A Modified Interpolation Algorithm in List Decoding of Generalized Reed-Solomon Codes

Ta-Hsiang Hu and Trong Nghia Nguyen

Department of Electrical Engineering, Da-Yeh University

ABSTRACT

This work presents a modified interpolation algorithm in list decoding of generalized Reed-Solomon (GRS) codes. A list decoding algorithm of generalized Reed-Solomon codes has two steps, interpolation and factorization. The extended key equation (EKE) algorithm proposed in [13] is an interpolation-based approach with a lower complexity than Sudan's algorithm[10]. A limitation in such an EKE algorithm is only suitable for a GRS code with code rate $R \le 1/3$. To overcome such a limitation and increase the decoding speed, this work presents a modified EKE algorithm for a GRS code with any code rate. This proposed EKE algorithm needs less complexity than the original EKE algorithm does, as GRS codes of code length 255 and $0.2 \le R \le 1/3$ are employed.

Keywords: Reed-Solomon decoding, list decoding, extended key equation algorithm

廣義 Reed-Solomon 碼的改進式插值列表解碼演算法

胡大湘*　阮仲義

大葉大學電機工程學系

摘　　要

自這項研究提出廣義 Reed-Solomon 碼的改進式插值列表解碼演算法。廣義 Reed-Solomon 列表解碼演算法有兩個步驟，插值和分解。相較於 Sudan 演算法[10]，此研究所提出的擴展關鍵方程式（EKE）[13]演算法是一種複雜性較低的插值方法。原本 EKE 演算法僅適合於碼率小於 1/3。為了克服這種限制，提高解碼速度，這項研究提出修改版的 EKE 演算法。當廣義 Reed-Solomon 碼長度為 255 和碼率介於 0.2 與 1/3 之間時，這項研究成果，比原來 EKE 演算法需要更少的複雜性。

關鍵詞： Reed-Solomon解碼，列表解碼，擴展關鍵方程式演算法

文稿收件日期 101.6.13;文稿修正後接受日期 101.10.22; *通訊作者
Manuscript received June 13, 2012; revised Oct 22, 2012; * Corresponding author

Ⅰ. INTRODUCTION

Reed-Solomon (RS) codes are currently used in a wide variety of applications, ranging from data storage systems, mobile communications to satellite communications. The third-generation (3G) wireless standard utilizes RS codes as outer codes. For CDMA2000 high-rate broadcast packet data air interface [1], they are expected to be adopted as outer codes in concatenated coding schemes for future fourth-generation (4G) wireless systems.

Algorithms for hard decision decoding of RS codes are typically classified into two well-known types, namely syndrome-based decoding and interpolation-based decoding. Well-developed algorithms in the first category include the Peterson-Gorenstein-Zierler algorithm [3], Berlekamp-Massey algorithm [2][3], Euclidean algorithm [2][3], frequency domain algorithm [2][3], step-by-step algorithm [4]-[7]. Algorithms in the second category include the Welch-Berlekamp algorithms [8][9] and list decoding algorithms [10][11][13], as Koetter-Vardy algorithm [12] is also a list decoding algorithm but with soft decision approaching.

Sudan's algorithm [10] decodes GRS codes in two steps involved, namely interpolation and factorization. An interpolation is performed on a received word $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$, producing a nonzero bivariate polynomial, $Q(x,y) = \sum_{t=0}^{l} Q^{(t)}(x) y^t = \sum_{i=0}^{n} a_i \phi(x,y)$, with at least $n - \tau$ points (α^i, v_i). The interpolation equations are $Q(\alpha^i, v_i) = 0$, and $i \in [n-1] = \{0,1,\cdots,n-1\}$. Factorization is then performed on $Q(x,y)$, yielding linear factors (or called *y*-root polynomials) $y - \hat{f}(x)$. The codewords are then generated from these distinct factors $\hat{f}(x)$ via an evaluation mapping. A decoded codeword $\hat{\mathbf{c}}$ is chosen if the Hamming distance between $\mathbf{c}$ and $\mathbf{v}$ is τ or less.

