Decoding the (89, 45, 17) Quadratic Residue Code

Pei-Yu Shih, Wen-Ku Su, Tsung-Ching Lin, and Trieu-Kien Truong

Department of Information Engineering

I-Shou University

Kaohsiung County 84001, Taiwan, R.O.C.

E-mail: d9203005@stmail.isu.edu.tw; d9403003@stmail.isu.edu.tw; joe@isu.edu.tw; truong@isu.edu.tw

Abstract—In this paper, a modification decoding algorithm for the (89, 45, 17) QR code is proposed to decode all error patterns with weight less than six and high percentage weight-6, weight-7, and weight-8 error patterns.

I. INTRODUCTION

A series of different algebraic decoding algorithm for the quadratic residue (QR) codes which was introduced by Prange [1] in 1958, have been proposed [2], [3]-[9] but the algebraic decoding schemes for the (89, 45, 17) QR code is not available in the literature. The (89, 45, 17) QR code, which has the capability to correct up to eight errors, can be constructed in a quite small field $GF(2^{18})$. Due to this small size of the finite field, the computational complexity can be dramatically reduced. Additionally, its generator polynomial is a product of four irreducible polynomials $g_1(x)$, $g_2(x)$, $g_3(x)$, and $g_{11}(x)$ over $GF(2)$. Therefore it is one of the best in the family of the binary quadratic residue codes and is difficult to decode all error patterns within error correcting capacity eight.

In the past decades, most of decoding methods for the QR codes are used to solve the Newton identities which are nonlinear, multivariate equations of quite high degree. It becomes very difficult when the weight of the occurred error becomes large. Moreover, different QR codes use different sets of conditions to determine the error locations. As a result, it is not practical for software implementation.

Recently, an algebraic decoding algorithm for QR codes was developed by Truong et al. [7]. The advantage of that decoding scheme is that it uses the inverse-free Berlekamp-Massey (BM) algorithm to determine the error-locator polynomial. It can also be used to develop the decoders for many other binary QR codes of lengths up to 113, see [8] and [9], except for the QR codes of lengths 31, 73, and 89.

II. BACKGROUND OF THE (89, 45, 17) QR CODE

A binary QR code of length $n$ is an $(n, (n+1)/2)$ cyclic code with the generator polynomial of the form:

$$g(x) = \prod_{i \in Q_n} (x - \beta^i)$$

where $Q_n$ is the collection of all nonzero quadratic residues modulo $n$ given by

$$Q_n = \{ i | i \equiv j^2 \mod n \text{ for } 1 \leq j \leq n-1 \}$$

and $\beta$ is a primitive $n$th root of unity in the finite field $GF(2^m)$ with $m$ being the smallest positive integer such that $n \mid 2^m - 1$.

For the (89, 45, 17) QR code, the generator polynomial $g(x)$ is defined as follows:

$$g(x) = \prod_{i \in Q_{89}} (x - \beta^i) = x^{44} + x^{42} + x^{41} + x^{39} + x^{37} + x^{34} + x^{33} + x^{31} + x^{30} + x^{29} + x^{28} + x^{26} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{16} + x^{15} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^5 + x^3 + x^2 + 1.$$  (3)

Let the code polynomial $c(x) = c_0 + c_1x + \cdots + c_{88}x^{88}$, which is a multiple of $g(x)$, be transmitted through a noisy channel. Also, $e(x) = e_0 + e_1x + \cdots + e_{88}x^{88}$ be the error polynomial. Then the received polynomial

$$r(x) = r_0 + r_1x + \cdots + r_{88}x^{88}$$

can be expressed as a sum of the code polynomial and the error polynomial; that is,

$$r(x) = c(x) + e(x).$$

The set of known syndromes is obtained by evaluating $r(x)$ at the roots of $g(x)$, namely,

$$S_i = r(\beta^i) = e(\beta^i), i \in Q_n.$$  (4)

Suppose that there are $v$ errors occurred in the received polynomial $r(x)$. Then the error polynomial has $v$ nonzero terms, namely, $e(x) = x^i + x^{i+1} + \cdots + x^j$, where $0 \leq i < l_1 < \cdots < l_j \leq n-1$. The syndrome $S_i$ can be written as $S_i = Z_{l_1} + Z_{l_2} + \cdots + Z_{l_j}$, where $Z_l = \beta^l$ for $1 \leq j \leq v$ are called the error locators. If $i$ is not in the set $Q_{89}$, the syndrome $S_i$ is what is called the unknown syndrome. Assume that $v$ errors are occurred in the received word. The
error-locator polynomial $\sigma(x)$ is defined to be a polynomial of degree $v$

$$\sigma(x) = \prod_{j=1}^{v} (1 + Z_j x) = 1 + \sum_{j=1}^{v} \sigma_j x^j$$ (5)

where $\sigma_j = x_1 + \cdots + x_j$, $\sigma_1 = x_1 x_2 + \cdots + x_{v-1} x_v$, and $\sigma_v = x_1 \cdots x_v$. One way to decode the QR code is to determine the error-locator polynomial $\sigma(x)$ and then the Chien search is applied to find the roots of $\sigma(x)$.

