ERROR CORRECTING CODE USING LATIN SQUARE

Ali Makki Sagheer **Makarim Abdul-Waheed. Abdul-Jabbar**

College of Computer, University of Anbar

ARTICLE INFO

Received: 20 / 5 /2008
Accepted: 12 / 10 /2008
Available online: 14/6/2012
DOI: 10.37652/juaps.2008.15649

Keywords:
Error Correcting Code,
Error Control Code,
Latin Square,
Networking.

ABSTRACT

Digital data stored in computers or transmitted over computer networks are constantly subject to error due to the physical medium in which they are stored or transmitted. Error-correction codes are means of introducing redundancy in the data so that even if part of it is corrupted or completely lost, the original data can be recovered. Error correcting codes are used in modern technology to protect information from errors. Burst error correcting codes are needed in virtually uncountable applications. Such codes will be called complete burst error correcting codes. There are quite a few constructions for complete burst error correcting codes. This paper presents an error correcting code based on the concept and the theory of the Latin Squares, where it employ the characteristics of the orthogonal Latin Squares to correct the errors. That is not complete burst error correcting codes, since it can correct most burst pattern of length $i \leq n$, but not all of them. However, if the number of uncorrectable patterns is sufficiently small, this code can be used in practice as a burst error correcting code.

Introduction

Information is passes every day in our society. It is essential that interference in the communication of this information hinders the information from being received as little as possible. Error-correcting codes provide us with this ability. Error-correcting codes allow us to receive a piece of information, identify any errors, locate them, and correct them.

Cyclic codes are an especially useful kind of error-correcting code, and BCH codes and QR codes are especially useful kinds of cyclic codes. Error-correcting code theory has also been used in areas outside of information communication. Error correcting code theory is an important subject to study. A digital message is a sequence of 0's and 1's which encodes a given message.

More data will be added to a given binary message that will help to detect if an error has been made in the transmission of the message; adding such data is called an error-detecting code. More data may also be added to the original message so that errors made in transmission may be detected, and also to figure out what the original message was from the possibly corrupt message that was received.

Development of algorithmically efficient error correcting codes has attracted attention of engineers, computer scientists and applied mathematicians for past five decades. If a message needs to be received quickly and without error, merely knowing where the errors occurred may not be enough; the second condition is not satisfied as the message will be incomplete.

* Corresponding author at: College of Computer, University of Anbar, Iraq.E-mail address: **ali_makki_sagheer@yahoo.com**

Error Correcting Code

When a message is transmitted, it has the potential to get scrambled by noise. This is certainly true of voice messages, and is also true of the digital messages that are sent to and from computers. Now even sound and video are being transmitted in this manner. A digital message is a sequence of 0's and 1's which encodes a given message. More data will be added to a given binary message that will help to detect if an error has been made in the transmission of the message; adding such data is called an error-detecting code. More data may also be added to the original message so that errors made in transmission may be detected, and also to figure out what the original message was from the possibly corrupt message that was received. This type of code is an error-correcting code. The encoder transforms an n-letter word x into an m-letter word y with m > n. The decoder must be able to recover x correctly when up to r letters of y are corrupted in any way.

Several schemes exist to achieve error detection, and are generally quite simple. All error detection codes (which include all error-detection-and-correction codes) transmit more bits than were in the original data. Most codes are "systematic" — the transmitter sends the original data bits, followed by check bits — extra bits (usually referred to as redundancy in the literature) which accompany data bits for the purpose of error detection.

Repetition Code

When sending information over a noisy channel, on the highest level of abstraction we distinguish only the cases whether a symbol is transmitted correctly or not. Then the difference between the input sequence and the output sequence is measured by the Hamming distance.

Definition 1 (Hamming distance /weight) [4]:

The Hamming distance between two sequences x = (x1 . . .xn) and y = (y1 . . . yn) is the number of positions where x and y differ, i.e.,

dHamming(x, y) = |{i : 1 ≤ i ≤ n | xi ≠ yi}|.

