An Efficient Erasure Array Code on Vandermonde Matrices

Wunan WAN[1, 2], Wang SUO[1], Yun CHEN[1]

[1]*Information Security Institute, Chengdu University of Information Technology, Chengdu, China,* 610225

[2]*School of Computer Science & Engineering ,University of Electronic Science and Technology of China, Chengdu, China,* 610054

Email: *nan_wwn@hotmail.com, suowang@cuit.edu.cs, chy@cuit.edu.cns*

Abstract: The distribution matrix based on Vandermonde matrices is analyzed, a class of optimizing Vandermonde binary array codes is given. And a method to evaluating good matrices is proposed. The performance of encoding using all manners of Vandermonde codes is optimal. the original Vandermonde codes are improved. Optimizing Vandermonde binary array codes can be used effectively to achieve the reliability and efficient of distributed storage.

Keywords: Vandermonde binary array codes; reliability; erasure array code

一类有效的范德蒙阵列纠删码

万武南[1, 2]，索　望[1]，陈　运[1]

[1]成都信息工程学院信息安全研究所，成都，中国，610225

[2]电子科技大学计算机科学与工程学院，成都，中国，610031

Email: nan_wwn@hotmail.com, suowang@cuit.edu.cns, chy@cuit.edu.cns

摘　要: 对范德蒙码的生成矩阵进行了分析，给出一种把范德蒙线性码转化为阵列码的有效算法。并根据生成矩阵中“1”的个数可判断转换之后范德蒙-阵列纠删码是否具有最佳的编码特性。同时给出一种范得蒙－阵列码译码方法。从理论上分析了基于范德蒙码的阵列纠删码的编译效率，与传统范德蒙纠删码进行了比较，编码效率得到了大大的提高。

关键词: 范德蒙阵列码；可靠性；纠删阵列码

1 引言

随着分布式存储系统规模的扩大，由于网卡、网络电缆中断，或者网络带宽的问题，出现网络I/O 故障，使得多个存储节点同时发生失效的概率比较高，存储系统的可靠性迅速降低。而目前阵列纠删码的容错能力难以解决由于网络I/O 故障的导致数据不可得的问题[1-3]。从存储空间、冗余率及容错能力来说，范德蒙线性码对于解决分布系统可靠性问题提供了一种很好的冗余容错方法。但是由于范德蒙纠删码编译码的时间复杂度为$O(n\log n)O(n^2)$[4]，一但数据量一增大，时间消耗量非常大，影响了其在分布式存储系统的应用。因此降低范德蒙线性码的编译码时间复杂度，是范德蒙线性码在分布式存储系统得到发展的最关键的问题[5-12]。现在有一些方法，使得范德蒙线性码的编译码速度能够降低，例如借助拉格朗日插值等方法用于纠删时RS码的运算复杂度可大大降低[4]。Bloemer，Feng等人针对互联网上大容量数据的传输和实时视频等应用，专门研究了范德蒙纠删码[6-11]，使得编码的计算过程只有异或运算，避免了有限域的运算，降低了范德蒙纠删码的编译码复杂度。

本文利用有限域元素矩阵表示方法，对不同的范德蒙二元矩阵构造出来的阵列纠删码其编码性能进行了分析，给出了范德蒙阵列纠删码的编译码方法，从理论上分析了基于范德蒙码的阵列纠删码的编译效率，与传统范德蒙纠删码进行了比较，编码效率得到了大大的提高。

资助信息：国家自然科学基金资助项目(60873216),成都市科技攻关项目（09GYB972GX-012）

2 范德蒙阵列纠删码

2.1 范德蒙阵列纠删码编码过程

在有限域$GF(2^m)$上，任意一有限域元素都可用$GF(2)$上的$m\times1$的列向量表示，记为$V(e)$。而有限域

978-1-935068-10-5 © 2010 SciRes.