Because solving these interpolation equations of Sudan's algorithm with a naïve Gaussian elimination requires the time complexity O(n^3), an EKE algorithm has been presented to decrease this complexity [13]. The EKE algorithm employs generalized Berlekamp-Massy algorithm that obtains the shortest recurrence that generates a given sequence, and the time complexity of EKE to solve these interpolation equations is $O(l(n-k)^2)$. l represents a design parameter, typically a small constant, which is an upper bound on the size of the list of decoded codewords.

Guruswami and Sudan (GS) presented an improvement on Sudan's algorithm [11], by introducing a multiplicity *m* at each interpolation point. A nonzero $Q(x,y)$ polynomial exists that interpolates the points (x_i, y_i), $i \in [n-1]$ with multiplicity *m*, and is formed by $Q(x,y) = \sum_{i=0}^{C} a_i \phi(x,y)$, $c = n\binom{m+1}{2}$. In comparison with Sudan's work, the GS algorithm provides more $n\binom{m+1}{2} - n$ linear homogeneous equations in interpolation, thus improving the decoding correction distance. Increasing *m* improves the decoding performance, but also increases the required complexity. The asymptotical decoding correction fraction is given by $1 - \sqrt{R}$, and the code rate *R* is given by $R = k/n$. The increase in decoding capability is substantial, especially for low rate GRS codes.

Koetter and Vardy [12] extended the GS algorithm by incorporating the soft information received from a channel into the interpolation process. With a complexity that is polynomial in the code length, the Koetter–Vardy (KV) algorithm can achieve a substantial coding gain over the GS algorithm. For instance, at a frame-error-rate (FER) of 10^{-5}, the KV algorithm can achieve a coding gain of about 1dB over the GS algorithm, for a (255,144) GRS code transmitted over an additive white Gaussian noise (AWGN) channel using 256-QAM modulation [12].

Those approaches commonly have a drawback, which codeword-checking (or syndrome computation) is absent during their decoding processes. In other words, regardless of whether the received sequence is correct or not, the decoding algorithm proceeds to decode

it. This work overcomes this drawback by presenting a modified EKE algorithm with codeword checking. The EKE algorithm [13] is proposed only for the decoding rate of no larger than 1/3. In overcoming such a limitation, based on ordering of the *y*-power in the polynomial $Q(x, y)$, the work proposes a way suitably to decode a GRS with any code rate.

The rest of this paper is organized as follows. Section II introduces the EKE algorithm. Section III presents the modified EKE algorithm based on syndrome computations and matrix operations to obtain the transmission information from the received codeword. Additionally, some examples are given to illustrate the benefit of the modified EKE algorithm. Section IV displays the complexity analysis of these two EKE algorithms. Finally, conclusions are presented in Section V.

Ⅱ. EXTENDED KEY EQUATION ALGORITHM

At the end of Consider an evaluation mapping $f(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$ and $n = 2^m - 1$. A codeword in an (*n*, *k*) GRS code over GF(2^m) is generated as

$$\begin{aligned} \mathbf{c} &= (c_0, c_1, \cdots, c_{n-1}) \\ &= (f(\alpha^0), f(\alpha), \cdots f(\alpha^{n-1})) \\ &= \mathbf{m} \cdot \mathbf{G} \end{aligned} \quad (1)$$

where the information vector $\mathbf{m} = (m_0, m_1, \cdots, m_{k-1})$, and the generator matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ & & \vdots & \\ 1 & \alpha^{k-1} & \cdots & \alpha^{(n-1)(k-1)} \end{bmatrix} \quad (2)$$

and α is a primitive element in GF(2^m).

The term l is the upper bound of the number of consistent codewords, which are at Hamming distance $\le \tau$ from any received word. For an (*n*, *k*) GRS code, Sudan's algorithm corrects any error pattern of up to τ errors for

$$\tau = n - (m+1) - l(k-1) \quad (3)$$

where *m* denotes the smallest nonnegative integer holding the following equation.

$$(m+1)(l+1) + (k-1)\binom{l+1}{2} > n \quad (4)$$

Assuming that $k \le (n+1)/3$, the value of τ becomes

$$\tau = \lfloor 2(n+1)/3 \rfloor - k \quad (5)$$

Let *F* be a field and $F_k[x]$ represent the set of all polynomials of degree < *k* in the variable *x* over *F*. Sudan's algorithm consists of the following steps:

1. Find a nonzero bivariate polynomial $Q(x, y)$ over *F* with at least $n - \tau$ points (α^i, v_i), interpolation equations $Q(\alpha^i, v_i) = 0$, and $i \in [n-1]$, for a received word $\mathbf{v} = (v_0, v_1, \cdots v_{n-1})$.
2. Output all polynomials $f(x) \in F_k[x]$ for which $y - f(x)$ is a factor of $Q(x, y)$ and $f(\alpha^i) = v_i$ for at least $n - \tau$ locators α^i.

