Remainder Decoding of Binary Quadratic Residue Codes Using the Gao's Algorithm

Pei-Yu Shih[1], Trieu-Kien Truong[2], and Yaotsu Chang[3]

[1] *Department of Information Eng., I-Shou University, Kaohsiung County, Taiwan, 84001, ROC*

[2] *College of Electrical and Information Eng., I-Shou University, Kaohsiung County, Taiwan, 84001, ROC*

[3] *Department of Applied Mathematics, I-Shou University, Kaohsiung County, Taiwan, 84001, ROC*

d9203005@stmail.isu.edu.tw, truong@isu.edu.tw, ytchang@isu.edu.tw

ABSTRACT

The quadratic residue (QR) codes whose code rates are greater than or equal 1/2 and generally have high error-correcting capacity are widely used in communication for channel coding. In this paper, a new decoding method is proposed for the binary QR codes. The key ideal behind the proposed method is to apply the properties of remainder decoding and the Gao's algorithm. In the remainder decoding, the main feature of efficient compute syndromes is contained in our decoding method. And the modified Gao's algorithm is also used in our decoding algorithm. The new algorithm has been verified by a software simulation using C^{++} language running through possible error patterns. An example of (17, 9, 5) QR code using this decoding algorithm is given.

Keywords: Quadratic residue codes, remainder decoding, the Gao's algorithm, error-correcting code.

1: INTRODUCTIONS

The QR codes which were introduced by Prange [1] are cyclic error-correcting codes. (7, 4, 3) and (23, 12, 7) QR codes are the well-known Hamming codes [2] and Golay code [3-5], respectively. The QR codes with length less than or equal to 113 have been decoding via varieties of decoding methods expect for the case of length 89. Those methods used most often to decoding include Sylvester resultant [6-7], GrÖbner bases [8-9] or the Berlekamp-Massey (BM) algorithm [10-11]. The first two methods can be used to solve the Newton identities that are non-linear multivariate equations of higher degrees. As the code length increasing, the first two methods become difficult. Furthermore, different QR codes use different sets of conditions to determine the error-locations. Consequently, it is hard to hardware implement. In the past, the BM algorithm was widely applied in decoding Reed-Solomon RS codes, Bose-Chaudhuri- Hocquenghem (BCH) codes, and many other codes. The evaluations of syndromes play an important role in decoding procedure. The remainder technique proposed by [12] could be applied to calculus the values of syndromes. To use BM algorithm in decoding QR codes, the enough consecutive syndromes is a necessary condition. In 2001, a new technique to express the unknown syndromes as functions of known syndromes was developed by He *et al* [13]. Recently, Gao [14] proposed an efficient scheme to decode RS codes which is called the Gao's algorithm by Fedorenko [15].

The proposed decoding scheme replaces the remainder technique with the directly computing the values of known syndromes from the received vector. This technique reduces the complexity of syndromes calculus. Also, the method developed by He *et al* [13] is used to determine the unknown syndromes of QR codes. Finally, the determined syndrome polynomial is applied in the key equation of the Gao's algorithm given in [15]. The Gao's algorithm proposes an efficient condition that is suitable for QR codes. Then we solve the key equation using the extended Euclidean algorithm (EEA) to obtain the error-locator polynomial. After Chain search, the error locations are found. In order to explain the proposed decoding scheme, an example of (17, 9, 5) QR code up to two errors is given.

2: PRELIMINARY

Let n be a prime congruent to +1 or -1 (mod 8), and let Q_n denote the set of nonzero quadratic residues (mod n). Let β be a primitive nth root of unity in an extension field of $GF(2)$, and let the polynomial $g(x)$ be defined by $g(x) = \prod_{i \in Q_n} (x - \beta^i)$. Then $g(x)$ is a polynomial with coefficient in $GF(2)$. The binary cyclic code of length n with generator polynomial $g(x)$ is called the (n, k, d) QR code.

The received vector $R(x)$ is represented as a polynomial is $R(x) = C(x) + E(x) = \sum_{i=0}^{n-1} c_i x^i + \sum_{i=0}^{n-1} e_i x^i$ where the codeword $C(x)$ equals to the product of the message polynomial $m(x)$ and the generator polynomial $g(x)$, $E(x)$ is the error polynomial and c_i, e_i belong to $GF(2)$.