If the alphabet contains a special symbol 0, we can also define the Hamming weight of a sequence which equals the number of nonzero positions.

In order to be able to correct errors, we use only a subset of all possible sequences. In particular, we may take a subset of all possible sequences of length n.

Definition 2 (block code) [4]:

A block code B of length n is a subset of all possible sequences of length n over an alphabet A, i.e., B ⊆ An. The rate of the code is:

$$R = \frac{\log|B|}{\log|A^n|} = \frac{\log|B|}{n\log|A|}$$

i.e., the average number of symbols encoded by a codeword.

The simplest code that can be used to detect or correct errors is the repetition code. A repetition code with rate 1/2 transmits every symbol twice. At the receiver, the two symbols are compared, and if they differ, an error is detected. Using this code over a channel with error probability p, the probability of an undetected error is p2. Sending more than two copies of each symbol, we can decrease the probability of an undetected error even more. But at the same time, the rate of the code decreases since the number of codewords remains fixed while the length of the code increases. A repetition code can not only be used to detect errors, but also to correct errors. For this, we send three copies of each symbol, i.e., we have a repetition code with rate 1/3. At the receiver, the three symbols are compared. If at most one symbol is wrong, the two error-free symbols agree and we assume that the corresponding symbol is correct. Again, increasing the number of copies sent increases the number of errors that can be corrected. For the general situation, we consider the distance between two words of the block code B.

Definition 3 (minimum distance) [4]:

The minimum distance of a block code B is the minimum number of positions in which two distinct codewords differ, i.e.

$dmin(B) = min\{dHamming(x, y):x, y \in B \mid x \neq y\}$.

The error-correcting ability of a code is related to its minimum distance.

Theorem 1: Let B be a block code with minimum Hamming distance d. Then one can either detect any error that acts on no more than d positions or correct any error that acts on no more than $\lfloor (d - 1)/2 \rfloor$ positions.

Proof: From the definition of the minimum distance of the code B it follows that at least d positions have to be changed in order to transform one codeword into another. Hence any error acting on less than d − 1 positions can be detected. If strictly less than d/2 positions are changed, there will be a unique codeword which is closest in the Hamming distance. Hence up to $\lfloor (d - 1)/2 \rfloor$ errors can be corrected [4].

Definition 4: An (m, n, d)-error-correction code is a subset $C \subseteq Zqm$ of size qn such that $d(x, y) \geq d$ for every pair of distinct elements $x, y \in C$. The parameter d is called the minimum distance of the code, and elements of C are called codewords [13].

Variations on this theme exist. Given a stream of data that is to be sent, the data is broken up into blocks of bits, and in sending, each block is sent some predetermined number of times. For example, if we want to send "1011", we may repeat this block three times each.

Suppose we send "1011 1011 1011", and this is received as "1010 1011 1011". As one group is not the same as the other two, we can determine that an error has occurred. This scheme is not very efficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g. "1010 1010 1010" in the example above will be detected as correct in this scheme). The scheme however is extremely simple, and is in fact used in some transmissions of numbers stations.

Hamming Code

In telecommunication, a Hamming code is a linear error-correcting code named after its inventor, Richard Hamming [5]. Hamming codes can detect and correct single-bit errors. In other words, the Hamming distance between the transmitted and received code-words must be zero or one for reliable communication. Alternatively, it can detect (but not correct) up to two simultaneous bit errors.

In contrast, the simple parity code cannot correct errors, nor can it be used to detect more than one error (such as where two bits are transposed).

In mathematical terms, Hamming codes are a class of binary linear codes. For each integer $m > 1$ there is a code with parameters: $[2m - 1, 2m - m - 1, 3]$. The parity-check matrix of a Hamming code is constructed by listing all columns of length m that are pair-wise independent.

We now give a simple example of an error-correction code: a Hamming or repetition code. In this example, redundancy is introduced directly into a message by repeating each bit (or number in Zq) three times.

For example, consider messages which are 3-bit strings, so n = 3. Each bit in the string is repeated three times, so the resulting message length is m = 9.

