DOI: 10.13718/j.cnki.xdzk.2017.01.016

(41, 21, 9)二次剩余码的快速译码①

武登杰[1]，　包小敏[1]，　瞿云云[2]，　袁治华[1]

1. 西南大学 数学与统计学院，重庆 400715；2. 贵州师范大学 数学科学学院，贵阳 550001

摘要：针对(41, 21, 9)二次剩余码，提出了一种快速的基于校验子重量的译码算法(FSWDA). 这种译码算法结合了循环码的性质，又巧用了校验子的汉明重量，其最大优点是不需要存储校验子和相应的错误模式构成的表，却能达到查表译码的效果. 另外，本算法还可以用于其它纠错能力为 4 的二次剩余码.

关　键　词：二次剩余码；查表译码；错误模式；伴随式；汉明重量

中图分类号：O211.4　　**文献标志码**：A　　**文章编号**：1673-9868(2017)01-0103-06

二次剩余码(简称 QR 码) 是文献[1]提出的一类循环 BCH 码. 由于这类码的码率都大于或等于 1/2，所以大部分 QR 码都是比较好的纠错码. 例如(24, 12, 8)QR 码已被用于航海通信[2].

在过去的几十年里，关于QR码的译码方法已经提出很多种[3-9]，但这些译码方法很多都采用了一些代数工具，需要比较复杂的有限域中的计算，从而延长了译码时间. 为减少译码复杂性，查表译码就成为一个比较好的选择. 查表译码需要预先存储由校验子和相应的错误模式构成的表. 对于纠错能力为 4 的(41, 21, 9)QR 码，它需要存储 $\sum_{i=1}^{4}\binom{41}{i}=112\ 791$ 个校验子和相应的错误模式，存储大小为 $112\ 791\times(6\text{bytes}+3\text{bytes})\approx 991.3$ Kbytes. 文献[4]经过改善，提出的算法需要存储 101.26 Kbytes. 事实上，考虑循环码的特性，只需存储 $\sum_{i=1}^{4}\binom{41}{i}/41=2751$ 个校验子和相应的错误模式即可，此时存储大小仅为 24.18 Kbytes[2]. 而文献[5]和文献[6]又进一步优化，分别将存储大小减少到2.47 Kbytes和2.03 Kbytes，大大节省了存储空间.

本文继续沿用查表译码的思想，同时又利用校验子与校验矩阵之间的关系，提出了一种基于校验子重量的快速译码算法(fast syndrome-weight decoding algorithm，FSWDA). 这个算法具有运算简单、存储量小的特点，而且可以用于其它纠错能力小于或等于 4 的 QR 码.

1　(41, 21, 9) 二次剩余码背景知识

二次剩余码是由域 GF(2) 上的生成多项式 $g(x)=g_0+g_1x+\cdots+g_{n-k}x^{n-k}$ 生成的一类线性循环码，用(n, k, d)来表示，其中 $n=8l\pm1$(l 为正整数) 为码长，k 为信息长度，d 为码的最小汉明距离. (41, 21, 9)QR 码就是其中一种. m 是能满足 n 整除 $2m-1$ 的最小正整数. 设 $Q_n=\{j\mid j\equiv x^2 \bmod n, 1\leqslant x\leqslant(n-1)/2\}$. 当 $n=41$ 时，$m=20$，且 $Q_{41}=\{1, 2, 4, 5, 8, 9, 10, 16, 18, 20, 21, 23, 25, 31, 32, 33, 36, 37,$

① 收稿日期：2015-09-28
基金项目：国家自然科学基金项目(61462016)；贵州省教育厅青年科技人才成长项目(黔教合 KY 字[2016]130).
作者简介：武登杰(1989-)，山西运城人，硕士研究生，主要从事编码理论、密码学的研究.
通信作者：包小敏，博士，教授.

39，40}. α 是本原多项式 $p(x)=x^{20}+x^3+1$ 的一个根. 因此，$\beta=\alpha^u$，$u=(2^m-1)/n=(2^{20}-1)/41=25\ 575$，是域 GF($2^{20}$) 上一个 41 阶本原根. 生成多项式 $g(x)$ 通过

$$g(x)=\prod_{i\in Q_{41}}(x-\beta^i)=x^{20}+x^{19}+x^{17}+x^{16}+x^{14}+x^{11}+x^{10}+x^9+x^6+x^4+x^3+x+1$$

可以得到.