In [2][3], for an (*n*, *k*) RS code, the error-locator polynomial $\Lambda(x)$ and the error-evaluator polynomial $\Omega(x)$ are computed in the following key equation (KE)

$$\Lambda(x) \cdot S(x) \equiv \Omega(x) \pmod{x^{n-k}} \quad (6)$$

In [13], based on the linear factors of bivariate polynomials $Q(x, y)$ where the polynomial arithmetic is carried out modulo a power of *x* in Sudan's algorithm, an EKE algorithm is derived as follows:

$$\sum_{t=1}^{l} \Lambda^{(t)}(x) \cdot x^{(t-1)(k-1)} \cdot S^{(t)}(x) \equiv \Omega(x) \pmod{x^{n-k}} \quad (7)$$

where $\Lambda^{(t)}(x)$, $t \in \{1,2,\cdots l\}$, and $\Omega(x)$ are polynomials that satisfy certain degree constraints, and $S^{(t)}(x)$ are syndrome polynomials as follows:

$$S^{(t)}(x) = \sum_{i=0}^{n-2-t(k-1)} S_i^{(t)} x^i,$$

$$S_i^{(t)} = \sum_{j=0}^{n-1} v_j^t \eta_j \alpha^{i \cdot j}, \quad (8)$$

and

$$\eta_j^{-1} = \prod_{r \in [n-1] \setminus \{j\}} (\alpha^j - \alpha^r) \quad (9)$$

Furthermore, the above equation can be obtained as follows:

$$\sum_{t=1}^{l} \sum_{j=0}^{N_t-1} Q_j^{(t)} S_{i+j}^{(t)} = 0, \quad 0 \le i < \tau,$$

$$N_t = n - \tau - t(k-1) \quad (10)$$

which denoted as

$$\begin{pmatrix} S_0^{(1)} & \cdots & S_{N_1-1}^{(1)} & S_0^{(2)} & \cdots & S_{N_l-1}^{(l)} \\ S_1^{(1)} & \cdots & S_{N_1}^{(1)} & S_1^{(2)} & \cdots & S_{N_l}^{(l)} \\ & & & \vdots & & \\ S_{\tau-1}^{(1)} & \cdots & S_{N_1+\tau-2}^{(1)} & S_{\tau-1}^{(2)} & \cdots & S_{N_l+\tau-2}^{(l)} \end{pmatrix} \bullet \begin{pmatrix} Q_0^{(1)} \\ \vdots \\ Q_{N_1-1}^{(1)} \\ Q_0^{(2)} \\ \vdots \\ Q_{N_l-1}^{(l)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad (11)$$

and

$$Q^{(t)}(x) = \sum_{j=0}^{N_t-1} Q_j^{(t)} x^j, \text{ and } \quad t \in \{1,2,\cdots l\} \quad (12)$$

After these polynomials $Q^{(t)}(x)$, $t \in \{1,2,\cdots,l\}$, have been computed in (11) by using the Generalized Berlekamp-Massey (GBM) algorithm[14] or a similar algorithm mentioned in [13], the polynomial $Q^{(0)}(x)$ is obtained as follows:

$$Q^{(0)}(x) + \sum_{t=1}^{l} Q^{(t)}(x) y^t = Q(x,y) \quad (13)$$

and satisfies

$$Q^{(0)}(\alpha^j) = -\sum_{t=1}^{l} Q^{(t)}(\alpha^j) v_j^t \text{ and } j \in [n-1] \quad (14)$$

Ⅲ. MODOFIED EXTENDED KEY EQUATION ALGORITHM

Since the polynomial $f(x) = \sum_{i=0}^{n-1} m_i x^i$ is associated with a codeword $\mathbf{c} \in C$, which has zeros $1, \alpha, \alpha^2, \cdots, \alpha^{n-k}$ [15], a parity-check matrix for C is given by[16][17],