Message	Codeword
000	000000000
001	000000111
010	000111000
011	000111111
100	111000000
101	111000111
110	111111000
111	111111111

Note that the minimum distance between the messages may be 1, but the minimum distance of the repetition code is 3. This means that any 1-bit error in the codewords may be corrected. Indeed, if we look at the three blocks of three bits each in a received message, we can recover the original bit by taking the most common bit among the three. If no error has occurred, the three bits would be 000 or 111, and if a single bit flip has occurred, the bits would be 100, 010, or 001 in case a zero was encoded, and 011, 101, or 110 if a one was encoded. In either case, the most common bit gives us the correct answer [13].

Error Correcting Codes only succeed if the errors made in the individual bit positions are relatively uncorrelated, so that the number of simultaneous errors in many bit positions is small. If there are many simultaneous errors, the error-correcting code will not be able to correct them (Peterson & Weldon, 1972).

Sequenceable Group and Communication:

Anon – trivial finite group G of order n is said to be sequenceable if its elements can be arranged in a sequence

(b1, b2 ………, bn) in such a way that the partial products (a1, a2, ……., an) where ai = b1b2 ……… bi are distinct.

The sequence (b1, b2, …… , bn) is called a sequencing for G.

If (b1, b2, …… , bn) is a sequencing for G then b1= e where e is the identity of G.

A Latin square of order n is an n × n array defined on a set X with n elements such that every element of X appears once in each row and once in each column.

A Latin is said to be based on a group G if the Latin square can be bordered with the elements of G to form the clayey table of G.

An n × n Latin is said to be row complete if every pair {x, y} of distinct elements of X occurs exactly once in each order in adjacent vertical cells. If a Latin square is both row complete and column complete then it is said to be complete [15].

Theorem 2: Let G be a sequenceable group and (b1, b2, …… , bn) be a sequencing with a associated directed product (a1, a2, ……., an). then L=(Lij) where Lij = a i -1 aj for 1 ≤ i, j ≤ n. is a complete Latin square.

Proof: Suppose Lij = Lik for some 1≤ i, j ≤ n. then a i -1 aj = a i -1 ak giving aj=ak..

Therefore j = k and L has no repeated entries in any row. Similarly, L has no repeated entries in any column therefore L is a Latin square. To show that:

L is row complete we need ai -1aj=X and ai -1aj+1 = Y to have a unique solution for i and j given any ordered pair (x, y) of distinct elements of G [15].

Inverting both sides of the first equation and post – multiplying by the second gives ai -1aj = x -1y.

That is bj+1 = x-1 y, uniquely determining j.

Now ai -1aj = x uniquely determines i and L is row complete. By same way we can show that L is also column complete. Therefore L is a complete Latin square.

Classifying Sequenceable Group

In this section we introduced completely classified sequenceable groups.

Abelian groups

Tthe following theorem exactly which abelian groups are sequenceable. A finite abelian group G is sequenceable if and only if G is a binary group. The binary group is defined to be a group with a single element of order 2.

3-1-2 Dihedral groups

Let $n \geq 3$ we describe the dihedral group D2n, as the set of ordered pairs (x, Є) with $x \in$ Zn and Є $\in$ Z2

Defined by $(x, 0)\,(y, \delta) = (x+y, \delta)$.

$(x, 1)\,(y, \delta) = (x - y, 1 + \delta)$.

In 1976 Anderson [1] showed that D2p is a sequenceable if p is a prime with a primitive root r such that $3r \equiv -1$ (mod p). also in 1976 Friedlander [14] showed that D2p is sequenceable if p is prime and $p \equiv 1$ (mod 4) and where p is a prime such that

$p \equiv 7$ (mod 8) and p has a primitive root r such that $2r \equiv -1$ (mod p) and by [10] the dihedral groups D2n of order 2n. are sequenceable for all n. where $n \neq 3$ (D6 is not sequenceable) and $n \neq 4k$ and the dihedral groups D2n are sequenceable when $n = 4k$, except when $n = 4$.