The error locator polynomial is defined by

$$W(x)=\prod_{i=1}^{t}(1-X_i x), \quad (1)$$

where t is the correcting-error capacity and X_i is the errors location.

The remainder polynomial $r(x)$ with degree less than the degree of $g(x)$ is of the form

$$R(x)\equiv r(x) \bmod g(x). \quad (2)$$

The syndrome is defined as $s_i = E(\beta^i)$ where $0\le i\le n-1$. There is an relation among syndromes, namely, $S_{2i}=S_i^2$, with subindex modulo n, if necessary. If i belong to Q_n, the syndromes are called the known syndromes and have the property

$$S_i = R(\beta^i) = r(\beta^i) \text{ for } 0\le i\le n-1. \quad (3)$$

Otherwise, the syndromes are called the unknown syndromes and are not obtained directly from the remainder polynomial $r(x)$.

In order to get the correctly values of unknown syndromes, a method developed by He *et al* [13] is summarized in the following.

Assume that v errors occur in the received vector. Consider the matrix $S(I, J)$ of size $(v+1)\times(v+1)$ as follows:

$$S(I,J)=\begin{bmatrix} S_{i_1+j_1} & S_{i_1+j_2} & \cdots & S_{i_1+j_{v+1}} \\ S_{i_2+j_1} & S_{i_2+j_2} & \cdots & S_{i_2+j_{v+1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i_{v+1}+j_1} & S_{i_{v+1}+j_1} & \cdots & S_{i_{v+1}+j_{v+1}} \end{bmatrix}, \quad (4)$$

where the summation of the subindices of the S_i's is modulo n, and $\det(S(I, J))=0$. If there is only one unknown syndrome among the entries of $S(I, J)$, then it can be expressed as a function in terms of some known syndromes. Hence, during the decoding process, one can evaluate the value of the unknown syndrome with the information about those known syndromes. The relation among syndromes is used to determinate all values of syndromes.

After getting all syndromes, the syndrome polynomial is defined to have the following forms:

$$S(x)=S_1x+S_2x+\cdots+S_{n-1}x^{n-1}. \quad (5)$$

The key equation of the Gao's algorithm [15] is as follows:

$$\begin{cases} W(x)T(x)\equiv P(x) \bmod x^n-1 \\ \deg P(x)<\dfrac{n+k}{2} \\ \max imize \deg P(x) \end{cases}, \quad (6)$$

where the interpolation polynomial $T(x)$ is connection with the remainder polynomial $r(x)$ and all the roots of x^n-1, i. e., $\{\beta^0,\beta^1,\cdots,\beta^{n-1}\}$. Applying the EEA to x^n-1 and $T(x)$, we obtain unique pair of polynomials $W(x)$ and $P(x)$. In the fact, the polynomial $W(x)$ is the error locator polynomial.

3:THE SYNDROME POLYNOMIAL AND THE MODIFIED KEY EQUATION OF THE GAO'S ALGORITHM

This section contains the main theorem about syndrome polynomial and the modified key equation of the Gao's algorithm which are the foundation of our algorithm.

Theorem 1: Consider a binary (n, k, d) QR code. Let $S(x)=S_1x+S_2x^2+\cdots+S_{n-1}x^{n-1}$ be the syndrome polynomial for the received vector $R(x)$. If the weight v of a correctable error pattern $E(x)$ is odd (resp., even), then $S(x)$ (resp., $1+S(x)$) has v distinct linear factors, $(1-\beta^i x)$ where β is a primitive root of x^n-1.

Proof: Assume that the number of errors is odd, i.e., $v=2u+1$. By the definition of syndromes mentioned in Section (3), we have $S_i=(\beta^{l_1})^i+(\beta^{l_2})^i+\cdots+(\beta^{l_{2u+1}})^i$, where $0\le i\le n-1$. Then the evaluation of S(x) at β^{-i} yields the following:

$$\begin{aligned} S(\beta^{-i}) &= (\beta^{l_1}+\beta^{l_2}+\cdots+\beta^{l_{2u+1}})(\beta^{-i}) \\ &\quad +(\beta^{2l_1}+\beta^{2l_2}+\cdots+\beta^{2l_{2u+1}})(\beta^{-2i}) \\ &\quad +\cdots+(\beta^{(n-1)l_1}+\beta^{(n-1)l_2}+\cdots \\ &\quad +\beta^{(n-1)l_{2u+1}})(\beta^{-(n-1)i}) \\ &= (\beta^{l_1-i}+\beta^{2(l_1-i)}+\cdots+\beta^{(n-1)(l_1-i)}) \\ &\quad +(\beta^{l_2-i}+\beta^{2(l_2-i)}+\cdots+\beta^{(n-1)(l_2-i)})+\cdots \\ &\quad +(\beta^{l_{2u+1}-i}+\beta^{2(l_{2u+1}-i)}+\cdots+\beta^{(n-1)(l_{2u+1}-i)}). \end{aligned} \quad (7)$$

Since each β^{-i} is a root of the AOP, $(\beta^{-i})^{n-1}+(\beta^{-i})^{n-2}+\cdots+1=0$, i.e., $(\beta^{-i})^{n-1}+(\beta^{-i})^{n-2}+\cdots+(\beta^{-i})=1$. In (7), if $i=l_j$, $1\le j\le 2u+1$, then the jth summand equals $\beta^{l_j-l_j}+\beta^{2(l_j-l_j)}+\cdots+\beta^{(n-1)(l_j-l_j)}=1+1+\cdots+1=n-1\equiv 0 \pmod 2$. All other summands have the same value $\beta^{l_j-l_w}+\beta^{2(l_j-l_w)}+\cdots+\beta^{(n-1)(l_j-l_w)}=\beta^{l_j-l_w}+(\beta^{l_j-l_w})^2+\cdots+(\beta^{l_j-l_w})^{n-1}=1$, for $w\ne j$, because $\beta^{(l_j-l_w)}$ is a root of AOP. Therefore, (7) becomes $S(\beta^{-i})=0+1+\cdots+1=2u\equiv 0 \pmod 2$ if $i=l_j$, $1\le j\le 2u+1$. On the other hand, if $i\ne l_j$, then (7) becomes $S(\beta^{-i})=1+1+\cdots+1=2u+1\equiv 1 \pmod 2$ $\ne 0$. That is, for the case of odd errors, i.e., v is odd, S(x) has exactly v roots in B, i.e., $\prod_{j=1}^{v}(1-\beta^{l_j}x)\mid S(x)$, where $\prod_{j=1}^{v}(1-\beta^{l_j}x)$ is the error-locator polynomial $W(x)$. By a similar argument, when the number of

errors is even, i.e., v=2u, $1+S(\beta^{-i})=1+(2u-1)\equiv 0 \pmod 2$ if $i=l_j,\ 1\le j\le 2u$. If $i\neq l_j$, then $1+S(\beta^{-i}) = 1+2u\equiv 1 \pmod 2 \neq 0$. In other words, for the case of even errors, there are precisely v roots in $\{\beta^0,\beta^1,\cdots,\beta^{n-1}\}$ such that 1+S(x)=0, i.e., $W(x)=\prod_{j=1}^{v}(1-\beta^{l_j}x) \mid (1+S(x))$. This completes the proof of Theorem 1.

Additionally, the fact $\prod_{i=0}^{n-1}(1-\beta^i x)=x^n-1$ implies that the greatest common division (g.c.d) of $S(x)$ (resp., $1+S(x)$) and x^n-1 is the error locator polynomial $W(x)$ for v odd (resp., v even).

The EEA is applied to find the g.c.d of two nonzero polynomials a, and b over GF(q). Given the initial conditions $r_{-1}=a, r_0=b, u_{-1}=1, u_0=0, v_{-1}=1, v_0=0$, it proceeds according to the following recursion relation:

$$r_i=r_{i-2}-q_i r_{i-1},\ u_i=u_{i-2}-q_i u_{i-1},\ v_i=v_{i-2}-q_i v_{i-1}, \quad (8)$$

where $\deg r_{i-1} < \deg r_i$. For all i, we have the relation $u_i a+v_i b=r_i$. The key equation in Equation (6) can be rewritten as follows.

$$W(x)T(x)+\theta(x)(x^n-1)=P(x). \quad (9)$$

Let polynomial $S(x)$ (resp., $1+S(x)$) replace to $T(x)$ in Equation (6), when the weight v of $E(x)$ is odd (resp., even). If we use the EEA to determine the g.c.d of $T(x)$ and x^n-1, we generate sets of solutions $(W_l(x),P_l(x),\theta_l(x))$. $W_l(x)$ and $P_l(x)$ are useful for our decoding method. The particular solution $W_l(x)$ is the error locator polynomial when $P_l(x)$ is degree less than (n+k)/2.