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ & & \vdots & \\ 1 & \alpha^{n-k} & \cdots & \alpha^{(n-k)(n-1)} \end{bmatrix} \quad (15)$$

Theorem 1: Let l_x and L be the highest x-power and y-power in a bivariate polynomial $Q(x,y)$. Assuming

$$l_x = (k-1)(L+1) \quad (16)$$

and $|Q(x,y)| \ge n$ such that the value of L is determined by

$$L = \left\lfloor (-3 + \sqrt{1 + \frac{8n}{k-1}})/2 \right\rfloor \quad (17)$$

Proof: The bivariate polynomial is defined as

$$Q(x,y) = \sum_{j=0}^{L} Q^{(j)}(x) y^j = \sum_{j=0}^{L} \sum_{i=0}^{l_x - j(k-1)} b_{i,j} x^i y^j \quad (18)$$

The value of $|Q(x,y)|$ is given by

$$|Q(x,y)| = \sum_{j=0}^{L} \sum_{i=0}^{l_x - j(k-1)} 1 = (L+1)(l_x - L(k-1)/2)$$

Assuming $l_x = (k-1)(L+1)$ and $|Q(x,y)| \ge n$, the above equation becomes

$$\frac{L^2+3L+2}{2} \geq \frac{n}{k-1}.$$

And then

$$L \geq \left\lfloor (-3+\sqrt{1+\frac{8n}{k-1}})/2 \right\rfloor.$$

#

Theorem 2: A modified EKE algorithm is derived in the following

$$\sum_{t=1}^{L}\sum_{j=0}^{l_x-t(k-1)} Q_j^{(t)} S_{i+j}^{(t)} = 0, \quad 0 \leq i < n-l_x-1, \tag{19}$$

which can be denoted as

$$\begin{pmatrix} S_0^{(1)} & \cdots & S_{l_x-k+1}^{(1)} & S_0^{(2)} & \cdots & S_{l_x-L(k-1)}^{(L)} \\ S_1^{(1)} & \cdots & S_{l_x-k+2}^{(1)} & S_1^{(2)} & \cdots & S_{l_x-L(k-1)+1}^{(L)} \\ & & & \vdots & & \\ S_{n-l_x-2}^{(1)} & \cdots & S_{n-k-1}^{(1)} & S_{n-l_x-2}^{(2)} & \cdots & S_{n-L(k-1)-2}^{(L)} \end{pmatrix} \bullet \begin{pmatrix} Q_0^{(1)} \\ \vdots \\ Q_{l_x-(k-1)}^{(1)} \\ Q_0^{(2)} \\ \vdots \\ Q_{l_x-L(k-1)}^{(L)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{20}$$

Proof: The proof is similar to this in [13]. The variable *y* is replaced by a received polynomial $V(x) = v_0 + v_1 x + \cdots v_{n-1}x^{n-1}$ and $V(\alpha_i) = v_i$ and $i \in [n-1]$ in (18), which yields the following univariate polynomial:

$$\sum_{t=0}^{L} Q^{(t)}(x)(V(x))^t = B(x)\prod_{j=1}^{n}(x-\alpha_j), \tag{21}$$

where

$$Q^{(t)}(x) = \sum_{j=0}^{l_x} Q_j^{(t)}(x)$$

and $\deg B(x) = L(n-1)+k-1-n$. If the order of coefficients in both sides of (21) is reversed, then the coefficients of $1, x, \cdots, x^{L(n-k)-1}$ in both sides of the result should be identical. Namely,

$$x^{k-1+(n-1)L}\sum_{t=1}^{L} Q^{(t)}(x^{-1})(V(x^{-1}))^t \equiv x^{k-1+(n-1)L-n} B(x^{-1})\prod_{j=1}^{n}(1-\alpha_j x) \bmod x^{L(n-k)}$$