Therefore, the following groups are known to be sequenceable.

Some groups of order pq where p and q are odd prime, direct product, of some of the groups of the previous type if both p and q are congruent to 3 modulo 4, at least one of the non – abelian groups of order pm, for p an odd prime and $m \geq 3$, non – abelian groups of order n, where $10 \leq n \leq 32$. and A5, S5.

Orthogonality:

Definition 5: Tow Latin squares A = (aij) and B = (bij) are orthogonal if the set

$\{(aij, bij): 1 \leq i, j \leq n\}$ contains all possible paris.

Example:- The following tow Latin squares are orthogonal

1	2	3	4		1	4	3	2
2	1	4	3		3	2	1	4
3	4	1	2		2	3	4	1
4	3	2	1		4	1	2	3

Theorem 3: If A1, A2, …., Am are mutually orthogonal Latin squares of order n then

$m \leq n-1$

Proof: Let Ak= (aij(k))nxn. By (if A and B are orthogonal Latin squares. Then the standard form of A and B is A* and B* respectively, are orthogonal) [18].

We may assume that all A1,…., Am are in standard form, otherwise we standardizes them, without affecting orthogonally. i.e. ajk(k)=1.

Consider the set S= { (i, j, k): aij (k)= 1}.

Clearly the number of elements of S is equal to the total number of 1's in A1, ….., Am, so that

$|S| = n\,m$ ……………… (1)

Consider a triple $(i, j, k) \in S$, each of the squares has 1 in the position (1, 1).

Hence, if $i = j = 1$ then k can be arbitrary. Also, no other entry in the position (1, j) or

(i,1) can be 1 so that we can not have $i = 1 \neq j$ or $i \neq 1 = j$, finally, if $i \neq 1$ and $j \neq 1$, then because of orthogonally, there may exit at most one k such that $(i, j, k) \in S$.

We conclude that

$|S| \leq m + (n-1)\,2$. ……………. (2)

Combining (1) and (2) we obtain m ≤ n -1.

Latin squares from finite fields:

In this section we introduce a method of constructing orthogonal Latin squares from finite fields.

Theorem 4: If n = pt. where p is a prime and t ≥ 1, then there exist n-1 mutually orthogonal Latin squares of order n [19].

Example:- Let us use the finite field Z5 to construct 4 mutually orthogonal Latin squares of order 5.

First, we Let f_1= 1, f_2 = 2, f_3 = 3, f_4 = 4, f_5 = 0.

The first Latin squares A1 = (aij(1))5×5 is given by aij(1)= fi + fj.

j		1	2	3	4	5
f_j		1	2	3	4	5
i	f_i					
1	1	2	3	4	0	1
2	2	3	4	0	1	2
3	3	4	0	1	2	3
4	4	0	1	2	3	4
5	0	1	2	3	4	0

Similarly, the second Latin square $A_2 = (a_{ij}^{(2)})$ is given by $a_{ij}^{(2)} = 2f_i + f_j$

j			1	2	3	4	5
f_j			1	2	3	4	5
i	f_i	$2f_i$					
1	1	2	3	4	0	1	2
2	2	4	0	1	2	3	4
3	3	1	2	3	4	0	1
4	4	3	4	0	1	2	3
5	0	0	1	2	3	4	0

Repeating similar calculation for A^3 and A^4 we obtain the squares:

$$A_1 = \begin{array}{ccccc} 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4\ 0 & 1 & 2 & 3 & \\ 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \end{array}$$

$$A_2 = \begin{array}{ccccc} 3 & 4 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 & 4 \\ 2\ 3 & 4 & 0 & 1 & \\ 4\ 0 & 1 & 2 & 3 & \\ 1 & 2 & 3 & 4 & 0 \end{array}$$

$$A_3 = \begin{array}{ccccc} 4 & 0 & 1 & 2 & 3 \\ 2 & 3 & 4 & 0 & 1 \\ 0\ 1 & 2 & 3 & 4 & \\ 3 & 4 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 & 0 \end{array}$$

$$A_4 = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 1 & 2 & 3 \\ 3\ 4 & 0 & 1 & 2 & \\ 2 & 3 & 4 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 \end{array}$$

Definition 5: Let $A = (a_{ij})_{m \times m}$ and $B = (b_{ij})_{n \times n}$ two Latin square. Their direct product $C = A \times B$ is an $mn \times mn$ array, in dexed by the elements of {1, ……,m} × {1,……..,n} and entries $C_{(I,j),\ (k,l)} = (a_{ik}, b_{jl})$ [19].