元素也可用$GF(2)$上$m\times m$的矩阵表示，记为Me。矩阵第i列向量相当于$V(2^{i-1}e)$[4]。为了更简单的表示Me，则有限域元素可以表示为一个$GF(2^m)$的二元矩阵集$\widehat{M}=\{0_m, I_m, M_\alpha, M_{\alpha^2}, \cdots, M_{\alpha^{2^{m-2}}}\}=\{0_m, I_m, M, M^2, \cdots, M^{2^{m-2}}\}$，其中$0_m, I_m$为$m\times m$的零矩阵和单位矩阵。

图1给出一个$GF(2^3)$的二元矩阵集$\widehat{M}=\{0_m, I_m, M, M^2, \cdots, M^6\}$实例。

借助有限域元素的矩阵表示，在有限域$GF(2)$上，范德蒙码-阵列纠删码生成矩阵G如下构造。

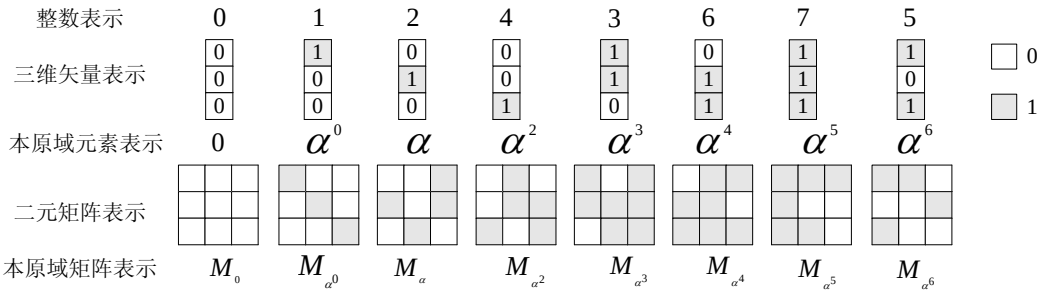

Figure 1. The various representation of Finite Field $GF(2^3)$
图 1. 有限域$GF(2^3)$的各种表示方法

定义1 ((n, k)范德蒙-阵列纠删码生成矩阵G) 令$A=V(a_1,a_2,\cdots,a_k)$为$k\times k$的范德蒙分块方矩阵，$B=V(b_1,b_2,\cdots,b_{n-k})$的$k\times(n-k)$的范德蒙分块矩阵，其中$a_i,b_i\in\widehat{M}$，并且$a_i\neq b_i$。(n, k)范德蒙-阵列纠删码信息拆分矩阵G为：

$$G=A^{-1}\times[A\,|\,B]=[I\,|\,A^{-1}B],$$

$$A=\begin{bmatrix} I_m & M & M^2 & \cdots & M^{k-1} \\ I_m & M^2 & M^4 & \cdots & M^{2(k-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I_m & M^k & M^{2k} & \cdots & M^{k(k-1)} \end{bmatrix}$$

$$B=\begin{bmatrix} M^k & M^{k+1} & M^{k+2} & \cdots & M^{n-1} \\ M^{2k} & M^{2(k+1)} & M^{2(k+2)} & \cdots & M^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M^{k\cdot k} & M^{k(k+1)} & M^{2(k+2)} & \cdots & M^{k(n-1)} \end{bmatrix}$$

G生成矩阵，具有下面几个性质：

1) G中的任意子方阵非退化(可逆)；

2) $A^{-1}B$中任意的子方矩阵非退化。

定理 1：二元矩阵集$\widehat{M}$任意r个二元矩阵构造范得蒙分块矩阵$\widehat{G_r}$是满秩矩阵。设范得蒙分块矩阵$\widehat{G_r}$的r列分别为$x_0, x_1, \cdots, x_{r-1}$,其$M^{x_i}\in\widehat{M}$

$$\widehat{G_r}=\begin{bmatrix} M^{x_0} & M^{x_1} & M^{x_2} & \cdots & M^{x_{r-1}} \\ M^{2x_0} & M^{2x_1} & M^{2x_2} & \cdots & M^{2x_{r-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M^{rx_1} & M^{rx_1} & M^{rx_2} & \cdots & M^{rx_r} \end{bmatrix}$$

证明：$\widehat{G_r}$左乘下面的系列矩阵：

$$\begin{bmatrix} I_m & 0_m & \cdots & 0_m & 0_m \\ 0_m & I_m & \cdots & 0_m & 0_m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_m & 0_m & \cdots & I_m & 0_m \\ 0_m & 0_m & \cdots & M^{x_{r-1}} & I_m \end{bmatrix}\times\cdots\times\begin{bmatrix} I_m & 0_m & \cdots & 0_m & 0_m \\ 0_m & I_m & \cdots & 0_m & 0_m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_m & 0_m & \cdots & I_m & 0_m \\ 0_m & 0_m & \cdots & M^{x_1} & I_m \end{bmatrix}$$

$$\times\begin{bmatrix} I_m & 0_m & \cdots & 0_m & 0_m \\ M^{x_0} & I_m & \cdots & 0_m & 0_m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_m & 0_m & \cdots & I_m & 0_m \\ 0_m & 0_m & \cdots & M^{x_0} & I_m \end{bmatrix}\times\widehat{G_r}$$