4:THE NEW DECODING ALGORITHM OF QR CODES

The new decoding algorithm for the QR codes and an example of (17, 9, 5) QR code are given in the section.

If the syndromes are all zero calculated by Equation (3), there is no error in the received word. When the errors occur in received word, the decoding algorithm is summarized below by nine steps.

Step1: Calculate the remainder polynomial $r(x)$ by Equation (2).

Step2: Evaluate the known syndromes by using Equation (3)

Step3: Initialize by letting v=1.

Step4: Compute the unknown syndromes by applying the technique in [13].

Step5: Solve congruence in equation (6) by applying the EEA to x^n-1 and $T(x)$, and the unique pair of polynomials $P_l(x)$ and $W_l(x)$ are determined.

Step6: Applying Chien search to find the roots of $W_l(x)$.

Step7: If there are exists v errors, go to Step9. Otherwise, set v=v+1.

Step8: If v>t, stop. If not, go to Step4.

Step9: The error polynomial is determined and then the received word can be corrected.

An example of decoding (17, 9, 5) QR code is shown as follows to explain our proposed decoding algorithm in detail.

Example:

Let α be a root of the primitive polynomial $x^8+x^4+x^3+x^2+1$ and let $\beta=\alpha^{(2^8-1)/17}=\alpha^{15}$ be a primitive 17st root of unity in $GF(2^8)$. The set of quadratic residue modulo 17 is $Q_{17}=\{1, 2, 4, 8, 9, 13, 15, 16\}$. The generator polynomial of binary (17, 9, 5) QR code can be written as

$$g(x)=\prod_{i\in Q_{17}}(x-\beta^i)=1+x+x^2+x^4+x^6+x^7+x^8.$$

If the information polynomial $m(x)$ is

$$m(x)=1+x^2+x^4+x^6+x^8,$$

then the code polynomial $C(x)$ is

$$C(x)=1+x+x^2+x^5+x^6+x^7+x^8+x^{10}+x^{12}+x^{14}+x^{16},$$

which is a multiple of g(x). We assume that the error polynomial $E(x)$ is

$$E(x)=x^1+x^{14}.$$

Then the received polynomial is the sum of the code polynomial $C(x)$ and the error polynomial $E(x)$, i.e.

$$R(x)=C(x)+E(x)\equiv r(x) \bmod g(x) = 1+x^2+x^3+x^6.$$

The decoding process developed in this paper is described as follows. First of all, the known syndrome S_k for each k in Q_{17} can be calculated from the remainder polynomial $r(x)$. That is,

$$S_k=\sum_{i=0}^{16} r_i(\beta^k)^i,\ k\in Q_{17}.$$

For the binary (17, 9, 5) QR code, every known syndromes (resp., unknown syndromes) can be expressed as some power of primary syndrome S_1 (resp., S_3). The relations among syndromes for (17, 9, 5) QR code is given in following:

$$S_2=S_1^2, S_4=S_1^4, S_8=S_1^8, S_{16}=S_1^{16},$$
$$S_{15}=S_1^{32}, S_{13}=S_1^{64}, S_9=S_1^{128},$$
$$S_6=S_3^2, S_{12}=S_3^4, S_7=S_3^8, S_{14}=S_3^{16},$$
$$S_{11}=S_3^{32}, S_5=S_3^{64}, S_{10}=S_3^{128}.$$

By evaluating $r(x)$ at the roots of $g(x)$ mentioned above, the primary known syndrome is $S_1 = \alpha^{87} \neq 0$, which means that there are errors occurred in the received polynomial $r(x)$.