Example:- consider the following tow Latin square

$$\begin{array}{cc} 1 & 2 \\ 2 & 1 \end{array} \quad , \quad \begin{array}{ccc} 2 & 3 & 1 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{array}$$

Their direct product, according to definition, is

	(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)
(1,1)	(1,2),(1,3),(1,1),(2,2),(2,3),(2,1)
(1,2)	(1,3),(1,1),(1,2),(2,3),(2,1),(2,2)
(1,3)	(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)
(2,1)	(2,2),(2,3),(2,1),(1,2),(1,3),(1,1)
(2,2)	(2,3),(2,1),(2,2),(1,3),(1,1),(1,2)
(2,3)	(2,1),(2,2),(2,3),(1,1),(1,2),(1,3)

After renumbering this becomes

2	**3**	**1**	**5**	**6**	**4**
3	**1**	**2**	**6**	**4**	**5**
1	**2**	**3**	**4**	**5**	**6**
5	**6**	**4**	**2**	**3**	**1**
6	**4**	**5**	**3**	**1**	**2**
4	**5**	**6**	**1**	**2**	**3**

Theorem 5: If A and B are orthogonal Latin squares of order m, and if C and D are orthogonal Latin squares of order n. then A × C and B × D are orthogonal Latin square [19].

Corollary : If n ≠ 2 (mod 4) then there exists a pair of orthogonal Latin squares of order n.

Proof: Let n p1a1 p2a2…. Pkak be the decomposition of n into a product of primes, with p1 < …< pk, since n ≠ 2 (mod 4) it follows that p1a1 > 2, and so piai >2 foe every i, by theorem (4), for each i (1 ≤ i ≤ k) there exist a pair Ai, Bi of orthogonal Latin squares of order pia1, but then the Latin squares A = A1 × …× Ak and B = B1 × …× Bk are orthogonal by theorem (5) and have order n [19].

The Proposed Latin Square Error Correcting Code (LSECC)

This section obtain who we can exploit the characteristics of the orthogonal Latin Squares mentioned to design a new technique of the Error Correcting Code we call it: Latin Square Error Correcting Code (LSECC). The proposed new technique is an Error Correcting code method that is used to save the information from the lost may be occur in the transmission media. The new technique is uses the characteristics of the Orthogonal Latin Squares and employ it to correct most of the simultaneous errors in bits caused by noise.

Definition 6: A code is said to be t-error correcting if when no more than t-error has occurred in the transmissions of codeword.

We note that if we have n × n Latin Square (ai j), we can build n2 codewords, by using ordered triples (i, j, ai j).

These triples are of Hamming distances of at least 2 a part because of constructions Latin square.

Example:- Let the Latin Square of group Z3, the codewords are:

The Latin Square:

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{pmatrix}$$

The code words:

(0, 0, 0), (0, 1, 1), (0, 2, 2),
(1, 0, 1), (1, 1, 2), (1, 2, 0),
(2, 0, 2), (2, 1, 0), (2, 2, 1),

A single error detecting code formed from Z3 and its corresponding code words.

Theorem 6: Any pair of orthogonal Latin Square of order n yields a 1-error correcting code with n2 code words.

Proof: Let the n2 code words of length 4 over the alphabet {0, 1, …., n-1} the code words are merely the 4-tuples code words of the form (i, j, aij, bi j) 0 ≤ i, j ≤ .n-1.