$\widehat{G_r}$矩阵变成如下一个矩阵

$$\begin{bmatrix} M^{x_0} & M^{x_1} & \cdots & M^{x_{r-2}} & M^{x_{r-1}} \\ 0 & M^{x_1}(M^{x_1}+M^{x_0}) & \cdots & M^{x_{r-2}}(M^{x_{r-2}}+M^{x_0}) & M^{x_{r-1}}(M^{x_{r-1}}+M^{x_0}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_m & 0_m & \cdots & M^{x_{r-2}}\prod_{i=0}^{r-2}(M^{x_i}+M^{x_{i+1}}) & M^{x_{r-1}}\prod_{i=0}^{r-2}(M^{x_i}+M^{x_{i+1}}) \\ 0_m & 0_m & \cdots & 0 & M^{x_{r-1}}\prod_{i=0}^{r-1}(M^{x_i}+M^{x_{i+1}}) \end{bmatrix}$$

因为 $M^{x_{r-1}}\prod_{i=0}^{r-1}(M^{x_i}+M^{x_{i+1}})\in\widehat{M}$ 为可逆阵，其秩为m，因此而 $\widehat{G_r}$ 矩阵左乘系列矩阵之后的秩为rm，因此 $\widehat{G_r}$ 的秩为rm，即矩阵 $\widehat{G_r}$ 是满秩矩阵。

证毕

$(n, k)/(m, l)$范德蒙-阵列纠删码每个码字C为一个 $m\times n$ 的二维阵列，其中 $0,1,2,\cdots,k-1$ 列为信息列，$k,k+1,\cdots,n$ 列为校验列。每列有 m 块长度为 $l$bits 的数据块。与一般的阵列纠删码不同，范德蒙-阵列纠删码没有明显的几何特性，因此我们通过代数方法来定义范德蒙-阵列纠删码的编码过程。利用生成矩阵G，则$(n, k)/(m, l)$范德蒙-阵列纠删码的码字C如下表示：

定义 2：$(n, k)/(m, l)$范德蒙-阵列纠删码的码字C) $C=\{(c=(\vec{c_0},\vec{c_1},\cdots,\overrightarrow{c_{n-1}}))\}$，$\vec{c_i}=(c_{0,i},c_{1,i},\cdots,c_{m-1,i})^T$，$\vec{c_i}$ 为每列信息符，则 $0\le i\le k-1$ 时，$\vec{c_i}$ 为信息列，则 $k\le i\le n-1$ 时，$\vec{c_i}$ 为校验列；则 $c_{i,j}$ 表示位为第 i 行第 j 列上长度为l的数据块。

定义 3：$(n, k)/(m, l)$范德蒙-阵列纠删码代数定义) $(n, k)/(m, l)$范德蒙-阵列纠删码 $\vec{c_0}^T,\vec{c_1}^T,\cdots,\overrightarrow{c_{k-1}}^T$ k 个(源信息)列向量，可以计算出范德蒙-阵列纠删码的n-k个校验向量 $\vec{c_k}^T,\overrightarrow{c_{k+1}}^T,\cdots,\overrightarrow{c_{n-1}}^T$。若生成矩阵 $[I\mid A^{-1}B]$，令 $\widehat{G_r}=A^{-1}B$ 即编码过程如下：

$$\left[\vec{c_k}^T,\overrightarrow{c_{k+1}}^T,\cdots,\overrightarrow{c_{n-1}}^T\right]=\left[\vec{c_0}^T,\vec{c_1}^T,\cdots,\overrightarrow{c_{k-1}}^T\right]\times\widehat{G_r}$$

