Radical Data Compression Algorithm Using Factorization

Zirra Peter Buba **zirrapeter@yahoo.com**
Department of Mathematical Sciences **zirrapeter@gmail.com**
Adamawa State University
Mubi, 650221, Nigeria.

Gregory Maksha Wajiga ***gwajiga@gmail.com***
Department of Mathematics and Computer Science
Federal University of Technology
Yola, 640284, Nigeria.

Abstract

This work deals with encoding algorithm that conveys a message that generates a "compressed" form with fewer characters only understood through decoding the encoded data which reconstructs the original message. The proposed factorization techniques in conjunction with lossless method were adopted in compression and decompression of data for exploiting the size of memory, thereby decreasing the cost of the communications. The proposed algorithms shade the data from the eyes of the cryptanalysts during the data storage or transmission.

Keywords: Data Compression, Cryptography, Lossless, Algorithm

1. INTRODUCTION

In this paper the generic term message is used for the object to be compressed. Compression consists of two components, an encoding algorithm that takes a message and generates a "compressed" representation (with fewer characters), and a decoding algorithm that reconstructs the original message from the compressed representation.

There are "lossless" and "lossy" forms of data compression. Lossless (reversible) data compression is used when the data has to be uncompressed exactly as it was before compression. Text files are stored using lossless techniques, since losing a single character can in the worst case make the text dangerously misleading. Archival storage of master sources for images, video, and audio data generally needs to be lossless as well. However, there are strict limits to the amount of compression that can be obtained with lossless compression.

Lossy (irreversible) compression, in contrast, works on the assumption that the data doesn't have to be stored perfectly. Much information can be simply thrown away from images, video, and audio data, and when uncompressed, such data will still be of

acceptable quality. Compression ratios can be an order of magnitude greater than those available from lossless methods [2].

The proposed factorization techniques of data compression uses lossless method of compressing and decompressing of data to exploit the size of memory, thereby decreasing the cost of the communications and shading the data from the eyes of the cryptanalysts during the data storage or transmission.

2. DATA COMPRESSION

Data compression is defined as the reduction of the volume of a data file without loss of information [1]. As pointed out by [5] and [3], data compression aims to condense the data in order to reduce the size of a data file.

Data compression has important application in the areas of data transmission and data storage [4]. Many data processing applications require storage of large volumes of data, and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the proliferation of computer communication networks is resulting in massive transfer of data over communication links. Compressing data to be stored or transmitted reduces storage and communication costs. When the amount of data to be transmitted is reduced, the effect is that of increasing the capacity of the communication channel. Similarly, compressing a file to half of its original size is equivalent to doubling the capacity of the storage medium. It may then become feasible to store the data at a higher, thus faster, level of the storage hierarchy and reduce the load on the input/output channels of the computer system.

3. THEORETICAL FRAME WORK

The main feature of the radical data compression algorithm takes as an input a plaintext, compress the string of the characters and produce an output in ciphertext. To decompress, the ciphertext requires a decompression key. The decompression key is transmitted to the receiver through different media such as email. The whole process is depicted in Figure 1.

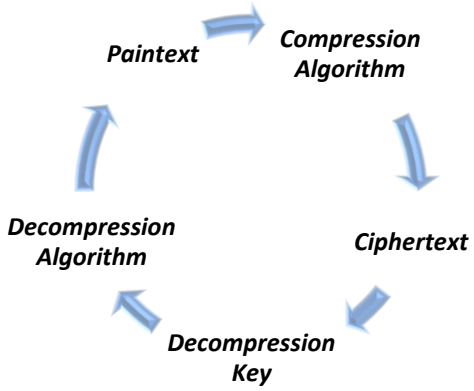

FIGURE 1: Compression and Decompression Process

4. THE PROPOSED RADICAL COMPRESSION ALGORITHMS

4.1 Compression Algorithm

The compression of the data is achieved through the following:

i. Each character or symbol in the message such as number, white space and punctuation are assigned a numerical value.

ii. The encoding scheme in Table 1 is employed.

4.1.1 Compression Key

a. Serialization of string.
b. Write down character of the input string without repetition.
c. The encoding scheme in Table 1.

TABLE 1: Encoding Scheme

Symbol	*Count*	*Locations*	*Product*	*Sum*	*Maximum Location*
S_j	C_j	ℓ_{jk}	$\alpha_j = \prod_{k=1}^{C_j} \ell_{jk}$	$\beta_j = \sum_{k=1}^{C_j} \ell_{jk}$	$h_j = max(\ell_{jk})$

Example: If we are going to compress the random message "*WHO IS PROMISSING WHO*", we would have an encoded message that is given in *Table 3* using *Table 2*.

TABLE 2: Serialization of plaintext

S/No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Symbol	W	H	O		I	S		P	R	O	M	I	S	I	N	G		W	H	O

4.2 Decompression Algorithm

The algorithm, which describes each step in compression process, is listed in Algorithm 1.