Such that [ai j] = A and [bi j] = B forming two Latin Squares.

Suppose that w = (i, j, ai j, bi j) and w` = (i`, j`, ai j`, bi j`) are two such words.

If i = i` and j = j` clearly the two words are the same, if ai j=ai` j` and bi j=bi` j` they must be the same words A and B are orthogonal. If i = i` and ai j=ai` j` then the words are same, since, A is Latin Square [14]. The other cases are all similar.

Thus any two codewords of distances 3 which will be corrected one error.

Now, from this theorem we can use sets of orthogonal Latin Squares to construct codes.

If we have q × q Latin Squares L1, L2, …., Ln, we construct codewords by taking a coordinate pair and adjoining the corresponding element from each Latin Squares

(i, j, L1, L2, …., Ln).

These q2 codewords have hamming distance of at least 2t + 1 from each other.

We can show that any pair of orthogonal Latin Squares of order n yields a 1-error correcting code with n2 code words of length 4 over the alphabet {0, 1, …., n-1}. Thus any two code words at distance 2 or less are the same and have a code of distance 3 which will correct one error.

Example:- Let the following cayley table of Z4 and one of its orthogonal mates is:

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 3 & 2 & 1 \\ 2 & 1 & 0 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \end{pmatrix}$$

(0, 0, 0, 0), (0, 1, 1, 3), (0, 2, 2, 2), (0, 3, 3, 1),
(1, 0, 1, 2), (1, 1, 0, 1), (1, 2, 3, 0), (1, 3, 2, 3),
(2, 0, 2, 1), (2, 1, 3, 2), (2, 2, 0, 3), (2, 3, 1, 0),
(3, 0, 3, 3), (3, 1, 2, 0), (3, 2, 1, 1), (3, 3, 0, 2)

The codewords generated from the above Orthogonal Latin Squares are.

When the sender want to transmit the following bits:

10 11 01 10 00 01 11 01

The sender do the following for each four bits:

Takes the four bits to make it pair of two bits numbers (i, j).

Takes the codeword correspond to i and j from possible code words as (i, j, ai j, bi j).

Send the codeword (i, j, ai j, bi j).

He send the following code words as obtained below:

10b = 2d

11b = 3d

Then he send the codeword: (2, 3, 1, 0) ≡ 10 11 01 00

01b = 1d

10b = 2d

Then he send the codeword: (1, 2, 3, 0) ≡ 01 10 11 00

00b = 0d

01b = 1d

Then e send the codeword: (0, 1, 1, 3) ≡ 00 01 01 11

11b =3d

01b =1d

Then he send the codeword: (3, 1, 2, 0) ≡ 11 01 10 00

And so on for other bits in the transmission media,

Therefore, the data: 1011 0110 0001 1101

is encoded into: 1011 0100 0110 1100 0001 0111 1101 1000

and transmitted.

Suppose the transmitted bits affect by noise cause the following errors:

1001 0100 0101 1100 1101 0111 1101 0000

The receiver takes each eight bits to convert it into corresponding codeword and match with its possible code words and do the following for each eight bits:

1. If the received codeword match with one of the possible code words there is no error. He takes the first two symbols of the codeword as four bits
2. If the received codeword no match with one of the possible code words there is an error, search the code words to find almost match three symbols of the

codeword and correct it. He takes the first two symbols of the corrected codeword as four bits

3. Otherwise there is damage in the transmission and send an acknowledgement to the sender to retransmit the data.

Take the first eight bits (codeword): 1001 0100 has a single error; the third bit is changed from 1 to 0.

Where the error codeword is 1001 0100 ≡ (2, 1, 1, 0).

Therefore, there is no more other the single codeword (2, 3, 1, 0) of the possible codeword match three elements of the error codeword. Then he receive 1011.

Take the second eight bits (codeword): 0101 1100 have two simultaneous errors; the third bit is changed from 1 to 0 and the fourth bit is changed from 0 to 1,

Where the error codeword is 0101 1100 ≡ (1, 1, 3, 0).