下面首先编译码过程中需要采用的“bit-vector”相乘运算定义。

定义 4：(“bit-vector”相乘运算) [53, 117] 令v是一个$GF(2)$的二进制变量，而 $A^{'}$ 是一个多比特的向量，则“bit-vector”相乘运算如下所示：

$$v\bullet A^{'}=\begin{cases}0 & v=0\\ A^{'} & v=1\end{cases}$$

根据范德蒙码-阵列纠删码的代数定义，下面给出一个的实例。

例1. $(n, k)/(m, l)$范德蒙-阵列纠删码编码实例) 令 $m=3$，$n=6$，$k=3$，则 $GF(2^3)$ 二元矩阵集 $\widehat{M}=\{0_m,I_m,M,M^2,\cdots,M^6\}$，如图 1 所示。若 $A=V(I_m,M,M^2)$，$B=V(M^3,M^4,M^5)$，得到 $\widehat{G_r}=A^{-1}B$ 如下所示：

$$\widehat{G_r}=\begin{bmatrix}M^6 & M^5 & M^5\\ M & M^2 & M^4\\ M^6 & M^6 & M^3\end{bmatrix}$$

校验列 $\vec{c_3}^T,\vec{c_4}^T,\vec{c_5}^T$ 则可以如下计算得到：

$$\left[\vec{c_3}^T,\vec{c_4}^T,\vec{c_5}^T\right]=\left[\vec{c_0}^T,\vec{c_1}^T,\vec{c_2}^T\right]\times\widehat{G_r}$$

则$(6,3)/(3, l)$范德蒙-阵列纠删码如下表 1 所示：

若 $A=V(I_m,M,M^2)$，$B=V(M^3,M^4,M^6)$，得到 $\widehat{G_r}=A^{-1}B$ 如下所示：

$$\widehat{G_r}=\begin{bmatrix}M^6 & M^5 & M^5\\ M & M^2 & I\\ M^6 & M^6 & M\end{bmatrix}$$

校验列 $\vec{c_3}^T,\vec{c_4}^T,\vec{c_5}^T$ 则可以如下计算得到：

$$\left[\vec{c_3}^T,\vec{c_4}^T,\vec{c_5}^T\right]=\left[\vec{c_0}^T,\vec{c_1}^T,\vec{c_2}^T\right]\times\widehat{G_r}$$

则$(6,3)/(3, l)$范德蒙-阵列纠删码如下表2所示：

Table 1. $(6,3)/(3, l)$ Coding of Vandermonde Erasue Array Code

表1. $(6,3)/(3, l)$范德蒙-阵列纠删码编码

c_{00}	c_{01}	c_{02}	$c_{00}\oplus c_{20}\oplus c_{11}\oplus c_{02}\oplus c_{22}$	$c_{00}\oplus c_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{22}$	$c_{00}\oplus c_{10}\oplus c_{20}\oplus c_{11}\oplus c_{21}\oplus c_{02}\oplus c_{02}$
c_{10}	c_{11}	c_{12}	$c_{00}\oplus c_{21}\oplus c_{02}$	$c_{00}\oplus c_{20}\oplus c_{01}\oplus c_{11}\oplus c_{02}$	$c_{00}\oplus c_{20}\oplus c_{01}\oplus c_{11}\oplus c_{21}\oplus c_{12}\oplus c_{22}$
c_{20}	c_{21}	c_{22}	$C_{10}\oplus c_{01}\oplus c_{12}$	$c_{00}\oplus c_{11}\oplus c_{21}\oplus c_{12}$	$C_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{12}$

Table 2. $(6,3)/(3, l)$ Coding of Vandermonde Erasue Array Code

表2. $(6,3)/(3, l)$范德蒙-阵列纠删码编码

c_{00}	c_{01}	c_{02}	$c_{00}\oplus c_{20}\oplus c_{11}\oplus c_{02}\oplus c_{22}$	$c_{00}\oplus c_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{22}$	$C_{20}\oplus c_{01}\oplus c_{12}$
c_{10}	c_{11}	c_{12}	$c_{00}\oplus c_{21}\oplus c_{02}$	$c_{00}\oplus c_{20}\oplus c_{01}\oplus c_{11}\oplus c_{02}$	$c_{00}\oplus c_{10}\oplus c_{11}\oplus c_{22}$
c_{20}	c_{21}	c_{22}	$C_{10}\oplus c_{01}\oplus c_{12}$	$c_{00}\oplus c_{11}\oplus c_{21}\oplus c_{12}$	$C_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{12}$

978-1-935068-10-5 © 2010 SciRes.

