^g(\omega_i)\lambda(\omega_i)}{f_t(\omega_i)} \tag{42}$$

$$= \bar{v}_1(\omega_i) - \frac{\bar{s}(\omega)\lambda(\omega_i)\gamma(\omega_i) - \bar{s}^g(\omega_i)\lambda(\omega_i)}{f_t(\omega_i)}. \tag{43}$$

When $f_t(\omega_i) = 0$, more effort is needed. By taking formal derivative on both sides of equation (32), we have

$$\begin{aligned} \bar{q}^g(x) + f_t(x)(\bar{q}^g)'(x) &= \bar{s}'(x)\gamma(x) + \bar{s}(x)\gamma'(x) \\ &\quad - (\bar{s}^g)'(x). \end{aligned}$$

Substituting ω_i into the above equation, we get

$$\bar{q}^g(\omega_i) = \bar{s}(\omega_i)\gamma'(\omega_i) - (\bar{s}^g)'(\omega_i).$$

Hence, when $\omega_i \in E_f$ and $s_t(\omega_i) = 0$, we have

$$\begin{aligned} q(\omega_i) &= \bar{v}_1(\omega_i) - \bar{s}(\omega_i)\lambda(\omega_i)\gamma'(\omega_i) \\ &\quad - (\bar{s}^g)'(\omega_i)\lambda(\omega_i). \end{aligned} \tag{44}$$

In this step, several $\text{FFT}_{\mathbb{X}}$ and $\text{IFFT}_{\mathbb{X}}$ are involved and their time complexity are $O((n-k)\log(n-k))$. Furthermore, the formal derivative of $\lambda(x)$ (indeed $\bar{\lambda}(x)$) is performed and its time complexity is $O(n \log n)$ (Lin et al., 2016a). The evaluation of the formal derivative of $\gamma(x)$ on $w_i \in E_f$ can be obtained by the FWHT given in Appendix. Hence, the overall complexity of this step is $O(n \log n)$.

In summary, the overall time complexity of the erasure-and-error decoding is $O(n \log n) + O((n-k)\log^2(n-k))$.

5 Conclusions

In this work, we present a fast erasure-and-error decoding algorithm for RS codes. This algorithm is based on a newly invented FFT over finite fields. The time complexity of the erasure-and-error decoding for an (n, k) RS code is reduced to $O(n \log n)$+$O((n-k)\log^2(n-k))$. An interesting future work is to design an erasure-and-error decoding algorithm of RS codes such that when no error occurs, the complexity of it can reduce to that of the erasure decoding proposed in Lin et al. (2014).

References

Cantor, D.G. (1989) 'On arithmetical algorithms over finite fields', *Journal of Combinatorial Theory, Series A*, Vol. 50, No. 2, pp.285–300.

Chen, N. and Yan, Z. (2008) 'Complexity analysis of Reed-Solomon decoding over $GF(2^m)$ without using syndromes', *EURASIP J. Wirel. Commun. Netw.*, 11 January, Vol. 16, pp.1–16.

Dianat, R., Marvasti, F. and Ghanbari, M. (2006) 'Reliable video transmission using codes close to the channel capacity', *IEEE Trans. Cir. Syst. Video Tech.*, December, Vol. 16, No. 12, pp.1550–1556.

Didier, F. (2009) 'Efficient erasure decoding of Reed-Solomon codes', *CoRR*, DOI: abs/0901.1886.

Fino, B.J. and Algazi, V. (1976) 'Unified matrix treatment of the fast Walsh-Hadamard transform', *IEEE Transactions on Computers*, November, Vol. C-25, No. 11, pp.1142–1146.

Gao, S. (2002) 'A new algorithm for decoding Reed-Solomon codes', in *Communications, Information and Network Security*, pp.55–68, Kluwer, New York.

Gibbs, J.E. and Pichler, F.R. (1971) 'Comments on transformation of 'Fourier' power spectra into 'Walsh' power spectra', *IEEE Transactions on Electromagnetic Compatibility*, August, Vol. EMC-13, No. 3, pp.51–54.

Justesen, J. (1976) 'On the complexity of decoding Reed-Solomon codes (corresp.)', *IEEE Transactions on Information Theory*, March, Vol. 22, No. 2, pp.237–238.

Lin, S-J., Chung, W-H. and Han, Y.S. (2014) 'Novel polynomial basis and its application to Reed-Solomon erasure codes', in *The 55th Annual Symposium on Foundations of Computer Science (FOCS'14)*.

Lin, S-J., Al-Naffouri, T.Y., Han, Y.S. and Chung, W-H. (2016a) 'Novel polynomial basis with fast Fourier transform and its application to Reed-Solomon erasure codes', *IEEE Transactions on Information Theory*, November, Vol. 62, No. 11, pp.6284–6299.

Lin, S.J., Al-Naffouri, T.Y. and Han, Y.S. (2016b) 'FFT algorithm for binary extension finite fields and its application to Reed-Solomon codes', *IEEE Transactions on Information Theory*, October, Vol. 62, No. 10, pp.5343–5358.

Moon, T.K. (2005) *Error Correction Coding: Mathematical Methods and Algorithms*, John Wiley & Sons, Inc., Hoboken, NJ.

Motazedi, M.R. and Dianat, R. (2018) 'An erasure-based scheme for reduction of PAPR in spatial multiplexing MIMO-OFDM using Reed-Solomon codes over $GF(2^{16}+1)$', *Int. J. Electron.*, September, Vol. 105, No. 9, pp.1583–1597.