由(41，21，9)QR 码的最小汉明距离 $d=9$，可知它的纠错能力 $t=\left\lfloor\dfrac{d-1}{2}\right\rfloor=4$. (41，21，9)QR 码的码字可用多项式 $\boldsymbol{c}(x)=c_0+c_1x+\cdots+c_{40}x^{40}$ 来表示，其中 $\boldsymbol{c}(x)$ 是 $g(x)$ 的倍式，即 $\boldsymbol{c}(x)=\boldsymbol{m}(x)g(x)$，这里 $\boldsymbol{m}(x)=m_0+m_1x+\cdots+m_{20}x^{20}$ 为码的信息多项式. 当码字通过一个有噪音的信道，接收到的多项式 $\boldsymbol{r}(x)=r_0+r_1x+\cdots+r_{40}x^{40}$ 是 $\boldsymbol{c}(x)$ 与错误多项式 $\boldsymbol{e}(x)=e_0+e_1x+\cdots+e_{40}x^{40}$ 的和. 为简单起见，信息、码字、错误模式、接收量和校验子分别用向量 $\boldsymbol{m}=[m_0,m_1,\cdots,m_{k-1}]$，$\boldsymbol{c}=[c_0,c_1,\cdots,c_{n-1}]$，$\boldsymbol{e}=[e_0,e_1,\cdots,e_{n-1}]$，$\boldsymbol{r}=[r_0,r_1,\cdots,r_{n-1}]$ 和 $\boldsymbol{s}=[s_0,s_1,\cdots,s_{n-k-1}]$ 来表示.

QR 码作为循环码，按下列方法可得它的一个生成多项式 $\boldsymbol{G}_1$[3]：

$$\boldsymbol{G}_1=\begin{bmatrix}g(x)\\xg(x)\\\vdots\\x^{k-1}g(x)\end{bmatrix}=\begin{bmatrix}g_0&\cdots&g_{n-k}&0&\cdots&0\\0&g_0&\cdots&g_{n-k}&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\ddots&\vdots\\0&0&0&g_0&\cdots&g_{n-k}\end{bmatrix}$$

对 $\boldsymbol{G}_1$ 进行行初等变换，可以得到形式为 $\boldsymbol{G}=[\boldsymbol{I}_k,\boldsymbol{A}]$ 的生成矩阵，其中 $\boldsymbol{I}_k$ 是一个 $k\times k$ 单位矩阵，$\boldsymbol{A}$ 是一个 $k\times(n-k)$ 矩阵. 由 $\boldsymbol{G}$ 可得形如 $\boldsymbol{H}=[-\boldsymbol{A}^{\mathrm{T}},\boldsymbol{I}_{n-k}]$ 的一个校验矩阵. 用 $\boldsymbol{H}_i$ 表示 $\boldsymbol{H}^{\mathrm{T}}$ 的第 i 行向量，即

$$\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}-\boldsymbol{A}\\\boldsymbol{I}_{n-k}\end{bmatrix}=[\boldsymbol{H}_1^{\mathrm{T}},\boldsymbol{H}_2^{\mathrm{T}},\cdots,\boldsymbol{H}_n^{\mathrm{T}}]^{\mathrm{T}}$$