Algorithm_Compression (input: string)

Step 0: compute the product (α_j), sum (β_j) and the maximum occurrence of the character (h_j)

begin

Step 1: compute C_j factor(s) of $\alpha_j \leq h_j$ such that the factor(s) satisfy the value α_j and β_j in Table 3 or

Step 2: if $C_j = 1$, output h_j
endif

Step 3: if (i) is true, then it contains all the location(s) (ℓ_{jk}) of S_j

Step 4: write the character or symbols S_j in its appropriate locations as determined by (iii)

Step 5: repeat steps (1) to (4) for C_j of S_j until $S_j = 0$

end

Algorithm 1. Compression Algorithm Process

TABLE 3: Encoded Symbols

Symbol	Count	Locations	Product	Sum	Maximum Location
S_j	C_j	ℓ_{jk}	$\alpha_j = \prod_{k=1}^{C_j} \ell_{jk}$	$\beta_j = \sum_{k=1}^{C_j} \ell_{jk}$	$h_j = \max(\ell_{jk})$
White space	3	4, 7, 18	504	29	18
G	1	17	17	17	17
H	2	2, 20	40	22	20
I	3	5, 12, 15	900	32	15
M	1	11	11	11	11
N	1	16	16	16	16
O	3	3, 10, 21	630	34	21
P	1	8	8	8	8
R	1	9	9	9	9
S	3	6, 13, 14	1092	33	14
W	2	1, 19	19	20	19

4.2.1 Decompression key

f_j factors of $\alpha_j \leq h_j$ such that the factor(s) satisfy the values of α_j and β_j in Table 3 or if $f_j = 1$, write h_j.

This key is transmitted through a different channel (email) to the recipient in order to be able to decompress the characters.

Example: we can easily find where characters ‘white space’, ‘G’,’H’, ‘I’, ‘M’, ‘N’, and so on occurred in the message by exploring the compression algorithm in section 4.2 visa vice Table 3.

To decode the character symbol ‘white space’:
From Algorithm 1, the computed value of $C_j = 2$, $\alpha_j = 504$, $\beta_j = 29$, and $h_j = 18$, as depicted in Table 3.

a) We compute the factors of $\alpha_j \leq h_j$, which are (1, 2, 3, 4, 6, 7, 8, 9, 12, 14 and 18)
b) From (a) we take C_j factors of $\alpha_j \leq h_j$ that satisfied the value of α_j and h_j. These are 4, 7 and 18

Since the value of α_j and h_j are satisfied, a blank space is created at locations 4, 7 and 18.

Taking the second character symbol G: the value of $C_j = 1$, $\alpha_j = 17$, $\beta_j = 17$, and $h_j = 17$,

a) We compute the factors of $\alpha_j \leq h_j$, which are (1 and 17)

b) From (a) we take C_j factors of $\alpha_j \leq h_j$ that satisfied the value of α_j and h_j and that is $h_j = 17$

Since the value of α_j and h_j are satisfied, a character G is written at locations 17 only.

Similarly, to decode the character symbol 'H', we take the value of $C_j = 2$, $\alpha_j = 40$, $\beta_j = 22$, and $h_j = 20$,

a) We compute the factors of $\alpha_j \leq h_j$, which are (1, 2, 4, 5, 8, 10 and 20)
b) From (a) we take C_j factors of $\alpha_j \leq h_j$ that satisfied the value of α_j and h_j. These are 2 and 20

Since the value of α_j and h_j are satisfied, the character H is printed at locations 2 and 20.

This is continued until all the characters of the plaintext have been recovered to its original plaintext 'WHO IS PROMISSING WHO' as depicted in Table 2.

5 CONCLUSION

The new data compression transforms a string of characters into a new string (encoded) using compression key and reverses the encoded string using decompression key. The encoded string contains the same information as the original string but whose length is small as possible. These shade the information contents from the eyes of the intruders, but thereby increasing the capacity of the communication channel. Therefore, the algorithms will be found useful by organizations that deal with sensitive documents.

6. REFERENCES

[1] Fraser, A.G. (2011) Data Compression and Automatic Programming. Available at: comjnl.oxfordjournals.org [Accessed 16 October 2010]

[2] Guy, E. B. (2010) Introduction to Data Compression. Available at: www. eecs.harvard. edu/~michaelm /CS222/ compression.pdf, [Accessed 24 January 2011]

[3] Kattan, A. "Universal lossless compression technique with built in encryption". M. Sc. Thesis, University of Essex, UK., 2006

[4] Khalid, S "Introduction to Data Compression (2nd ed.)". New York : Morgan Kaufmann Publishers Inc., pp 151-218 (2000).

[5] Ziviani, N., Moura, E., Navarro, G., and Baeza-Yates, R. "Compression: A key for next generation text retrieval systems". *IEEE Computer Society*, 33(11), 37- 44. 2000