从(*n*, *k*)/(*m*, *l*)范德蒙-阵列纠删码的编码实例可以看出，范德蒙阵列纠删码的编码过程只需要简单的异或运算。但是不同的生成矩阵，其编码需要总的异或次数不同。

2.2 范德蒙阵列纠删码译码过程

根据范德蒙-阵列纠删码编码过程可知，校验列的构造过程没有明显的几何特性，因此不能借助码的几何特性来快速译码算法。在这一节将给出范德蒙-阵列纠删码代数译码算法。设(*n*, *k*)/(*m*, *l*)范德蒙-阵列纠删码的一个码字.

$C=\{(c=(\vec{c_0},\vec{c_1},\cdots,\vec{c_{n-1}}))\}$ 中，其中 $\vec{c_{\mu_i}}$，$i=0,1,2,\cdots,t-1$ 丢失，则接收到的码字为 $y=\{\vec{y_0},\vec{y_1},\cdots,\vec{y_{k-1}},\vec{y_k},\cdots,\vec{y_{n-1}}\}$，则

$$\vec{y_i}=\begin{cases}\vec{c_i}, & i\notin\{\mu_0,\mu_1,\cdots,\mu_{t-1}\}\\ 0, & i\in\{\mu_0,\mu_1,\cdots,\mu_{t-1}\}\end{cases}$$

根据编码理论可知，(*n*, *k*)/(*m*, *l*)范德蒙-阵列纠删码的校验矩阵:

$H=\left[(A^{-1}B)^T\mid I_{n-k}\right]=\left[G^T\mid I_{n-k}\right]$，因此可以得 $Hy^T=s^T$，其中 $s=\left(\vec{s_0},\vec{s_1},\vec{s_2},\cdots,\vec{s_{n-k-1}}\right)$，$\vec{s_i}=\left(s_{0,i},s_{1,i},\cdots,s_{m-1,i}\right)$

选择 t 个 $\vec{s_{v_i}}$，$v_i\in\{0,1,2,\cdots,(n-k-1)\}$ 组成非零列向量 $s'=\left(\vec{s_{v_0}},\vec{s_{v_1}},\vec{s_{v_2}},\cdots,\vec{s_{v_{t-1}}}\right)$。

则 $\widehat{H_t}$ 由校验矩阵 H 的 $\mu_0,\mu_1,\cdots,\mu_{t-1}$ 列和 $v_0,v_1,\cdots,v_{t-1}$ 行组成的分块方矩阵。因而 $\widehat{H_t}\left(\vec{c_{\mu_0}},\vec{c_{\mu_1}},\cdots,\vec{c_{\mu_{t-1}}}\right)^T=(s')^T$。矩阵求逆即可译码 $\left(\vec{c_{\mu_0}},\vec{c_{\mu_1}},\cdots,\vec{c_{\mu_{t-1}}}\right)^T=\widehat{H_t}^{-1}(s')^T$

例2. ((6,3)=(3, *l*)范德蒙-阵列纠删码译码实例)采用例子1，则设第1，2信息列丢失，即 $\mu_0=1$，$\mu_1=2$.

Table 3. (*6,3*)/(*3, l*) Decoding of Vandermonde Erasue Array Code
表3. (*6,3*)/(*3, l*)范德蒙-阵列纠删码译码

c_{00}	?	?	$c_{00}\oplus c_{20}\oplus c_{11}\oplus c_{02}\oplus c_{22}$	$c_{00}\oplus c_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{22}$	$c_{00}\oplus c_{10}\oplus c_{20}\oplus c_{11}\oplus c_{21}\oplus c_{02}\oplus c_{02}$
c_{10}	?	?	$c_{00}\oplus c_{21}\oplus c_{02}$	$c_{00}\oplus c_{20}\oplus c_{01}\oplus c_{11}\oplus c_{02}$	$c_{00}\oplus c_{20}\oplus c_{01}\oplus c_{11}\oplus c_{21}\oplus c_{12}\oplus c_{22}$
c_{20}	?	?	$C_{10}\oplus c_{01}\oplus c_{12}$	$c_{00}\oplus c_{11}\oplus c_{21}\oplus c_{12}$	$C_{10}\oplus c_{20}\oplus c_{21}\oplus c_{02}\oplus c_{12}$