对于(41，21，9)QR 码，矩阵 $\boldsymbol{A}$ 为

$$\begin{bmatrix}
1&1&0&1&1&0&1&0&0&1&1&1&0&0&1&0&1&1&0&1\\
1&0&1&1&0&1&1&1&0&1&0&0&1&0&1&1&1&0&1&1\\
1&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1&0&0&0&0\\
0&1&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1&0&0&0\\
0&0&1&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1&0\\
0&0&0&0&1&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1\\
1&1&0&1&1&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0\\
0&1&1&0&1&1&1&1&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&1&0&1&1&0&1&1&1&1&0&1&1&0&1&1&1&0&0\\
0&1&1&1&0&1&1&0&1&1&1&1&0&1&1&0&1&1&1&0\\
0&0&1&1&1&0&1&1&0&1&1&1&1&0&1&1&0&1&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&1&1&1&1&0&1&1&0\\
0&1&1&0&0&0&1&1&1&1&1&0&0&1&1&1&1&0&1&1\\
1&1&1&0&1&0&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\
0&1&1&1&0&1&0&1&1&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&1&1&0&1&0&1&1&1&0&0&0&0&0&0&1&0&0\\
0&0&0&1&1&1&0&1&0&1&1&1&0&0&0&0&0&0&1&0\\
0&0&0&0&1&1&1&0&1&0&1&1&1&0&0&0&0&0&0&1\\
1&1&0&1&1&1&0&1&0&0&1&0&1&1&1&0&1&1&0&1\\
1&0&1&1&0&1&0&0&1&1&1&0&0&1&0&1&1&0&1&1
\end{bmatrix}$$

当给定信息向量 $\boldsymbol{m}=[m_0,m_1,\cdots,m_{k-1}]$，码字 $\boldsymbol{c}$ 通过

$$\boldsymbol{c}=\boldsymbol{mG}=[m_0,\ m_1,\ \cdots,\ m_{k-1}][\boldsymbol{I}_k,\ \boldsymbol{A}_{k\times(n-k)}]=$$
$$[m_0,\ m_1,\ \cdots,\ m_{k-1},\ p_0,\ p_1,\ \cdots,\ p_{n-k-1}]$$

进行编码得到. 向量$[m_0,\ m_1,\ \cdots,\ m_{k-1}]$和$[p_0,\ p_1,\ \cdots,\ p_{n-k-1}]$分别为码字$\boldsymbol{c}$的信息部分和校验部分. 另外，对于接收向量$\boldsymbol{r}$，它的校验子$\boldsymbol{s}=\boldsymbol{rH}^{\mathrm{T}}$.

2　译码算法及相关定理

为了更好了解 FSWDA 算法，我们先给出相关的定义和结论. 需要注意的是这些结论都是针对$(n,\ k,\ d)$二进制循环码，所有计算都在域 GF(2) 上进行，而$\boldsymbol{H}=[\boldsymbol{A}^{\mathrm{T}},\ \boldsymbol{I}]$是它的一个校验矩阵.

定义 1　一个向量$\boldsymbol{z}$的非零分量的个数称为向量$\boldsymbol{z}$的汉明重量，记为$w(\boldsymbol{z})$. 向量$\boldsymbol{a}-\boldsymbol{b}$的汉明重量$w(\boldsymbol{a}-\boldsymbol{b})$表示向量$\boldsymbol{a}$和向量$\boldsymbol{b}$的不同分量的个数，又称为向量$\boldsymbol{a}$和向量$\boldsymbol{b}$的汉明距离.

设$\boldsymbol{e}$是错误向量$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{e}_P]=[e_0,\ e_1,\ \cdots,\ e_{n-1}]$，其中$\boldsymbol{e}_M=[e_0,\ e_1,\ \cdots,\ e_{k-1}]$是信息部分，$\boldsymbol{e}_P=[e_k,\ e_{k+1},\ \cdots,\ e_{n-1}]$是校验部分，$w(\boldsymbol{e})\leqslant t$，$\boldsymbol{s}=\boldsymbol{eH}^{T}$.

下面这个结论在后面的证明中多次用到：

定理 1　若$\boldsymbol{H}=[\boldsymbol{A}^{\mathrm{T}},\ \boldsymbol{I}]$，$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}]\neq 0_n$，则$w(\boldsymbol{eH}^{\mathrm{T}})\geqslant d-w(\boldsymbol{e}_M)$.