III. DECODING METHOD OF THE (89, 45, 17) QR CODE

For the (89, 45, 17) QR code, the generator polynomial is reducible, then all the known syndromes and unknown syndromes can be expressed as some powers of $S_1$, $S_2$, $S_3$, $S_4$, $S_5$, and $S_6$, $S_7$, $S_8$, $S_9$, respectively. One way to decode the QR code is to determine the error-locator polynomial $\sigma(x)$ and then the Chien search is applied to find the roots of $\sigma(x)$.

Even though the strategy used in this paper is, the same as the technique used in [8], to determine the values of the primary unknown syndromes, it is harder to find the matrices $S(I, J)$ consisting of only one primary unknown syndrome when the number of error occurred is larger.

After determining the primary unknown syndromes $S_3$ and $S_{13}$, one now has enough syndromes to apply the modified inverse-free BM algorithm proposed in this paper.

IV. DETERMINATIONS OF THE PRIMARY UNKNOWN SYNDROMES $S_3$ AND $S_{13}$

In the following, the primary unknown syndromes $S_3$ and $S_{13}$ needed for the inverse-free BM algorithm are determined. An index with parentheses is attached to the unknown syndrome “$S_3$” to obtain the notation “$S_3^{(v)}$”, indicating that the formulae obtained are valid for the $v$-error case only. Also, the digit $v$ after the word “Case” indicates that the number of errors occurred is $v$. Furthermore, “Case va)” and “Case vb)” denote two sub-cases for determining the primary unknown syndromes $S_3^{(v)}$ and $S_{13}^{(v)}$, respectively. In each case, one lists explicitly the two subsets $I$ and $J$ needed in [8] to determine $S_3^{(v)}$ and $S_{13}^{(v)}$, separately. Moreover, in each odd-error case, the syndrome $S_0$ is always equal to one, but zero in every even-error case.

Case 0: The unknown syndromes $S_3^{(0)}$ and $S_{13}^{(0)}$ for the zero-error case are obviously $S_3^{(0)} = 0$ and $S_{13}^{(0)} = 0$.

Case 1: For the one-error case, the unknown syndromes $S_3^{(1)} = S_3$ and $S_{13}^{(1)} = S_{13}$.

Case 2: Assume the number of errors is two and $S_0 = 0$. There are two sub-cases to be considered as follows:

a) Let $I = \{0, 1, 2 \}$ and $J = \{0, 1, 87\}$. By $\det(S(I, J)) = 0$, the unknown syndrome $S_3^{(2)}$ is given by $S_3^{(2)} = \frac{(S_3 S_2 S_{88} + S_2 S_5)}{(S_3 + S_{88})}$.

b) Let $I = \{0, 11, 88\}$ and $J = \{0, 2, 5\}$. It follows from $\det(S(I, J)) = 0$ that the unknown syndrome $S_{13}^{(2)}$ given by $S_{13}^{(2)} = \frac{(S_3 S_8 S_{11} + S_8 S_{11})}{(S_3 + S_{88})}$.

Case 3: Assume $v = 3$ and $S_0 = 1$.

a) Let $I = \{0, 1, 87, 88\}$ and $J = \{0, 1, 2, 10\}$. Then $S_{3}^{(3)}$ can be determined by solving $\det(S(I, J)) = 0$.

b) Let $I = \{8, 9, 10, 13\}$ and $J = \{0, 8, 59, 71\}$. By solving $\det(S(I, J)) = 0$, one yields the formula of $S_{13}^{(3)}$.

In the following cases, the value of $S_3^{(v)}$ is first obtained in “Case va)” and is then viewed as a known syndrome for the determination of $S_{13}^{(v)}$ in “Case vb)”.

Case 4: For the four-error case, one considers the following two sub-cases.

a) Let $I = \{1, 2, 3, 10, 32\}$ and $J = \{0, 7, 8, 77, 78\}$. Eq. $\det(S(I, J)) = 0$ is then used to determine $S_{3}^{(4)}$.

b) Let $I = \{0, 1, 2, 87, 88\}$ and $J = \{0, 1, 2, 3, 11\}$. The unknown syndrome $S_{3}^{(4)}$ is determined from the use of $\det(S(I, J)) = 0$.

