New Efficient Constructions of Binary Asymmetric Error-Correcting Codes

Han-Chang Liang
Department of Computer Science & Information Engineering
National Chiao Tung University
hcliang@csie.nctu.edu.tw

Jen-Chun Chang
Department of Computer Science & Information Engineering
National Taipei University
jcchang@csie.nctu.edu.tw

Rong-Jaye Chen
Department of Computer Science & Information Engineering
National Chiao Tung University
rjchen@csie.nctu.edu.tw

Abstract - *We study the new construction of binary asymmetric error-correcting codes presented by Fu, Ling and Xing. The direct construction algorithm requires $\theta(n2^n)$ operations in all cases. In this paper, we first develop a construction algorithm which requires only $\theta(2^n)$ operations in all cases. Next, we improve the algorithm by a bounding function. The final construction algorithm requires $O(2^n)$ operations in the worst case. In most cases, the number of operations is much lower than 2^n.*

Keywords: Asymmetric error-correcting codes, code construction, backtracking algorithm.

1. Introduction

In most binary communication systems, the error probabilities from 1 to 0 and from 0 to 1 are approximately the same. This kind of systems is well modeled by binary-symmetric channel. But in certain communication systems, the probability from 1 to 0 is much higher than the error probability from 0 to 1. These communications are modeled by the binary asymmetric channel, which are also named Z-channel. Similar to error-correcting codes for binary-symmetric channel, error-correcting codes for Z-channel are also discussed widely [1-6]. Recently, a new construction for asymmetric error-correcting codes was developed by Fu, Ling and Xing [3]. Their construction provided new lower bounds on code size.

In this paper, we present two construction algorithms based on those developed by Fu, Ling and Xing.

This paper is organized as follows: In Section 2, we introduce some definitions about binary asymmetric code. In Section 3, we introduce the new binary asymmetric error-correcting code construction introduced by Fu et al.

This work was supported by the National Science Council, R.O.C. , under contracts no. NSC 93-2213-E-009-010- and no. NSC93-2213-E-305-003-.

In Section 4, we first present a backtracking algorithm for constructing asymmetric codes. Then we provide a bounding function to improve this algorithm. In Section 5, we analyze all the construction algorithms discussed in this paper.

We conclude this section by introducing the following notations which will be used throughout this paper.

1. F_q : A finite field with q elements.
2. F_2^n : $\{x=(x_1,x_2,...,x_n) \mid x_i \in F_2\}$, a vector space over F_2 of dimension n.
3. $F_q[x]$: The ring of polynomials over F_q in variable x .

2. Binary Asymmetric Codes

A binary asymmetric error-correcting code is defined in terms of the following notations.

For binary vectors

$$x=<x_1,x_2,...,x_n> \text{ and } y=<y_1,y_2,...,y_n>,$$

the asymmetric distance between them is defined as

$$d_a(x,y)=\max\{N(x,y),N(y,x)\},$$

where

$$N(x,y)=\#\{i \mid x_i=0, y_i=1\}.$$

For $C \subseteq F_2^n$, the minimum asymmetric distance of C is defined as

$$\Delta(C)=\min\{d_a(x,y): x,y \in C, \text{and } x \neq y\}.$$

A binary code of length n and minimum asymmetric distance Δ is called a (n, Δ) asymmetric code.

It was shown in [4] that a (n, Δ) asymmetric code can correct Δ-1 or fewer asymmetric errors (from 1 to 0 errors).

3. The Fu, Ling and Xing's Construction

By Fu, Ling and Xing's construction, a (n, Δ≥d) asymmetric error-correcting code can be constructed in the following steps:

Step1:
Select a finite field F_q such that q is a prime power, and q≥n.

Step2:
Select a monic polynomial $f(x) \in F_q[x]$ with degree d.

Step3:
Select n distinct elements $\alpha_1, \alpha_2, ..., \alpha_n$ in F_q such that $f(\alpha_i) \neq 0$, for $1 \leqq i \leqq n$.

Step4:
Define a multiplicative group $(G, \otimes)$, where

$$G = \{g(x) \in F_q[x]: \deg(g(x)) < \deg(f(x)), \; g(x) \text{ is monic, and } (g(x), f(x)) = 1\}.$$

The multiplication operation $\otimes$ over G given by $a(x) \otimes b(x) = M(a(x)b(x) \bmod f(x))$, where $M(h(x)) = h_m^{-1}h(x)$, h_m is the coefficient of the highest degree term in h(x).