证　因为码字的重量至少是d，$\boldsymbol{e}_M[\boldsymbol{I},\ \boldsymbol{A}]=[\boldsymbol{e}_M,\ \boldsymbol{e}_M\boldsymbol{A}]$是码字，所以$w(\boldsymbol{e}_M)+w(\boldsymbol{e}_M\boldsymbol{A})\geqslant d$. 而

$$\boldsymbol{s}=\boldsymbol{eH}^{T}=[\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}][\boldsymbol{A}^{\mathrm{T}},\ \boldsymbol{I}]^{\mathrm{T}}=\boldsymbol{e}_M\boldsymbol{A}$$

故$w(\boldsymbol{eH}^{\mathrm{T}})=w(\boldsymbol{e}_M\boldsymbol{A})\geqslant d-w(\boldsymbol{e}_M)$.

在纠错范围内，定理 2 和定理 3 确定何时错误只出现在校验部分.

定理 2　若$\boldsymbol{e}=[\boldsymbol{0}_{1\times k},\ \boldsymbol{e}_P]$，其中$w(\boldsymbol{e}_P)\leqslant t$，则$w(\boldsymbol{s})=w(\boldsymbol{eH}^{T})\leqslant t$.

证　由$\boldsymbol{H}=[\boldsymbol{A}^{T},\ \boldsymbol{I}]$的结构可得

$$s=\boldsymbol{eH}^{T}=[\boldsymbol{0}_{1\times k},\ \boldsymbol{e}_P][\boldsymbol{A}^{T},\ \boldsymbol{I}]^{\mathrm{T}}=\boldsymbol{e}_P$$

故$w(\boldsymbol{s})=w(\boldsymbol{e}_P)\leqslant t$.

定理 3　若$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{e}_P]$，其中$w(\boldsymbol{e}_M)\geqslant 1$，$w(\boldsymbol{e}_M)+w(\boldsymbol{e}_P)\leqslant t$，则$w(\boldsymbol{s})=w(\boldsymbol{eH}^{T})\geqslant t+1$.

证　此时

$$\begin{aligned}w(\boldsymbol{eH}^{T})&=w([\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}]\boldsymbol{H}^{T}+[\boldsymbol{0}_{1\times k},\ \boldsymbol{e}_P]\boldsymbol{H}^{T})\geqslant\\&(d-w(\boldsymbol{e}_M))-w(\boldsymbol{e}_P)=\\&d-(w(\boldsymbol{e}_M)+w(\boldsymbol{e}_P))\geqslant\\&d-t\geqslant t+1\end{aligned}$$

故结论成立.

推论 1　若$w(\boldsymbol{e})\leqslant t$，则$w(\boldsymbol{eH}^{T})\leqslant t$当且仅当$\boldsymbol{e}=[\boldsymbol{0}_{1\times k},\ \boldsymbol{eH}^{T}]$.

当$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{e}_P]$时，下面的定理给出了$\boldsymbol{e}_P$与$\boldsymbol{e}$的伴随式和$\boldsymbol{e}_M$的伴随式之间的一个关系.

定理 4　设$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{e}_P]$. 若$[\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}]\boldsymbol{H}^{T}=\boldsymbol{s}_M$，则$\boldsymbol{e}_P=\boldsymbol{eH}^{T}-\boldsymbol{s}_M$.

证　因为$w(\boldsymbol{e}_P)\leqslant w(\boldsymbol{e})\leqslant t$，所以

$$\begin{aligned}[\boldsymbol{0}_{1\times k},\ \boldsymbol{e}_P]&=[\boldsymbol{0}_{1\times k},\ (\boldsymbol{e}-[\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}])\boldsymbol{H}^{T}]=\\&[\boldsymbol{0}_{1\times k},\ \boldsymbol{eH}^{T}-[\boldsymbol{e}_M,\ \boldsymbol{0}_{1\times(n-k)}]\boldsymbol{H}^{T}]=\\&[\boldsymbol{0}_{1\times k},\ \boldsymbol{eH}^{T}-\boldsymbol{s}_M]\end{aligned}$$

结论得证.

下面的定理是我们的查表译码算法正确性的理论基础.