1) 首先根据 $Hy^T=s^T$，计算出 $s=\left(\vec{s_0},\vec{s_1},\vec{s_2}\right)$，$\vec{s_i}=\left(s_{0,i},s_{1,i},s_{2,i}\right)$；

2) 选择 $\widehat{H_2}$ 矩阵如下：

$$\widehat{H_2}=\begin{bmatrix}M^T & (M^6)^T\\ M_1 & M^T\end{bmatrix}$$

$\widehat{H_2}\left[\vec{c_1},\vec{c_2}\right]^T=\left[\vec{s_1},\vec{s_2}\right]^T$，然后依次左乘下面的矩阵：

$$\begin{bmatrix}I_3 & M^T\\ 0_3 & M^T\end{bmatrix}\times\begin{bmatrix}I_3 & 0_3\\ (M^6)^T & I_3\end{bmatrix}$$

3）求解出 $\vec{c_1},\vec{c_2}$，译码结束。

$$\left[\vec{c_1},\vec{c_2}\right]^T=\begin{bmatrix}I_3 & M^T\\ 0_3 & M^T\end{bmatrix}\times\begin{bmatrix}I_3 & 0_3\\ (M^6)^T & I_3\end{bmatrix}\times\left[\vec{s_1},\vec{s_2}\right]^T$$

3 范德蒙阵列纠删码信息拆分矩阵 G

3.1 信息拆分矩阵 G

阵列纠删码的编译码复杂度主要是异或运算，因此(n, k)范德蒙-阵列纠删码生成矩阵G中“1”的个数决定了编码的复杂度，则定义*o*为信息生成矩阵*G*每行“1”的个数的平均值，来表示范德蒙-阵列纠删码的编码复杂度，以及衡量 G 的性能。根据矩阵 G 的构造，选择不同的 A，B 矩阵，G 中“1”的个数是不一样的。从实例1可以看出，若 $A=V(I_m,M,M^2)$，$B=V(M^3,M^4,M^5)$ 构造出 $G1$，则A相同，令 $B=V(M^3,M^4,M^6)$ 则构造出 $G2$，两个矩阵其 o 分别等于o = 5.222和o = 4.3333，*G*2的异或次数是差不多比*G*1少了20%，因此采用*G*2作为(6, 3)范德蒙-阵列纠删码的生成矩阵，其编码复杂度要低。因此*G*2相对于*G*1来说是“好矩阵”。

在生成矩阵*G*的构造中，m 阶的二元矩阵集 $\widehat{M}=\{0_m,I_m,M,M^2,\cdots,M^{2^{m-2}}\}$ 对于给定的*n*, *k*范德蒙-阵列纠删码总共有 $C\begin{pmatrix}2^{m-1}\\ n\end{pmatrix}$ 个不同的信息拆分矩阵*G*，*o*最小值的矩阵*G*，为最优的信息拆分矩阵。

若范德蒙-阵列纠删码的生成矩阵选择最优的信

息拆分矩阵G，则编码复杂度最低。图2 给出随着*m, n, k*取值不同，*o*取最大值和最小值，以及传统的方法，*G*的*o*值的趋势变化图。

从图2 中 *o* 随着趋势图可知：传统方法构建的信息拆分矩阵 *G* 并不是最佳的；若 *m*，*n* 固定，随着 *k* 值的变大，*o* 都是逐渐变小，*o* 的最小值与传统的方法的 *o* 的之间相差变大，即范德蒙-阵列纠删码的总列数 *n* 不变，信息列*k*变大，校验列 *n, k* 变小，传统方法选择的信息拆分矩阵 *G* 与最佳的信息拆分矩阵 *G* 之间的性能相差逐渐变大；第二就是对于相同的 *m, n, k* 值，随着 *n* 值的增大，*o* 的值逐渐变小；第三个特性就是，对于 *n, k* 相同，*m* 值越大，*o* 的值越小，从一维来看，也就是码长和信息列固定，则二元矩阵的阶数越大，则*o*的值越小。