Case 5: For $v = 5$, two sub-cases need to be considered.

a) Let $I = \{1, 6, 17, 20, 39, 57\}$ and $J = \{0, 33, 52, 68, 71, 72\}$ and $J_2 = \{0, 1, 8, 39, 49, 72\}$, $J_3 = \{0, 1, 8, 39, 49, 84\}$. The determinants of the matrices $S(I_3, J_3)$ and $S(I_3, J_2)$ are polynomials of degree one in $S_3^{(5)}$, respectively. Also, the determinants of the two matrices yield the following linear equations in $S_3^{(5)}$.

$$m_1 S_3^{(5)} + z_1 = 0, \quad m_2 S_3^{(5)} + z_2 = 0.$$ (6)

For all five-error patterns, the unknown syndrome $S_3^{(5)}$ can be determined uniquely by solving one of the equations in (6).

b) Let $I = \{0, 1, 2, 3, 4, 5\}$ and $J = \{0, 1, 2, 8, 87, 88\}$. The unknown syndrome $S_{13}^{(5)}$ is determined by solving $\det(S(I, J)) = 0$.

Case 6: For the six-error case, the following two sub-cases are considered.

a) Let $I = \{1, 2, 3, 4, 9, 71, 87\}$, $J_3 = \{0, 1, 2, 7, 8,$
Case 7: The seven-error case is similar to the six-error case. According to the technique in [8], one obtains the matrices \(S(I^1_v, J^1_v)\), and \(S(I^2_v, J^2_v)\) whose determinants are the two polynomials \(f_1(S^{(6)}_3)\), and \(g_1(S^{(6)}_3)\) in \(S^{(6)}_3\), respectively. Obviously, \(S^{(6)}_3\) must be the root of \(F_1(S^{(6)}_3)\), where \(F_1(S^{(6)}_3)\) is the greatest common divisor of nonzero polynomials among \(f_1(S^{(6)}_3)\), and \(g_1(S^{(6)}_3)\). A software simulation is run by a computer program in order to check all six-error patterns. In 95% such patterns, the degree of \(F_1(S^{(6)}_3)\) is equal to one. Thus, the unknown syndrome \(S^{(6)}_3\) can be determined uniquely for more than 95% six-errors.

b) As usual, let \(I^1_v = \{0, 1, 2, 3, 4, 5, 6\}\) and \(J^1_v = \{0, 1, 2, 3, 7, 87, 88\}\). Then, with the given \(S^{(6)}_3\) obtained above, \(S^{(6)}_3\) can be determined by solving \(\det(I, J) = 0\).

Case 7: The seven-error case is similar to the six-error case.

a) Let \(I^1_v = \{2, 3, 4, 6, 7, 8, 14, 20\}\), \(J^1_v = \{0, 2, 4, 14, 65, 85, 86, 87\}\), and \(I^2_v = \{1, 2, 3, 4, 6, 8, 14, 87\}\), \(J^2_v = \{0, 2, 3, 4, 6, 8, 86, 87\}\). By (2), the matrices \(S(I^1_v, J^1_v)\), and \(S(I^2_v, J^2_v)\) are obtained. Therefore, the determinants of these matrices yield two polynomials in \(S^{(7)}_3\), say, \(f_3(S^{(7)}_3)\), and \(g_3(S^{(7)}_3)\), respectively. Obviously, \(S^{(7)}_3\) must be the root of \(F_3(S^{(7)}_3)\), where \(F_3(S^{(7)}_3)\) is the greatest common divisor of nonzero polynomials among \(f_3(S^{(7)}_3)\), and \(g_3(S^{(7)}_3)\). One finds that \(F_3(S^{(7)}_3)\) is a polynomial of degree one by checking up to 95% seven-error patterns. The single \(S^{(7)}_3\) is thus determined.

b) In this sub-case, one chooses \(I^1_v = \{0, 1, 3, 6, 8, 17, 40, 84\}\), \(J^1_v = \{0, 4, 5, 6, 8, 17, 39, 47\}\), and \(I^2_v = \{0, 1, 3, 6, 8, 17, 40, 88\}\), \(J^2_v = \{0, 4, 5, 6, 8, 17, 47, 88\}\). The matrices \(S(I^1_v, J^1_v)\), \(S(I^2_v, J^2_v)\), obtained by the use of the technique in [8] are used to yield the following linear equations in \(S^{(7)}_3\):

\[w_1S^{(7)}_3+y_1=0, \quad w_2S^{(7)}_3+y_2=0. \tag{7}\]

In (7), the coefficients in these two equations are too long to be listed here and one uses \(w_1\) and \(w_2\) as well as \(y_1\) and \(y_2\) instead. For more than 95% seven-error patterns, the unknown syndrome \(S^{(7)}_3\) can be determined uniquely by solving one of the linear equations given in (7).