Further more, it becomes

$$\sum_{t=1}^{L}\Lambda^{(t)}(x)x^{(L-t)(n-k)}(\overline{V}(x))^t \equiv \overline{B}(x)G(x) \bmod x^{L(n-k)} \tag{22}$$

where

$$\Lambda^{(t)}(x) = \sum_{j=0}^{l_x-t(k-1)}\Lambda_j^{(t)}x^j = x^{l_x-t(k-1)}Q^{(t)}(x^{-1})$$
$$\overline{V}(x) = x^{n-1}V(x^{-1})$$
$$\overline{B}(x) = x^{L(n-1)+(k-1)-n}B(x^{-1}),$$
$$G(x) = \prod_{j=1}^{n}(1-\alpha_j x) \ .$$

Let $\overline{V}(x)$ be expressed as following:

$$(\overline{V}(x))^t = (x^{(t-1)(n-1)-1}S^{(t)}(x) + U^{(t)}(x))G(x) \tag{23}$$

where $S^{(t)}(x)$ has been defined in (8) and $U^{(t)}(x) \in F_{(t-1)(n-1)-1}[x]$. We obtain the following equation by substituting (23) into (22),

$$\sum_{t=1}^{L}\Lambda^{(t)}(x)x^{(L-t)(n-k)+(t-1)(n-1)-1}S^{(t)}(x) \equiv \widetilde{V}(x) \bmod x^{L(n-k)} \tag{24}$$

where

$$\widetilde{V}(x) = \overline{B}(x) - \sum_{t=1}^{L}\Lambda^{(t)}(x)x^{(L-t)(n-k)}U^{(t)}(x)$$

Moreover, (26) is divided by $x^{(L-1)(n-k)}$, which yields the following

$$\sum_{t=1}^{L}\Lambda^{(t)}(x)x^{(t-1)(k-1)-1}S^{(t)}(x)\equiv\Omega(x)\bmod x^{n-k} \tag{25}$$

where

$$\Omega(x)=\widetilde{V}(x)/x^{(L-1)(n-k)}.$$

Since $\deg\Omega(x)=L(k-1)-1$, the coefficients of powers x^i, $L(k-1)\le i\le n-k-1$, in (25) are zeros. Therefore,

$$\sum_{t=1}^{L}\sum_{j=0}^{l_x-t(k-1)}\Lambda_j^{(t)}S_{i-j-(k-1)(t-1)+1}^{(t)}=0\ ,$$
$$L(k-1)\le i\le n-k-1$$

For more computations, we obtain the following

$$\sum_{t=1}^{L}\sum_{j=0}^{l_x-t(k-1)}Q_j^{(t)}S_{i+j}^{(t)}=0\ ,\ 0\le i<n-l_x-1$$

#

The list decoding algorithm [13] of an [*n*, *k*] GRS code is adjusted as

1. Compute syndrome elements, $\mathbf{S}^{(1)}=\mathbf{v}\cdot\mathbf{H}^T$ [18], for a received vector $\mathbf{v}$. If $\mathbf{S}^{(1)}$ is an all-zero vector, then output the corresponding message vector $\mathbf{m}$ determined as $\mathbf{m}=\mathbf{m}'\cdot\mathbf{M}$ [18], where a vector $\mathbf{m}'$ is the last *k*-tuple of $\mathbf{v}$ and $\mathbf{M}$ is a $k\times k$ matrix such that $\mathbf{G}'=\mathbf{M}\cdot\mathbf{G}$ is a systematical matrix. Go to Step 5.

2. Perform the modified EKE interpolation:

1) Compute the other syndrome polynomials in parallel: $S^{(t)}(x)=\sum_{i=0}^{n-l_x-2}S_i^{(t)}x^i$, and $S_i^{(t)}=\sum_{j=0}^{n-1}v_j^t\alpha^{(i+1)j}$ and $t\in\{2,\cdots,L\}$.

2) Find the polynomial $Q^{(t)}(x)=\sum_{j=0}^{l_x-t(k-1)}Q_j^{(t)}x^j$, $1\le t\le L$, by the GBM algorithm such that $\sum_{t=1}^{L}\sum_{j=0}^{l_x-t(k-1)}Q_j^{(t)}S_{i+j}^{(t)}=0$, $0\le i<n-l_x-1$.

3) Obtain the polynomial $Q^{(0)}(x)$ such that $Q^{(0)}(\alpha^j)=-\sum_{t=1}^{l_x}Q^{(t)}(\alpha^j)v_j^t$, $j\in[n-1]$, and then form the bivariate polynomial $Q(x,y)=\sum_{t=0}^{L}Q^{(t)}(x)y^t$.