定理 5　设$w(\boldsymbol{e})\leqslant t$，$\boldsymbol{e}=[\boldsymbol{e}_M,\ \boldsymbol{e}_P]$，$w(\boldsymbol{e}_M)\geqslant 1$. 若$\boldsymbol{eH}^{T}=\boldsymbol{s}$，则存在唯一一个$\hat{\boldsymbol{e}}=[\hat{\boldsymbol{e}}_M,\ \boldsymbol{0}]$及相应的校验子$\hat{\boldsymbol{s}}$，使得$w(\boldsymbol{s}-\hat{\boldsymbol{s}})+w(\hat{\boldsymbol{e}}_M)\leqslant t$. 此时一定有$\boldsymbol{e}=[\hat{\boldsymbol{e}},\ \boldsymbol{s}-\hat{\boldsymbol{s}}]$.

证　当$\hat{\boldsymbol{e}}_M=\boldsymbol{e}_M$时，由定理 4 知$\boldsymbol{e}_P=\boldsymbol{s}-\hat{\boldsymbol{s}}$，故$w(\boldsymbol{s}-\hat{\boldsymbol{s}})+w(\hat{\boldsymbol{e}}_M)=w(\boldsymbol{e}_p)+w(\boldsymbol{e}_M)\leqslant t$，说明存在

性成立. 下面只需证当 $\hat{\boldsymbol{e}}_M \neq \boldsymbol{e}_M$ 时必有 $w([\boldsymbol{s}-\boldsymbol{s}_i, \boldsymbol{e}_M^i]) \geqslant t+1$，可证明唯一性也成立. 实际上

$$\begin{aligned} w(\boldsymbol{s}-\hat{\boldsymbol{s}})+w(\hat{\boldsymbol{e}}_M) &= w([\boldsymbol{e}_M, \boldsymbol{e}_P]\boldsymbol{H}^{\mathrm{T}}-[\hat{\boldsymbol{e}}_M, \boldsymbol{0}]\boldsymbol{H}^{\mathrm{T}})+w(\hat{\boldsymbol{e}}_M)= \\ & w([\boldsymbol{e}_M-\hat{\boldsymbol{e}}_M, \boldsymbol{0}]\boldsymbol{H}^{\mathrm{T}}-[\boldsymbol{0}, \boldsymbol{e}_P]\boldsymbol{H}^{\mathrm{T}})+w(\hat{\boldsymbol{e}}_M)\geqslant \\ & w([\boldsymbol{e}_M-\hat{\boldsymbol{e}}_M, \boldsymbol{0}]\boldsymbol{H}^{\mathrm{T}})-w([\boldsymbol{0}, \boldsymbol{e}_P]\boldsymbol{H}^{\mathrm{T}})+w(\hat{\boldsymbol{e}}_M)\geqslant \\ & d-w(\boldsymbol{e}_M-\hat{\boldsymbol{e}}_M)-w(\boldsymbol{e}_P)+w(\hat{\boldsymbol{e}}_M)\geqslant \\ & d-(w(\boldsymbol{e}_M)+w(\hat{\boldsymbol{e}}_M))-w(\boldsymbol{e}_P)+w(\hat{\boldsymbol{e}}_M)= \\ & d-w(\boldsymbol{e})\geqslant d-t\geqslant t+1 \end{aligned}$$