If the number of errors is one, i.e., v=1, the primary unknown syndrome is $S_3 = S_1^3 = \alpha^6$. After the determination of the primary syndromes S_1 and S_3, all syndromes can be also determined. Therefore, we further obtain the syndrome polynomial

$$S(x) = \alpha^{87}x + \alpha^{174}x^2 + \alpha^6 x^3 + \alpha^{93}x^4 + \alpha^{129}x^5 + \alpha^{12}x^6 + \alpha^{48}x^7 + \alpha^{186}x^8 + \alpha^{171}x^9 + \alpha^3 x^{10} + \alpha^{192}x^{11} + \alpha^{24}x^{12} + \alpha^{213}x^{13} + \alpha^{96}x^{14} + \alpha^{234}x^{15} + \alpha^{117}x^{16}.$$

The EEA is applied to polynomial $T(x)$=$S(x)$+1 and $x^n - 1$ in Equation (9). This is accomplish by the recursive formulas Equation (8) illustrated in Table 1, where initially $P_{-1}(x) = x^n - 1$ and $P_0(x) = T(x)$. From Table 1, one observes that $\deg P(x) = \deg P_4(x) = 12 < (17+9)/2 = 13$. Thus, the computation terminates at this point for i=4, and

$$W_4(x) = 1 + \alpha^{227}x^3 + \alpha^{110}x^4.$$

Using Chien search to find the root of the $W_4(x)$, there is no root $\{\beta^i \mid 0 \le i \le 16\}$ in $W_4(x)$, and thus the assumption is not valid.

If the number of the errors is two, the primary unknown syndrome S_3 can be determined by the technique developed in [13]. A computer search is used to find the following matrix of size 3×3

$$\begin{bmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_{15} & S_{16} & S_0 \end{bmatrix}.$$

There is only one unknown syndrome S_3 among the entries of this matrix. By [13], the determinant of the above matrix is zero. The unknown syndrome S_3 for the two-error case is thus

$$S_3 = \frac{S_1 S_2 S_{16} + S_2^2 S_{15}}{S_1 S_{15}} = \alpha^{244},$$

where $S_0 = 0$ and $S_1 = \alpha^{87}$. Since v=2 is even, the polynomial $T(x)$=1+$S(x)$ is used in the EEA. Similarly, the processing of the EEA is illustrated in Table 2. The computation terminates at this point for i=4, and

$$W_4(x) = 1 + \alpha^{87}x + \alpha^{225}x^2.$$

There exists exactly two roots β^{-1}, β^{-14}, in $W_4(x)$ via Chien search. In other words, the error polynomial $e(x) = x^1 + x^{14}$ is determined.

5: CONCLUSION

In this paper, a new decoding algorithm of the QR codes is proposed. We apply the remainder technique and the key equation of the Gao's algorithm in our decoding method. The remainder technique is used in calculating known syndromes effectively and the key equation of the Gao's algorithm supplies a successful condition to determine the error locator polynomial. It would be interesting to see if there exists a generalized condition to determine the number of occurred errors.

REFERENCES

[1] E. Prange, "Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms," *Air Force Cambridge Research Center*-TN-58-156, Cambridge, MA: 1958.

[2] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147-160, 1950.

[3] M. J. E. Golay," Notes on digital coding," *Proc. IRE*, vol. 37, p.67, 1949.

[4] S. Reed, X. Yin, T. K. Truong, and J. K. Holmes, "Decoding the (24, 12, 8) Golay code," *Proc. IEE*, vol. 137, pp. 202-206, May 1990.

[5] M. Elia, "Algebraic decoding of the (23, 12, 7) Golay code," *IEEE Trans. Inf. Theory*, vol. 33, pp. 150-151, Jan. 1987.

[6] S. Reed, X. Yin, and T. K. Truong, "Algebraic decoding of the (32, 16, 8) quadratic residue code," *IEEE Trans. Inf. Theory*, vol. 36, pp. 876-880, July. 1990.

[7] S. Reed, T. K. Truong, X. Chen, and X. Yin, "The algebraic decoding of the (41, 21, 9) quadratic residue code," *IEEE Trans. Inf. Theory*, vol. 38, pp. 974-985, May 1992.

[8] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, "Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1654-1661, Sep. 1994.

[9] P. Loustaunau and E. V. York, "On the decoding of cyclic codes using Gröbner bases," *Appl. Alg. Eng. Commun. Comput.*, vol. 8, pp. 469-483, 1997.