Step5:
Define $\Omega: F_2^{\ n} \rightarrow G$

$$(c_1, c_2, ..., c_n) \mapsto \prod_{i=1}^{n} \otimes (x - \alpha_i)^{c_i} \in G.$$

Step6:
Select a polynomial $g(x) \in G$.
Let $C_g = \Omega^{-1}(g(x))$, if $C_g \neq \phi$, then C_g is a (n, $\Delta \geq$ d) asymmetric code.

4. The New Recursive Construction

With Fu, Ling and Xing's construction, we have to compute Ω^{-1} function to obtain the code. In the direct construction algorithm, it is necessary to compute $\Omega(v)$, for all $v \in Z_2^{\ n}$, and then collect the set $\{v \mid \Omega(v) = g(x), v \in F_2^{\ n}\}$ as a code C_g. In this section, we propose two recursive algorithms to speed up the computations in all cases.

First, we define a set

$$C_{g,i} = \{v \mid v \in C_g, v_j = 0 \text{ for } n - i + 1 \leq j \leq n\}.$$

Note that $C_{g,0} = C_g$. The main idea of this algorithm is to compute $C_{g,0}$, instead of C_g.

Our first algorithm is given below:

Algorithm Construction_1 (*g(x)*, *i*)
Input: *g(x)*, *i*
Output: C_g
begin
  Initially T_1,T_2 are two empty sets;
  if *i* = n-1 **then**
    if *g(x)* = (x-α_1) **then**
      return {<1,0,0,…,0>};
    else if *g(x)* = 1 **then**
      return {<0,0,0,…,0>};
    else
      return {};
    endif
  endif
  T_1 = *Construction_1*(*g(x)*, *i*+1);
  T_2 = *Construction_1*(*g(x)*$\otimes$(x-α_{n-i})$^{-1}$, *i*+1);
  for all vectors *v* in T_2 **do**
    v_{n-i}=1;
  return $T_1 \cup T_2$;
end

In the call tree of construction 1, we notice that several branches receive empty-set return. We introduce a bounding function $\lambda: F_q \rightarrow N$ to help eliminating sub-trees of the call tree.

We define a λ-function

$$\lambda(\alpha_i) = \begin{cases} \min\{j : |C_{(x-\alpha_i),j}| = 1\} & \text{, if exist some } j \ni |C_{(x-\alpha_i),j}| = 1. \\ n & \text{, otherwise.} \end{cases}$$

Example1 Let $F_q = F_{13}$, n=13, $f(x) = x^4 + 11x^3 + 12x^2 + 2x + 1$, $\alpha_1 \sim \alpha_{13} = 0 \sim 12$,

then we have the following λ function:

α_i	0	1	2	3	4	5
$\lambda(\alpha_i)$	3	3	7	3	1	2

α_i	6	7	8	9	10	11	12
$\lambda(\alpha_i)$	2	4	2	2	1	13	13

The second algorithm given below is improved from construction 1 by applying the λ-function.

Algorithm Construction_2 (*g(x)*, *i*)
Input: *g(x)*, *i*
Output: C_g
begin
  Initially T_1,T_2 are two empty sets;
  if *i* = n-1 **then**
    if *g(x)* = (x-α_1) **then**
      return {<1,0,0,…,0>};
    else if *g(x)* = 1 **then**
      return {<0,0,0,…,0>};
    else
      return {};
    endif
  endif
  if *g(x)* = (x-α_j) for some *j* **then**
    if $i \geq \lambda(\alpha_j)$ and $i \leq$ n-*j* **then**
      return {*v*} where *v* is the j_{th} row of identity matrix $I_{n \times n}$;
    else if $i \geq \lambda(\alpha_j)$ and *i* > n-*j* **then**
      return {};

endif
endif
T_1 = *Construction_2(g(x), i*+1);
T_2 = *Construction_2(g(x)*$\otimes$(x-α_{n-i})$^{-1}$, *i*+1);
for all vectors *v* in T_2 **do**
v_{n-i}=1;
return $T_1 \cup T_2$;
end

5. Analysis of The Construction Algorithms

The most expensive operation in all construction algorithms is the multiplication operation over the group $(G,\otimes)$. We analyze all the construction algorithms on the number of $\otimes$ operations that have been discussed so far.

Theorem 1: The direct construction algorithm proposed by Fu, Ling, and Xing requires $\theta(n2^n)$ multiplication operations over group G in all cases.