当得到接收向量 $\boldsymbol{r}$ 时，向右循环 i 个比特可以得到 $\boldsymbol{r}^{(i)}=[r_{n-i}, \cdots, r_{n-1}, r_0, \cdots, r_{n-i-1}]$. 通过 $\boldsymbol{s}^{(i)}=\boldsymbol{r}^{(i)}\boldsymbol{H}$ 可以得到 $\boldsymbol{r}^{(i)}$ 的校验子. 由 $\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e}$ 易得 $\boldsymbol{r}^{(i)}=\boldsymbol{c}^{(i)}+\boldsymbol{e}^{(i)}$. 换句话说，如果 $\boldsymbol{r}$ 的错误模式是 $\boldsymbol{e}$，那么 $\boldsymbol{e}^{(i)}$ 是 $\boldsymbol{r}^{(i)}$ 相应的错误模式. 根据推论1，若 $1\leqslant w(\boldsymbol{s})\leqslant 4$，则错误仅出现在 $\boldsymbol{r}$ 的校验部分；若 $1\leqslant w(s^{(n-k)})\leqslant 4$，则错误仅出现在 $\boldsymbol{r}$ 的信息部分. 设 $\bar{\boldsymbol{e}}_0=[1, 0, \cdots, 0]$ 是 n 维向量，$\bar{\boldsymbol{e}}_j$ 为仅在下标为 j 的位置有错误的 n 维向量(除第 j 位为1外，其它位均为0)，相应的校验子 $\bar{\boldsymbol{s}}_j=\bar{\boldsymbol{e}}_j\boldsymbol{H}^T=\boldsymbol{H}_j$，$0\leqslant j\leqslant k-1$. 显然，在第 i 位和第 j 位比特发生错误可用 $\bar{\boldsymbol{e}}_i+\bar{\boldsymbol{e}}_j$ 来表示，$\boldsymbol{H}_i+\boldsymbol{H}_j$ 是 $\bar{\boldsymbol{e}}_i+\bar{\boldsymbol{e}}_j$ 的校验子，其中 $0\leqslant i\neq j\leqslant k-1$. 在这里可以看出，校验子、校验矩阵和错误模式的紧密联系，这就是不需要存储校验子和错误模式构成的表却仍可查表译码的原因. 设 $\boldsymbol{sd}_w$ 为第 w 步校子 $\boldsymbol{s}$ 与 $\boldsymbol{h}_i$ 的差值，即 $\boldsymbol{s}-\boldsymbol{h}_i$，FSWDA算法通过 $\boldsymbol{sd}_w$ 的汉明重量，能很快地确定错误模式.

用大写字母F,M和P分别表示 $\boldsymbol{r}$ 的第一个信息比特、其余信息比特和校验比特. 对于(41，21，9)QR码，共有24种错误情况(P，PP，PPP，PPPP，M，MM，MMM，MMMM，F，FP，MP，FPP，MPP，FPPP，MPPP，FM，FMM，MMP，FMMM，MMMP，FMP，FMPP，MMPP，FMMP)，覆盖了所有 $\sum_{i=1}^{4}\binom{41}{i}=112791$ 个错误模式. 如果 $w(\boldsymbol{s})=0$，那么接收向量 $\boldsymbol{r}$ 没有错误. 如果 $1\leqslant w(\boldsymbol{s})\leqslant 4$ 或者 $1\leqslant w(\boldsymbol{s}^{(20)})\leqslant 4$，那么可以解决8种错误情况(P，PP，PPP，PPPP，M，MM，MMM，MMMM). 计算 $\boldsymbol{sd}_3=(\boldsymbol{s}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 20$. 如果 $1\leqslant w(\boldsymbol{sd}_3)\leqslant 3$，那么又可解决7种错误情况(F，FP，MP，FPP，MPP，FPPP，MPPP). 若还没解决，则继续计算 $\boldsymbol{sd}_4=(\boldsymbol{s}^{(20)}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 20$. 如果 $1\leqslant w(\boldsymbol{sd}_4)\leqslant 3$，那么又有5种错误情况(FM，FMM，MMP，FMMM，MMMP)被检测解决. 若仍未解决，则再计算 $\boldsymbol{sd}_5=(\boldsymbol{s}-\boldsymbol{h}_i-\boldsymbol{h}_j)$，$0\leqslant i\leqslant 19$，$i+1\leqslant j\leqslant 20$. 如果 $1\leqslant w(\boldsymbol{sd}_5)\leqslant 2$，那么3种错误情况(FMP，FMPP，MMPP)能被检测解决. 若上述还不能找到错误，则需要计算 $\boldsymbol{sd}_6=(\boldsymbol{s}^{(20)}-\boldsymbol{h}_{20}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 19$. 如果 $w(\boldsymbol{sd}_6)=2$，那么一定属于(FMMP)这种错误情况. 表1列出了FSWDA算法每一步能解决的错误情况及相应的错误模式数量. FSWDA算法具体译码步骤如下：

1) 错误情况为(P，PP，PPP，PPPP). 根据推论1，计算 $\boldsymbol{s}$ 和 $w(\boldsymbol{s})$. 如果 $0\leqslant w(\boldsymbol{s})\leqslant 4$，那么错误模式 $\boldsymbol{e}=[\boldsymbol{0}_{1\times 21}, \boldsymbol{s}]$. 跳至7)；