3. Perform the factorization on the bivariate polynomial $Q(x,y)$ by employing the reconstruction algorithm [13] to find the *y*-root polynomials

$$\hat{f}(x)=\hat{m}_0+\hat{m}_1x+\cdots+\hat{m}_{k-1}x^{k-1}$$

4. Compute the corresponding codeword $\hat{\mathbf{c}}=(\hat{f}(1),\hat{f}(\alpha),\cdots,\hat{f}(\alpha^{n-1}))$ for each polynomial $\hat{f}(x)$ .Output the message vector $\mathbf{m}^*=(m_0^*,m_1^*,\cdots,m_{k-1}^*)$ of the codeword $\mathbf{c}^*$ with the shortest Hamming distance between values of $\mathbf{v}$.

5. Terminate decoding.

The following two examples illustrate the modified EKE decoding algorithm. Firstly, this proposed algorithm decodes GRS (7, 3) code, as error-free and error-existing conditions are encountered. The other case is to decode GRS (15, 3) code with errors more than $(n-k)/2$, which 7 errors are corrected in proposed decoding.

Example : For a (15, 3) GRS code over GF(2^4) is generated by the polynomial $p(X)=1+X+X^4$. The message is $\mathbf{m}=(\alpha^9,\alpha^{14},\alpha^8)$ and its codeword is

$$\mathbf{v} = (\alpha^5, \alpha^6, \alpha^{10}, \alpha^{10}, 0, 1, \alpha^9, \alpha^{13}, \alpha^7, 1, \alpha^{13}, \alpha^6, 0, \alpha^5, \alpha^7)$$

If the error vector $\mathbf{e} = (0, \alpha^{13}, 0, \alpha^2, \alpha^{14}, 0, 0, 0, \alpha^6, 0, \alpha^6, 0, 0, \alpha^2, \alpha)$ appears over the transmission channel, then the received vector at the output of demodulator is

$$\begin{aligned}\mathbf{r} &= \mathbf{v} + \mathbf{e} \\ &= (\alpha^5, 1, \alpha^{10}, \alpha^4, \alpha^{14}, 1, \alpha^9, \alpha^{13}, \alpha^{10}, 1, 1, \alpha^6, 0, \alpha, \alpha^{14})\end{aligned}$$

The syndrome vectors are computed in the following

$$\mathbf{S}^{(1)} = (\alpha^6, \alpha^{11}, \alpha^6, \alpha^{11}, \alpha^5, \alpha^{10}, \alpha^3, \alpha^{11}, \alpha, \alpha^2, \alpha^{13}, \alpha^4)$$

and

$$\mathbf{S}^{(2)} = (\alpha^7, \alpha^{12}, \alpha^2, \alpha^7, \alpha^4, \alpha^{12}, \alpha^{11}, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{13}, \alpha^5)$$

The values of the highest degrees of *y*-power and *x*-power in the bivariate polynomial $Q(x, y)$ computed in (18) and (19) are

$$L = \left\lfloor (-3 + \sqrt{1 + \frac{8 \times 15}{3 - 1}}) / 2 \right\rfloor = 2$$

and $l_x = (3-1)(2+1) = 6$.

The bivariate polynomial $Q(x, y)$ determined at Step 2 is shown in the following:

$$\begin{aligned}Q(x, y) = {} & \alpha^{10} + \alpha^{12} x + \alpha^4 x^2 + \alpha^5 x^3 + \alpha^8 x^5 + \\ & \alpha^{10} x^6 + y(\alpha^{10} + \alpha^{12} x^2 + x^4) + \\ & y^2(\alpha^{14} + \alpha^5 x + x^2)\end{aligned}$$

The *y*-root polynomial of $Q(x, y)$ factorized by the reconstruction algorithm is

$$\hat{f}(x) = \alpha^9 + \alpha^{14} x + \alpha^8 x^2$$

whose message is $\mathbf{m}^* = (\alpha^9, \alpha^{14}, \alpha^8)$.