Proof: The direct algorithm has to calculate a $\otimes$ operation for each "1" occurs in each binary vector from $F_2^{\ n}$. Note that we need not calculate the $\otimes$ operation for the first "1" in each binary vector. Thus, for $n \geq 2$, the number of $\otimes$ operation is

$$\sum_{i=0}^{n}\binom{n}{i}\times i-(2^n-1)$$
$$=(n-2)2^{n-1}+1$$
$$=\theta(n2^n).$$

□

Theorem 2: The Construction 1 algorithm requires $\theta(2^n)$ multiplication operations over group G in all cases.

Proof: Consider a call tree of the construction 1 algorithm, it has exact 2^n-1 nodes. We have to calculate a $\otimes$ operation on each internal node in the call tree. By pre-calculating

$$(x-\alpha_i)^{-1} \text{ over } G,\ \text{ for } 1 \leq \text{i} \leq \text{n},$$

the construction 1 algorithm has to calculate the $\otimes$ operation on $2^{n-1}-1$ nodes in the call tree. So the number of $\otimes$ operations is $2^{n-1}-1=\theta(2^n)$. □

Theorem 3: The Construction 2 algorithm requires $O(2^n)$ multiplication operations over group G in the worst case.

Proof: By the bounding function λ, we eliminate at least

$$\sum_{\substack{c:c\in C_g,wt(c)\geq 2\\ i=\min\{i:c_i=1\},\,j=\min\{j:j>i,c_j=1\},\\ n-j+1\geq\lambda(\alpha_i)}}(2^{j-1}-2)$$

nodes in the complete call tree. So, the call tree has at most

$$2^n-1-(\sum_{\substack{c:c\in C_g,wt(c)\geq 2\\ i=\min\{i:c_i=1\},\,j=\min\{j:j>i,c_j=1\},\\ n-j+1\geq\lambda(\alpha_i)}}(2^{j-1}-2))$$

nodes.

Thus we have an upper bound of internal nodes

$$2^n-1-(\sum_{\substack{c:c\in C_g,wt(c)\geq 2\\ i=\min\{i:c_i=1\},\,j=\min\{j:j>i,c_j=1\},\\ n-j+1\geq\lambda(\alpha_i)}}(2^{j-1}-1)).$$

Since we have to calculate a $\otimes$ operation on each internal node. In worst cases, the number of $\otimes$ operations is $O(2^n)$. □

Example2 Using parameters in example1, in order to construct $C_{(x-10)}$, the number of $\otimes$ operations required by each algorithm is:

Algorithm	Direct	Constuct1	Construct2
# of operations	45057	4095	2043

As shown in Figure1, several branches are eliminated by λ-function (the shadowed area).

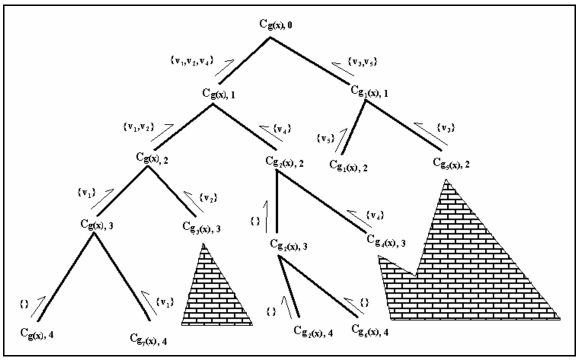

Figure 1. Call tree of construction 2

References

[1] S. Al-Bassam and S. Al-Muhammadi, "A Single asymmetric error-correcting code with 2^{13} codewords of dimension 17," *IEEE Trans. Inform. Theory*, vol. 46, pp.269-271, Jan, 2000.
[2] B. Bose and S. Al-Bassam, "On systematic asymmetric error-correcting codes," *IEEE Trans. Inform. Theory,* vol. 46, pp.669-672, Mar, 2000.
[3] Fang-Wei Fu, San Ling, and Chaoping Xing, "New Lower Bounds and Constructions for Binary Codes Correcting Asymmetric Errors," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp.3294-3299, December, 2003.
[4] T. Klove, "Error correcting codes for the asymmetric channel," Dept. Mathematics, Univ. Bergen, Bergen, Norway, Tech. Rep.18-09-07-81, 1995.
[5] T. R. N Rao and A. S. Chawla, "Asymmetric error codes for some LSI semi-conductor memories," in *Proc. Annu. Southeastern Symp. Systems Theory*, 1975.
[6] J. P. Robinson, "An asymmetric error-correcting ternary code," *IEEE Trans. Inform. Theory*, vol. IT-24, pp.258-261, Mar, 1978.