2) 错误情况为(M，MM，MMM，MMMM). 根据推论1，计算 $w(\boldsymbol{s}^{(20)})$. 如果 $1\leqslant w(\boldsymbol{s}^{(20)})\leqslant 4$，那么错误模式 $\boldsymbol{e}^{(20)}=[\boldsymbol{0}_{1\times 21}, \boldsymbol{s}^{(20)}]$，也就是说，$\boldsymbol{e}=[\boldsymbol{0}_{1\times 1}, \boldsymbol{s}^{(20)}, \boldsymbol{0}_{1\times 20}]$，跳至7)；

3) 错误情况为(F，FP，MP，FPP，MPP，FPPP，MPPP). 计算 $\boldsymbol{sd}_3=(\boldsymbol{s}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 20$. 如果 $0\leqslant w(\boldsymbol{sd}_3)\leqslant 3$，那么错误向量 $\boldsymbol{e}=\bar{\boldsymbol{e}}_i+[\boldsymbol{0}_{1\times 21}, \boldsymbol{sd}_3]$. 跳至7)；

4) 错误情况为(FM，FMM，MMP，FMMM，MMMP). 计算 $\boldsymbol{sd}_4=(\boldsymbol{s}^{(20)}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 20$. 如果 $1\leqslant w(\boldsymbol{sd}_4)\leqslant 3$，那么错误向量 $\boldsymbol{e}^{(20)}=\bar{\boldsymbol{e}}_i+[\boldsymbol{0}_{1\times 21}, \boldsymbol{sd}_4]$，也就是说，$\boldsymbol{e}=\bar{\boldsymbol{e}}_{[i-20]}+[\boldsymbol{0}_{1\times 1}, \boldsymbol{sd}_4, \boldsymbol{0}_{1\times 20}]$，其中，$\bar{\boldsymbol{e}}$ 的下标$[x]$表示 x 模41. 跳至7)；

5) 错误情况为(FMP，FMPP，MMPP). 计算 $\boldsymbol{sd}_5=(\boldsymbol{s}-\boldsymbol{h}_i-\boldsymbol{h}_j)$，$0\leqslant i\leqslant 19$，$i+1\leqslant j\leqslant 20$. 如果 $1\leqslant w(\boldsymbol{sd}_5)\leqslant 2$，那么错误向量 $\boldsymbol{e}=\bar{\boldsymbol{e}}_i+\bar{\boldsymbol{e}}_j+[\boldsymbol{0}_{1\times 21}, \boldsymbol{sd}_5]$. 跳至7)；

6) 错误情况为(FMMP). 计算 $\boldsymbol{sd}_6=(\boldsymbol{s}^{(20)}-\boldsymbol{h}_{20}-\boldsymbol{h}_i)$，$0\leqslant i\leqslant 19$. 如果 $1\leqslant w(\boldsymbol{sd}_6)\leqslant 2$，那么错误模式 $\boldsymbol{e}^{(20)}=\bar{\boldsymbol{e}}_{20}+\bar{\boldsymbol{e}}_i+[\boldsymbol{0}_{1\times 21},\boldsymbol{sd}_6]$. 也就是说，$\boldsymbol{e}=\bar{\boldsymbol{e}}_0+\bar{\boldsymbol{e}}_{[i-20]}+[\boldsymbol{0}_{1\times 1},\boldsymbol{sd}_6,\boldsymbol{0}_{1\times 20}]$，跳至 7)；

7) 计算 $\boldsymbol{c}=\boldsymbol{r}+\boldsymbol{e}$. 跳至 8)；

8) 结束.