Case 8: For \(v=8\), two sub-cases need to be considered.

a) Let \(I^1_v = \{1, 2, 3, 6, 17, 20, 45, 84, 87\}\), \(J^1_v = \{0, 3, 4, 5, 8, 19, 22, 47, 86\}\), \(I^2_v = \{1, 3, 5, 6, 7, 11, 17, 39, 84\}\), \(J^2_v = \{0, 1, 3, 5, 11, 17, 39, 84, 86\}\).

One obtains the matrices \(S(I^1_v, J^1_v)\), and \(S(I^2_v, J^2_v)\). The determinants of these matrices give four polynomials in \(S^{(8)}_3\), say, \(f_3(S^{(8)}_3)\), and \(g_3(S^{(8)}_3)\). Let \(F_3(S^{(8)}_3)\) be the greatest common divisor of \(f_3(S^{(8)}_3)\), and \(g_3(S^{(8)}_3)\). One finds that \(F_3(S^{(8)}_3)\) is a polynomial of degree one by checking up to 95% eight-error patterns. The single \(S^{(8)}_3\) is thus determined.

b) Let \(I^1_v = \{2, 6, 7, 8, 9, 18, 53, 85, 87\}\), \(J^1_v = \{0, 2, 3, 4, 14, 16, 38, 82, 83\}\), and \(I^2_v = \{0, 1, 2, 3, 4, 5, 6, 20, 87\}\), \(J^2_v = \{0, 1, 2, 3, 4, 5, 8, 44, 87\}\).

The determinants of the matrices \(S(I^1_v, J^1_v)\), and \(S(I^2_v, J^2_v)\) yield, respectively, the following linear equations in \(S^{(8)}_3\):

\[t_1S^{(8)}_3+k_1=0, \quad t_2S^{(8)}_3+k_2=0. \tag{8}\]

Again, in (8), since the coefficients in these equations are too long to be listed here, one uses \(t_1\) and \(t_2\) as well as \(k_1\) and \(k_2\) instead. For more than 95% eight-error patterns, the unknown syndrome \(S^{(8)}_3\) can be determined uniquely by solving one of the linear equations given in (8).

After determining the primary unknown syndrome, one now has enough syndromes to apply the inverse-free BM algorithm.

V. THE NEW DECODING ALGORITHM

The flowchart of the proposed (89, 45, 17) QR decoder is shown in Fig. 1. Let \(S^{(v)}_3\) denote the unknown syndromes which are computed under the assumption that \(v\) errors occur in the received word, where \(k \in \{1, 2, \ldots, 16\}\) and \(v \in \{1, 2, \ldots, 8\}\). Then compute the consecutive syndromes \(S_1, S_2, S^{(v)}_3, S_4, S_5, S_6, S^{(v)}_8, S_8, S_9, S_{10}, S_{11}, S_{12}, S^{(v)}_{13}, S_{14}, S^{(v)}_{15}, S_{16}\) that are needed to apply the inverse-free BM algorithm.

The algebraic decoding algorithm, shown in Fig 1, can be summarized by the following seven steps.

1) Initialize by letting \(v = 1\).

2) Compute the known syndromes \(S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_9, S_{10}, S_{11}, S_{12}\) by (4).

3) Compute the unknown syndromes \(S^{(v)}_3, S^{(v)}_6, S^{(v)}_7, S^{(v)}_{12}, S^{(v)}_{13}, S^{(v)}_{14}, S^{(v)}_{15}, S_{16}\) for \(1 \leq v \leq 8\) from the algorithm developed in Section IV.

4) Apply the inverse-free BM algorithm to determine the error-locator polynomial \(\sigma(x)\) of the QR code using the syndromes for \(v\) errors, obtained in steps 2) and 3).

5) Compute the roots of \(\sigma(x)\) by a use of Chien search. If the number of roots of \(\sigma(x)\) is equal to \(v\), the corrected codeword is obtained by subtracting the error pattern from the received word. Otherwise, set \(v = v + 1\).

6) If \(v > 8\), stop. Otherwise, return to step 3).

VI. SIMULATION RESULTS OF THE NEW DECODING ALGORITHM
A software simulation shows that among all the seven-error patterns, some of them can not be decoded properly by using the algebraic decoding algorithm developed by the authors [8]. In order to apply the algorithm [7] to decode the (89, 45, 17) QR code, a modification of this algorithm should be made. Therefore, one more condition is needed to the previous algorithm [7]. In other words, the additional condition is used.

The modified algebraic decoding algorithm has been verified by a computer program in C++ language running through all error patterns up to five errors and possible weight-6, weight-7, and weight-8 error patterns randomly. Moreover, the simulation results of the modified decoding algorithm show that all the error patterns with weight less than six and at least 95% for each of weight-6, weight-7, and weight-8 error patterns are decoded successfully.

REFERENCES