Ore, O. (1933) 'On a special class of polynomials', *Trans. Amer. Math. Soc.*, November, Vol. 35, No. 11, pp.559–584.

Rashmi, K.V., Shah, N.B. and Kumar, P.V. (2011) 'Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction', *IEEE Transactions on Information Theory*, August, Vol. 57, No. 8. pp.5227–5239.

Reed, I.S. and Solomon, G. (1960) 'Polynomial codes over certain finite fields', *Journal of the Society for Industrial and Applied Mathematics*, Vol. 8, No. 2, pp.300–304.

Robinson, G. (1972) 'Logical convolution and discrete Walsh and Fourier power spectra', *IEEE Transactions on Audio and Electroacoustics*, October, Vol. 20, No. 4, pp.271–280.

Truong, T.K., Chen, P.D., Wang, L.J. and Cheng, T.C. (2006) 'Fast transform for decoding both errors and erasures of Reed-Solomon codes over $GF(2^m)$ for $8 \leq m \leq 10$', *IEEE Transactions on Communications*, February, Vol. 54, No. 2, pp.181–186.

von zur Gathen, J. and Gerhard, J. (2013) 'Fast multiplication', in *Modern Computer Algebra*, 3rd ed., pp.221–256, Cambridge University Press, Cambridge, England.

Appendix

Evaluating erasure-locator polynomials with FWHTs

Since the operating finite field is $\mathbf{F}_{2^m}$, we have $\gamma(x) = \prod_{i \in E_f}(x + \omega_i)$. By substituting $x = \omega_\ell \in \mathbf{F}_{2^m} \setminus E_f$ into $\Pi(x)$, we have

$$\begin{aligned}\Pi(\omega_\ell) &= \prod_{\omega_i \in E_f, \omega_i \neq \omega_\ell} (\omega_\ell + \omega_i) \\ &= \prod_{\omega \in \mathbf{F}_{2^m}, \omega_\ell \neq \omega} (\omega_\ell + \omega)^{R_\omega}, \end{aligned} \tag{45}$$

where $\{R_\omega | \omega \in \mathbf{F}_{2^m}\}$ is defined as

$$R_\omega = \begin{cases} 1 \text{ if } \omega \in E_f; \\ 0 \text{ otherwise.} \end{cases} \tag{46}$$

Let $\mathrm{Log}(x)$ denote the discrete logarithm function of $\mathbf{F}^*_{2^m}$, where $\mathbf{F}^*_{2^m} = \mathbf{F}_{2^m} \setminus \{0\}$. That is, for each $\omega \in \mathbf{F}^*_{2^m}$, we have $\mathrm{Log}(\omega) = j$ iff $\omega = \alpha^j$, where α is a primitive element of $\mathbf{F}^*_{2^m}$. Then equation (45) can be rewritten as

$$\begin{aligned}&\mathrm{Log}(\Pi(\omega_\ell)) = \biguplus_{\omega \in \mathbf{F}_{2^m}, \omega \neq \omega_\ell} R_\omega \mathrm{Log}(\omega_\ell + \omega), \\ &\forall \omega_\ell \in \mathbf{F}_{2^m} \setminus E_f,\end{aligned}$$

where $\biguplus$ is the summation with normal additions, rather than the additions in the finite field. By letting $\mathrm{Log}(0) = 0$, the above equation can be rewritten as

$$\begin{aligned}&\mathrm{Log}(\Pi(\omega_\ell)) = \biguplus_{\omega \in \mathbf{F}_{2^m}} R_\omega \mathrm{Log}(\omega_\ell + \omega), \\ &\forall \omega_\ell \in \mathbf{F}_{2^m} \setminus E_f.\end{aligned} \tag{47}$$

In equation (47), $+$ is the addition in $\mathbb{F}_{2^m}$ and it can be treated as exclusive-or operation. Hence, equation (47) is the logical convolution (Gibbs and Pichler, 1971; Robinson, 1972) that can be efficiently computed by FWHTs (Fino and Algazi, 1976). The steps of the algorithm are given as follows.

Let $\mathrm{FWT}(\bullet)$ denote the h-point FWHT. An h-point FWHT requires $h \lg(h)$ additions. Define

$$\tilde{R} = (R_{\omega_0}, R_{\omega_1}, ..., R_{\omega_{2^m-1}}),$$

$$\tilde{L} = (0, \mathrm{Log}(\omega_1), \mathrm{Log}(\omega_2), ..., \mathrm{Log}(\omega_{2^m-1})).$$

Equation (47) can be computed by

$$R_{\mathrm{w}} = \mathrm{FWT}(\mathrm{FWT}(\tilde{R}) \times \mathrm{FWT}(\tilde{L})), \tag{48}$$

where $\times$ denotes pairwise integer multiplication. Note that $\mathrm{FWT}(\tilde{L})$ can be precomputed, and equation (48) can be computed with performing two fast Walsh transforms of length 2^m. Since R_{w} is the logarithm of the desired values, the exponent for each element is computed. Hence, the computation complexity requires $\mathcal{O}(2^m \lg(2^m))$ modulus additions, $\mathcal{O}(2^m)$ modulus multiplications, and $\mathcal{O}(2^m)$ exponentiations for $2^m = n$.