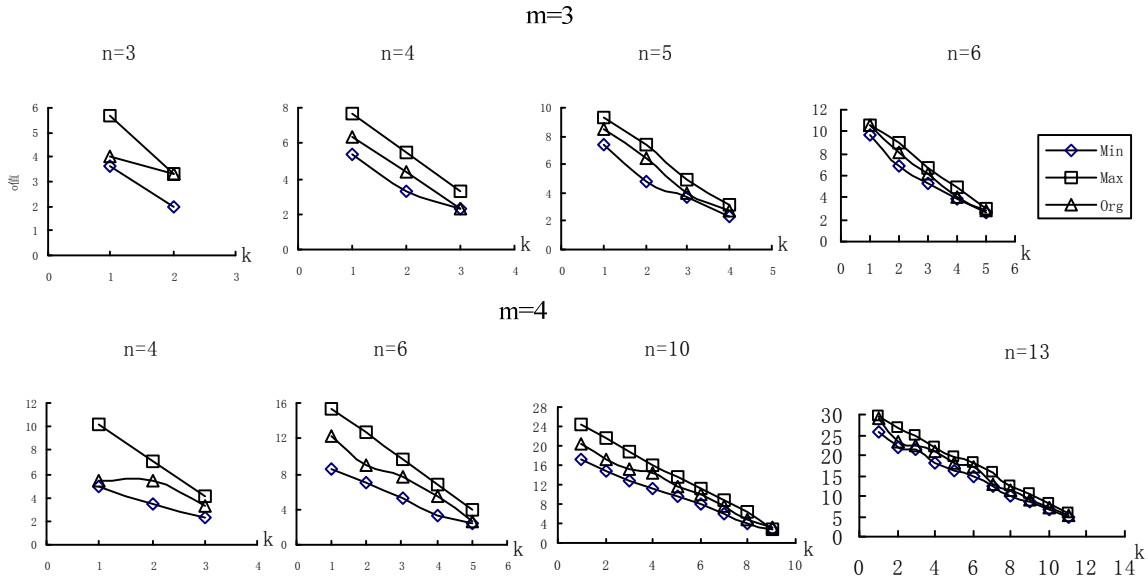

Figure 2. Curve: GVC、SVC、BVC

图2. 最好、最差、传统范德蒙二元矩阵*o*值趋势图

3.2 信息拆分矩阵 G 性能分析

由不同的范德蒙二元矩阵构造出来的范得德蒙阵列纠删码其编码复杂度并不同。主要基于最佳生成矩阵的范德蒙阵列纠删码 (GVC Good Vandermonde code)、标准的范德蒙阵列纠删码 (SVC Standard-Vandermond code)、最差的阵列纠删码 (BVC Bad Vandermond code) 的性能进行分析。

范德蒙-阵列纠删码的编码过程只需要异或运算，因此要比较三种不同的矩阵构造的范德蒙阵列纠删码的编码复杂度，可用参数*o*来表示范德蒙-阵列纠删码的编码复杂度。图3给出了三种不同范德蒙二元矩阵的MDS码，与XEOD码的编码性能比较图。从上图3中可以看出，而基于最佳信息拆分矩阵的(*n*，*k*)/(*m*,*l*)范德蒙阵列纠删码的编码特性最好，并随着*r*值的变大，其*o*与XEOD码的*o*的差距逐渐缩小。

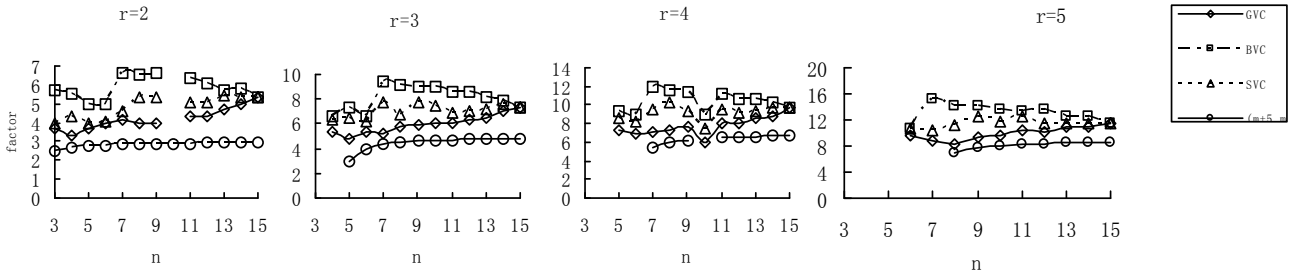

Figure 3. Coding performance of Vandermonde Array Code

图3. 范德蒙阵列纠删码编码性能

978-1-935068-10-5 © 2010 SciRes.