表 1　(41，21，9)QR 码的错误情况及相应的错误模式数量

步骤	错误情况	错误模式数量	错误情况	错误模式数量
1	P	$\binom{20}{1}=20$	PP	$\binom{20}{2}=190$
	PPP	$\binom{20}{3}=1140$	PPPP	$\binom{20}{4}=4845$
2	M	$\binom{20}{1}=20$	MM	$\binom{20}{2}=190$
	MMM	$\binom{20}{3}=1140$	MMMM	$\binom{20}{4}=4845$
3	F	1	FP	$\binom{20}{1}=20$
	MP	$\binom{20}{1}\binom{20}{1}=400$	FPP	$\binom{20}{2}=190$
	MPPP	$\binom{20}{1}\binom{20}{2}=3800$	FPPP	$\binom{20}{3}=1140$
	MPPP	$\binom{20}{1}\binom{20}{3}=22800$		
4	FM	$\binom{20}{1}=20$	FMM	$\binom{20}{2}=190$
	MMP	$\binom{20}{2}\binom{20}{1}=3800$	FMMM	$\binom{20}{3}=1140$
	MMMP	$\binom{20}{3}\binom{20}{1}=22800$		
5	FMP	$\binom{20}{1}\binom{20}{1}=400$	FMPP	$\binom{20}{1}\binom{20}{2}=3800$
	MMPP	$\binom{20}{2}\binom{20}{2}=36100$		
6	FMMP	$\binom{20}{1}\binom{20}{2}=3800$		

本文利用 Matlab 对(41，21，9)码纠错范围内的所有 112791 种错误进行模拟测试，所需平均译码时间大约为 $3.062\ 0\times 10^{-4}$ s，并与文献[5－6]中提出的算法进行对比，运算速度基本相当，但其突出特点是能够节省存储空间，对于存储能力有限的终端，特别是移动终端来说具有一定实际意义.

参考文献：

[1] PRANGE E. Some Cyclic Error－Correcting Codes with Simple Decoding Algorithms [R]. Cambridge：Air Force Cambridge Research Center，1958.

[2] WICKER S B. Error Control Systems for Digital Communication and Storage [M]. New Jersey：Prentice Hall，1995.

[3] MCELIECE R J. The Theory of Information and Coding [M]. 2nd ed. 北京：电子工业出版社，2002.

[4] CHEN Y H，CHIEN C H，HUANG C H，et al. Efficient Decoding of Systematic (23，12，7) and (41，21，9) Quadratic Residue Codes [J]. Journal of Information Science and Engineering，2010，26：1831－1843.

[5] LIN T C，LEE H P，CHANG H C，et al. High Speed Decoding of the Binary (47，24，11) Quadratic Residue Code [J].

Information Sciences, 2010, 180: 4060－4068.

[6] LEE H P, CHANG H C. A Memory Improvement on Decoding of the (41, 21, 9) Quadratic Residue Code [J]. International Journal of Computer Theory and Engineering, 2012, 4: 590－594.

[7] ELIA M. Algebraic Decoding of the (23, 12, 7) Golay Codes [J]. IEEE Transactions on Information Theory, 1987, 33: 150－151.

[8] WOLFMANN J. A Permutation Decoding of the (24, 12, 8) Golay code [J]. IEEE Transactions on Information Theory, 1983, 29: 748－750.

[9] HE R, REED I S, TRUONG T K, et al. Decoding the (47, 24, 11) Quadratic Residue Code [J]. IEEE Transactions on Information Theory, 2001, 47: 1181－1186.

On Fast Decoding of (41, 21, 9) Quadratic Residue Code

WU Deng-jie[1], BAO Xiao-min[1], QU Yun-yun[2], YUAN Zhi-hua[1]

1. *School of Mathematics and Statistics, Southwest University, Chongqing 400715, China;*
2. *School of Mathematics Science, Guizhou Normal University, Guiyang 550001, China*

Abstract: The binary QR codes are a family of linear cyclic BCH codes. In this paper, a fast syndrome-weight decoding algorithm (FSWDA) has been presented to decode up to four possible errors in a binary systematic (41, 21, 9) quadratic residue (QR) code. The main idea of FSWDA is based on the property of cyclic codes together with the weight of syndrome difference. The advantage of the FSWDA decoding algorithm over the previous table look-up methods is that it has no need of a look-up table to store the syndromes and their corresponding error patterns in the memory. Moreover, it can be extended to decode all four-error-correcting binary QR codes.

Key words: quadratic residue code; table look-up decoding; error pattern; syndrome; hamming weight

责任编辑 张 枸