Therefore, there is no more other single codeword (1, 2, 3, 0) of the possible codeword match three elements of the error codeword. Then he receive 0110.

Take the third eight bits (codeword): 1101 0111 have two simultaneous errors; the first bit is changed from 0 to 1 and the second bit is changed from 0 to 1.

Where the error codeword is 1101 0111 ≡ (3, 1, 1, 3).

Therefore, there is no more other single codeword (0, 1, 1, 3), of the possible codeword match three elements of the error codeword. Then he receive 0001.

Take the fourth eight bits (codeword): 1101 0000 has single bit errors; the fifth bit is changed from 1 to 0.

Where the error codeword is 1101 0000 ≡ (3, 1, 0, 0).

Therefore, there is no more other single codeword (3, 1, 2, 0), of the possible codeword match three elements of the error codeword Then he receive 1101.

Finally, he receives the data 1011 0110 0001 1101.

Latin Square Error Correcting Code Algorithm

The previous example explain the idea of LSECC, it is correct even most the two simultaneous bits errors, if the sender uses two orthogonal Latin Squares 8 × 8 (i.e. cayley table of Z8), the three simultaneous bits errors may be corrected, therefore, the using of the cayley table of Z2n may be correct the n-simultaneous bits errors. We can construct the following LSECC Algorithm:

Algorithm: (LSECC)

1- Initialization

1.1- Choose m = 2n, where m represent the dimension of the Latin Square.

1.2- Build two orthogonal Latin Squares A and B of dimension m×m (i.e. cayley table of Zm).

1.3- Construct all possible 4-tuples codewords of the form (i, j, aij, bi j).

2- Coding and Sending

The sender separates the data into 2n-bits words, and then does the following for each 2n-bits:

2.1- Takes the 2n-bits to make it pair of n-bits numbers (i, j).

2.2- Takes the codeword correspond to i and j from possible codewords as (i, j, ai j, bi j).

2.3- Send the codeword (i, j, ai j, bi j).

3- Decoding and Receiving

The receiver takes each 4n-bits to convert it into corresponding codeword and match with its possible code words and do the following for each 4n-bits:

3.1- If the received codeword match with one of the possible code words there is no error. He takes the first two symbols of the codeword as two n-bits received data.

3.2- If the received codeword do not match with one of the possible code words there is an error, search the code words to find almost match three symbols of the codeword and correct it. He takes the first two symbols of the corrected codeword as two n-bits received data.

3.3- Otherwise there is damage in the transmission and send an acknowledgement to the sender to retransmit the data.

Conclusion

Error-correcting code theory is essential to our modern life. The rapid growth of the amount of information needed to be transmitted makes it very important to continue our study of this subject. Codes that are more efficient to transmit, correct more errors, and are more efficient to decode are always needed. The proposed LSECC is a good algorithm and more efficient than some previous ECC techniques, which is correct all 1-error and the most of the burst errors n-error. The main advantages of LSECC are the n-error correcting code, the second, it is the redundancy code have length equal the length of the data we want to transmit, i.e. no more than the length of the original data such as the previous techniques. The advantages of the non-complete burst error correcting code presented in this paper are the very efficient and simple decoding algorithm, the low redundancy, and the fact that it is systematic.

Finally, the using of the cayley table of Z2n may be correct the n-simultaneous bits errors. The more efficient way to Error Correcting codes would be very helpful. With increase demands for information transfer, in addition to new uses for the subject in other areas, the importance of research in error-correcting code theory will only increase as time goes on.

References

[1] B. A. Anderson, Sequencings and Starters, Pacific J. Math . 64 (1976) 17-24.

[2] E. R. Berlekamp, Algebraic Coding Theory, MC Graw – Hill

[3] N. Cal and R. W. Yeung, Network Error Correction, Part II: Lower Bounds, Communications In Information And Systems, Vol. 6, No. 1, pp. 37-54, 2006, International Press, 003.