[10] Y. Chang, T. K. Truong, I. S. Reed, H. Y. Cheng, and C. D. Lee, "Algebraic decoding of (71, 36, 11), (79, 40, 15), and (97, 49, 15) quadratic residue codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 1463-1473, Sep. 2003.

[11] T. K. Truong, Y. Chang, Y. H. Chen, and C. D. Lee, "Algebraic decoding of (103, 52, 19), and (113, 57, 15) quadratic residue codes," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 749-754, May 2005.

[12] L. Welch and E. R. Berlekamp, "Error Correction for Algebraic Block Codes," US Patent 4 633 470, 1983.

[13] R. He, I. S. Reed, T. K. Truong, and X. Chen, "Decoding the (47, 24, 11) quadratic residue code," *IEEE Trans. Inf. Theory*, vol. 47, no. 3 pp. 1181-1186, Mar. 2001.

[14] S. Gao, "A new algorithm for decoding Reed-Solomon codes," in Communications, Information and Network Security, V. Bhargava,

H. V. Poor, V. Tarokh, and S. Yoon, Eds. Norwell, MA: Kluwer, 2003, vol. 712, pp. 55-68.

[15] Sergei V. Fedorenko, " A Simple Algorithm for Decoding Reed-Solomon Codes and its Relation to the Welch-Berlekamp Algorithm," *IEEE Trans. Inf. Theory*, vol. 51, pp. 1196-1198, no. 3, Mar. 2005.

Table 1. Applying the EEA to find W(x) as v=1

i	$q_i(x)$	$P_i(x)$	$W_i(x)$
-1		$x^{17}-1$	0
0	$\alpha^{138}x$	$\alpha^{87}x+\alpha^{174}x^2+\alpha^{6}x^3+\alpha^{93}x^4+\alpha^{129}x^5+\alpha^{12}x^6+\alpha^{48}x^7+\alpha^{186}x^8+\alpha^{171}x^9+\alpha^{3}x^{10}+\alpha^{192}x^{11}+\alpha^{24}x^{12}+\alpha^{213}x^{13}+\alpha^{96}x^{14}+\alpha^{234}x^{15}+\alpha^{117}x^{16}$	1
1	1	$1+\alpha^{28}x^2+\alpha^{57}x^3+\alpha^{144}x^4+\alpha^{231}x^5+\alpha^{12}x^6+\alpha^{150}x^7+\alpha^{168}x^8+\alpha^{69}x^9+\alpha^{54}x^{10}+\alpha^{141}x^{11}+\alpha^{75}x^{12}+\alpha^{162}x^{13}+\alpha^{96}x^{14}+\alpha^{234}x^{15}+\alpha^{117}x^{16}$	$\alpha^{138}x$
2	$\alpha^{227}x^3$	$1+\alpha^{87}x+\alpha^{157}x^2+\alpha^{244}x^3+\alpha^{76}x^4+\alpha^{95}x^5+\alpha^{14}x^7+\alpha^{35}x^9+\alpha^{241}x^{10}+\alpha^{124}x^{11}+\alpha^{7}x^{12}+\alpha^{145}x^{13}$	$1+\alpha^{138}x$
3	1	$1+\alpha^{28}x^2+\alpha^{142}x^3+\alpha^{229}x^4+\alpha^{95}x^5+\alpha^{199}x^6+\alpha^{14}x^7+\alpha^{220}x^8+\alpha^{69}x^9+\alpha^{3}x^{10}+\alpha^{141}x^{11}+\alpha^{24}x^{12}+\alpha^{145}x^{13}$	$\alpha^{138}x+\alpha^{227}x^3+\alpha^{110}x^4$
4		$\alpha^{87}x+\alpha^{174}x^2+\alpha^{108}x^3+\alpha^{195}x^4+\alpha^{199}x^6+\alpha^{220}x^8+\alpha^{171}x^9+\alpha^{54}x^{10}+\alpha^{192}x^{11}+\alpha^{75}x^{12}$	$1+\alpha^{227}x^3+\alpha^{110}x^4$