Ⅳ. COMPLEXITY ANALYSIS

For obtaining of the bivariate polynomial $Q(x, y)$, the EKE algorithm in [13], which is similar to the GBM algorithms[5][14], is realized to determine the shortest length $Q^{(t)}(x) = \sum_{j=0}^{N_t - 1} Q_j^{(t)} x^j$, $t \in \{1, 2, \cdots l\}$, such that (11) holds. Since the GBM algorithm is run column by column in a matrix: in a column of the matrix, while a nonzero discrepancy on a row occurs, the corresponding shortest polynomial $Q^{(t)}(x)$ is updated at this point. As a result, the size of the matrix dominates the decoding complexity. Obviously, reducing the size of the matrix lowers the complexity of locating the smallest length of linear dependent coefficients. The dimensions of the S-matrix in (11) are τ by $l(n - \tau - (k-1)(l+1)/2)$. The EKE algorithm requires the solving of τ homogeneous linear equations in (11) to obtain $Q^{(t)}(x)$, $t \in \{1, 2, \cdots l\}$, and then finding the corresponding coefficients of $Q^{(0)}(x)$ in (14). The time complexity of the EKE algorithm is $O(l \cdot (n-k)^2)$ (equally, $O(l \cdot n^2 (1-R)^2)$), which improves on the time complexity of $O(n^3)$ of Sudan's algorithm. In the modified EKE algorithm, the dimensions of the S-matrix in (22) are $n - l_x - 1$ by $L(1 + l_x / 2)$. According the complexity analysis on [13, p253], the time complexity of the proposed algorithm is

$O(n \cdot (n - \sqrt{nk}))$ (equally, $O(n^2 (1 - \sqrt{R}))$).

A design parameter *l* in [13] is an upper bound on the size of the list of decoded codewords. As

code rate $R \le 1/3$, the value of l is determined by the following range,

$$\frac{2n-2\tau-k+1-\sqrt{(2n-2\tau-k+1)^2-8\tau(k-1)}}{2(k-1)} < l < \frac{2n-2\tau-k+1+\sqrt{(2n-2\tau-k+1)^2-8\tau(k-1)}}{2(k-1)}$$

The value of l determines the size of S-matrix in (11). The higher value of l results in more decoding complexity in (11). However, the lower value of l could decrease the error correction capability of the list decoding algorithm [13]. While the size of an S-matrix is large, the complexity of locating the corresponding shortest linear dependent coefficients in $Q^{(t)}(x)$ becomes huge accordingly. In order to simply compare the complexity of these two ELE algorithms, the dimensions and sizes of S-matrices of (11) and (20) are listed in Table 1, in which GRS codes of length 255 and code rate $R \le 1/3$ are employed and the smallest value of l, which is $l = 2$, is selected. For cases of $R \le 0.1$ (equally, message length $k \le 25$), the sizes of S-matrices in (11) are less than those in (20). However, for the other cases, the complexity of (11) might be larger than that in (20), based on that the sizes of S-matrices in (11) are larger than those in (20).

V. CONCLUSIONS

This work presents a modified EKE algorithm, incorporating syndrome computations and matrix operations, which obtain the transmission information from the received codeword. The computations of syndrome elements do not increase the complexity of the original EKE algorithm, because it is an item in such a proposed decoding. The proposed EKE algorithm is beneficial when the signal-to-noise ratio is high. Additionally, this work provides a way to generalize the EKE algorithm, which decodes a GRS code with any code rate. The proposed EKE algorithm requires less complexity than the original EKE algorithm does while GRS codes of code length 255 and code rate $0.1 \le R \le 0.3$ are used.

ACKNOWLEDGEMENTS

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under contract no. NSC 97-2221-E-212-005.

REFERENCES

[1] Agashe, P., Rezaiifar, R., and Bender P., “Cdma2000 high rate broadcast packet data air interface design,” IEEE Commun. Mag., Vol. 42, No. 2, pp. 83–89, Feb. 2004.

[2] Lin, Shu, and Costello, Daniel Jr., Error Control Coding, Prentice hall, 2nd edition, 2004.

[3] Wicker, Stephen B., Error Control Systems for Digital Communication and Storage, Prentice hall, 1995.

[4] Peterson, W. W. and Weldon, E. J., Error-Control Codes, MIT press, Cambridge, 2nd edition, 1972.