[4] M. Grassl, Classical Information Theory and Classical Error Correction, Lectures on Quantum Information, WILEY-VCH Verlag GmbH & Co. KGaA,Weinheim ISBN: 978-3-527-40527-5, 2007.

[5] R. W. Hamming. Error-detecting and error-correcting codes. Bell System Technical Journal, 29(2):147–160, 1950.

[6] G. Held, Data communication networking devices: operations, utilization and LAN and WAN internetworking, forth edition, John Wiley & Sons Ltd, 2001.

[7] J. Higham , A Product Theorem For Row – Complete Latin Squares, J. Combi. Des 5 311-318, 1997.

[8] M. Y. Hsiao, D. C. Bossen and R. T. Chien, Orthogonal Latin Square Codes, 1969.

[9] D. R. Irvin, Embedded a Secondary Communication Channel Transparently within a Cyclic Redundancy Check (CRC), 2001.

[10] J. Isbell, Sequencing Certain Dihedral Group, Discrete Math. 85(1990) 323-328.

[11] P. Li, Sequencing the Dihedral Groups D4K, Discrete Math. 157 271-276, 1997.

[12] A. D. Keed Well, Complete Mapping and Sequencing of Finite Group, The CRC Handbook of combinatorial Designs (Eds. C.J.Colbourn and J.H. Dinitz) CRC, Press 246-253, 1996.

[13] A. Nayak, Error-Correction Codes, UW Math 135, July 29, 2005.

[14] R. Friedlander, Sequences in Group with Distinct Partial Products , Aequations math. 14 (1976) 59-66.

[15] M. A. Ollis, Sequenceable Groups and Related Topics, 2002

[16] V. Pless, Introduction to the Theory of Error–Correcting Code, Wiley, New York, 1982.

[17] D. Raphaeli, A Simple and Efficient Burst Error Correcting Code Based on an Array Code, Senior Member IEEE

[18] J. Rotman, An Introduction To The Theory Of Groups, 4th Edition, Springer-Verlag, 1994.

[19] N. Ruskus, Finite Mathematics, March 6, 2002.

[20] R. W. Yeung and N. Cal, Network Error Correction, Part I: Basic Concepts And Upper Bounds, Communications In Information And Systems, Vol. 6, No. 1, pp. 19-36, 2006, International Press, 002.

تصحيح خطأ الرموز باستخدام المربع اللاتيني

علي مكي صغير مكارم عبد الواحد عبد الجبار

Email: ali_makki_sagheer@yahoo.com

الخلاصة

البيانات الرقمية التي تخزن في الحاسبات أو التي ترسل عبر شبكات الحاسوب بالتاكيد خاضعة للخطأ بسبب الوسط الفيزيائي المستخدم في الخزن او الارسال. ان رموز تصحيح الخطأ (ECC) هي وسائل تستخدم فضلة من البيانات حتى إذا تعرض جزء منها للخطأ أو الفقدان، فيمكنها ان تسترجع البيانات الأصلية. رموز تصحيح الخطأ (ECC) تستخدم في التقنية الحديثة لكي نحمي المعلومات من الأخطاء. رموز تصحيح الخطأ المتتابع مطلوبة في التطبيقات المختلفة. مثل هذه الرموز تدعى رموز تصحيح الخطأ المتتابع الكامل. هناك عدد محدود نسبياً من رموز تصحيح الخطأ المتتابع الكامل.

يقدم هذا البحث طريقة مقترحة من رموز تصحيح الخطأ تستند على مفهوم ونظرية المربعات اللاتينية، حيث يستخدم خصائص المربعان اللاتينية المتعامدة لتصحيح الأخطاء. ان الطريقة المطورة هي ليست من طرق رموز تصحيح الخطأ المتتابع الكامل، لكن يممكن أن تصحح أكثر تتابع من الاخطاء بطول $n \geq i$ ، لكن ليس الكل. على أية حال، إذا كانت الاخطاء صغيرة بما فيه الكفاية، فان هذه الطريقة ممكن أن تستخدم كطريقة من طرق رموز تصحيح الخطأ المتتابع.