Table 2. Applying the EEA to find W(x) as v=2

i	$q_i(x)$	$P_i(x)$	$W_i(x)$
-1		$x^{17}-1$	0
0	$\alpha^{138}x$	$\alpha^{87}x+\alpha^{174}x^2+\alpha^{244}x^3+\alpha^{93}x^4+\alpha^{61}x^5+\alpha^{233}x^6+\alpha^{167}x^7+\alpha^{186}x^8+\alpha^{171}x^9+\alpha^{122}x^{10}+\alpha^{158}x^{11}+\alpha^{211}x^{12}+\alpha^{213}x^{13}+\alpha^{79}x^{14}+\alpha^{234}x^{15}+\alpha^{117}x^{16}$	1
1	1	$1+\alpha^{138}x+\alpha^{28}x^2+\alpha^{57}x^3+\alpha^{127}x^4+\alpha^{231}x^5+\alpha^{199}x^6+\alpha^{116}x^7+\alpha^{50}x^8+\alpha^{69}x^9+\alpha^{54}x^{10}+\alpha^{5}x^{11}+\alpha^{41}x^{12}+\alpha^{94}x^{13}+\alpha^{96}x^{14}+\alpha^{217}x^{15}+\alpha^{117}x^{16}$	$\alpha^{138}x$
2	$\alpha^{87}x$	$\alpha^{70}x+\alpha^{157}x^2+\alpha^{6}x^3+\alpha^{229}x^4+\alpha^{146}x^5+\alpha^{80}x^6+\alpha^{99}x^7+\alpha^{84}x^8+\alpha^{35}x^9+\alpha^{71}x^{10}+\alpha^{124}x^{11}+\alpha^{126}x^{12}+\alpha^{247}x^{13}+\alpha^{147}x^{14}+\alpha^{30}x^{15}$	$1+\alpha^{138}x$
3	1	$1+\alpha^{138}x+\alpha^{174}x^2+\alpha^{6}x^3+\alpha^{229}x^4+\alpha^{146}x^5+\alpha^{80}x^6+\alpha^{99}x^7+\alpha^{84}x^8+\alpha^{35}x^9+\alpha^{71}x^{10}+\alpha^{124}x^{11}+\alpha^{126}x^{12}+\alpha^{247}x^{13}+\alpha^{147}x^{14}+\alpha^{30}x^{15}$	$\alpha^{70}x+\alpha^{24}x^2$
4		$1+\alpha^{87}x+\alpha^{225}x^2$	$1+\alpha^{87}x+\alpha^{225}x^2$

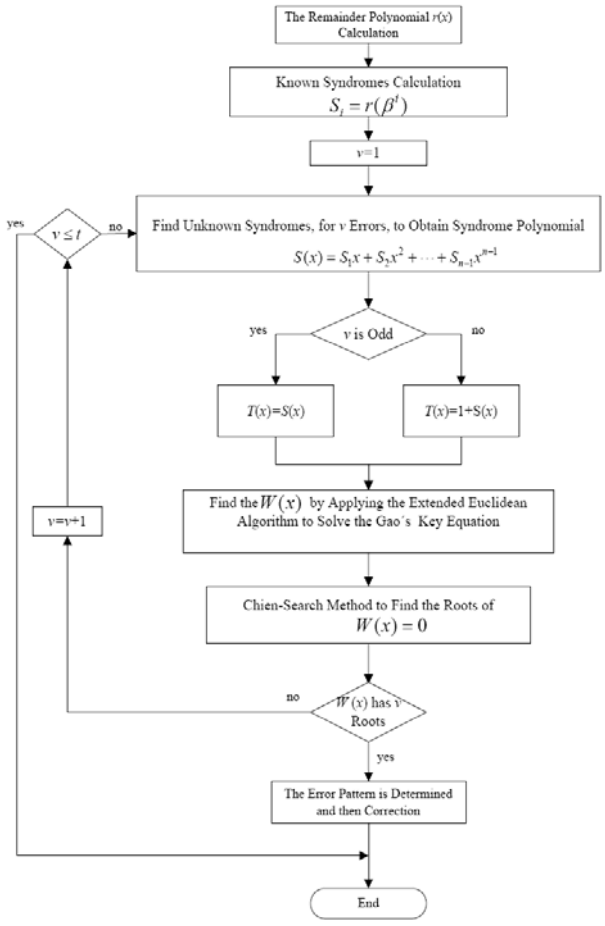

Fig. 1 Flowchart of new QR decoder