[5] Massy, J. L, “Step-by-Step Decoding of Bose-Chauhuri-Hocquenghem codes,” IEEE Trans. Inform. Theory, IT-11, No. 4, pp.580-585, Nov., 1965.

[6] Wei, S.-W. and Wei, C-H. “High-Speed Decoder of Reed-Solomon Codes, “ IEEE Trans. Commun., Vol. 41, No.11, Nov. 1993.

[7] Chen, T.-C., Wei, C.-H. and Wei, S.-W. “Step-by-step decoding algorithm for Reed-Solomon codes,” IEE Proc.-Commun., Vol. 147, No.1, Feb. 2000.

[8] Welch, L., and Berlekamp, E.R., "Error Correction for Algebraic Block Codes", U.S. Patent 4 633 470, September 1983.
[9] Fedorenko, S. V., "A simple algorithm for decoding Reed-Solomon codes and its relation to the Welch-Berlekamp algorithm," IEEE Trans. Inform. Theory, Vol. 51, No.3, pp. 1196-1198, March. 2005.
[10] Sudan, M., "Decoding of Reed-Solomon codes beyond the error-correction bound," J. Compl., Vol. 13, pp. 180-193, 1997.
[11] Guruswami, V. and Sudan, M., "Improved decoding of Rees-Solomon and algebraic-geometric codes," IEEE Trans. Inform. Theory, Vol. 45, pp.1757-1767, 1999.
[12] Koetter, R., and Vardy, A., "Algebraic soft-decision decoding of Reed–Solomon Codes," IEEE Trans. Inform. Theory, Vol. 49, No.11, pp.2809-2825, Nov. 2003.
[13] Roth, Ron M. and Ruckenstein, Gitit, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," Trans. Inform. Theory, Vol. 46, No.1, pp.246-257, Jan. 2000.
[14] Feng, Gui-Liang and Tzeng, Kenneth K., "A generalization of Berlekamp-Massy algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," Trans. Inform. Theory, Vol. 37, No.5, pp.1274-1287, Sept. 1991.
[15] MacWilliams, F. J. and Sloane, N. J., The Theory of Error Correcting Codes. Amsterdam, The Netherlands: North Holland, 1977.
[16] McEliece, R. J., The Theory of Information and Coding, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[17] Pellikaan, R. and Wu, X.-W., "List Decoding of q-ary Reed-Muller Codes," Trans. Inform. Theory, Vol. 50, No.4, pp.679-682, April. 2004.

Table 1: The dimensions and sizes of S-matrices in two EKE algorithms

	EKE algorithm			modified EKE algorithm		
k	τ	$l(n-\tau-(k-1)(l+1)/2)$ $l=2$	Size of S-matrix in (11)	$n-l_x-1$	$L(1+l_x/2)$	Size of S-matrix in (22)
3	167	170	28390	224	224	50176
5	165	168	27720	214	189	40446
7	163	166	27058	206	175	36050
9	161	164	26404	198	174	34452
11	159	162	25758	194	155	30070
13	157	160	25120	182	185	33670
15	155	158	24490	184	144	26496
17	153	156	23868	174	164	28536
19	151	154	23254	182	111	20202
21	149	152	22648	174	123	21402
23	147	150	22050	166	135	22410
25	145	148	21460	158	147	23226
27	143	146	20878	176	80	14080
29	141	144	20304	170	86	14620
31	139	142	19738	164	92	15088
33	137	140	19180	158	98	15484
35	135	138	18630	152	104	15808
37	133	136	18088	146	110	16060
39	131	134	17554	140	116	16240
41	129	132	17028	134	122	16348
43	127	130	16510	128	128	16384
45	125	128	16000	166	45	7470
47	123	126	15498	162	47	7614
49	121	124	15004	158	49	7742
51	119	122	14518	154	51	7854
53	117	120	14040	150	53	7950
55	115	118	13570	146	55	8030
57	113	116	13108	142	57	8094
59	111	114	12654	138	59	8142
61	109	112	12208	134	61	8174
63	107	110	11770	130	63	8190
65	105	108	11340	126	65	8190
67	103	106	10918	122	67	8174
69	101	104	10504	118	69	8142
71	99	102	10098	114	71	8094
73	97	100	9700	110	73	8030
75	95	98	9310	106	75	7950