下图4给出了三种不同码率下，k/n=1/2，k/n=1/3，k/n=2/3，GVC、SVC、BVC三种不同范德蒙二元矩阵构造得范德蒙-阵列纠删码编码的性能比较。从图中可知，码率为R = 1/3时，GVC码比SVC码、BVC码的三编码性能相差较大， 但是随着码率R的增大，GVC、SVC、BVC，SVC与GVC相比，o逐渐相差较小。

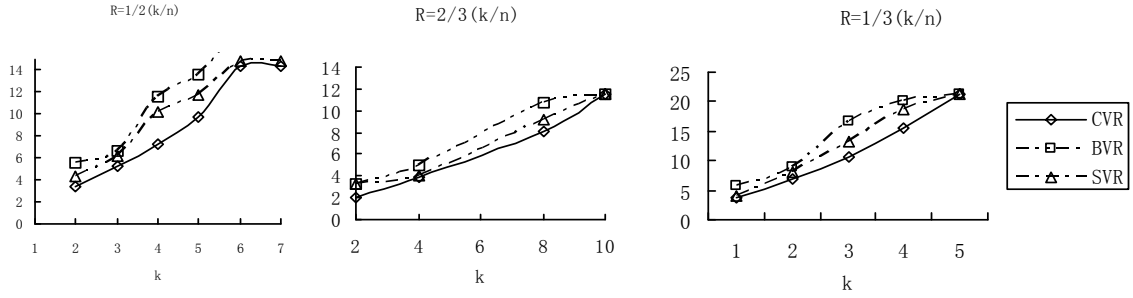

Figure 4. Coding performance of Vandermonde Array Code based on Different Rate

图 4. 不同码率范德蒙阵列纠删码编码性能

4 结论

利用有限域元素的矩阵表达方法，提出了一类纠错能力强的范德蒙－阵列纠删码；并对范德蒙-阵列纠删码的生产矩阵进行了深入研究，给出了如何选择最优范德蒙-阵列纠删码的生产矩阵构造范德蒙-阵列纠删码，并通过实验列举出好的生成矩阵。同时给出了译码算法。

References (参考文献)

[1] BLAUM M, BRADY J, BRUCK J, et al. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures[J]. IEEE Trans Comput, 1995, 44(2): 192–202.

[2] HAFNER J. L. WEAVER Codes: Highly fault tolerant erasure codes for storage systems[C]. In *FAST-2005: 4th Usenix Conference on File and Storage Technologies*, pages 211–224, San Francisco, December 2005.

[3] HAFNER J. L. HoVer erasure codes for disk arrays[C]. In *DSN-2006: The International Conference on Dependable Systems and Networks*, Philadelphia, June 2006. IEEE.

[4] NONNENMACHER J., BIERSACK E. W., et al. Parity based loss recovery for reliable multicast[J]. IEEE/ACM Trans. Networking, vol. 6, pages 349–361, Aug.

[5] UYEDA F., et al. Chien. Evaluation of a High Performance Erasure Code Implementation[R]. UCSD Technical Report CS2004–0798, 2004.

[6] BLOEMER J., et al. An XOR-Based Erasure-Resilient Coding Scheme[R]. ICSI TR-95-048, Technical report at ICSI, August 1995.

[7] FENG G., DENG R., et al. New efficient MDS array codes for RAIDPart II: Rabin-like codes for tolerating multiple("4) disk failures[J]. *IEEE Transactions on Computers*, 54(12): 1473–1483, Decemeber 2005.

[8] GREENAN K., WYLIE J. J. On the reliability of XOR based erasure codes[R]. Technical report, HP, February 2008.

[9] WYLIE J. J., et al. Determining fault tolerance of XOR-based erasure codes efficiently[C]. In *DSN-2007*, pages 206–215. IEEE, June 2007.

[10] HAFNER J. L., et al. Matrix methods for lost data reconstruction in erasure codes[C]. In *FAST-2005: 4th Usenix Conference on File and Storage Technologies*, pages 183–196, San Francisco, December 2005.

[11] PLANK J. S. The RAID-6 Liberation codes[C]. In *FAST-2008: 6th Usenix Conference on File and Storage Technologies*, pages 97–110, San Jose, February 2008.