Radical Data Compression Algorithm Using Factorization

Zirra Peter Buba *zirrapeter@yahoo.com*
Department of Mathematical Sciences
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 [1]. 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 [3]. As pointed out by [4] and [5], 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 [6]. 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 [7]. 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 [8]. The decompression key is transmitted to the receiver through different media such as email or Short Message Service (SMS) to strengthen its security. The whole process is depicted in Figure 1.

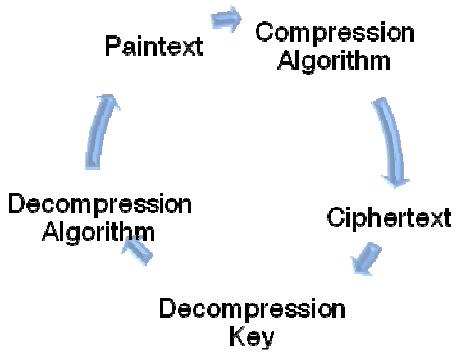

FIGURE 1: Compression and Decompression Process

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.

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

TABLE 1: Encoding Scheme

Illustrative example 1: 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*.

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

TABLE 2: Serialization of plaintext

Symbol	S_j	White space	W	H	O	I	S	P	R	M	N	G
Location	ℓ_{jk}	4,7,17	1,18	2,19	3,10,20	5,12,14	6,13	8	9	11	15	16
Character Count	C_j	3	2	2	3	3	2	1	1	1	1	1
Sum	$\beta_j = \sum_{k=1}^{C_j} \ell_{jk}$	28	19	21	33	31	19	8	9	11	15	16
Product	$\alpha_j = \prod_{k=1}^{C_j} \ell_{jk}$	476	18	38	600	840	78	8	9	11	15	16
Maximum Location	$h_j = \max(\ell_{jk})$	17	18	19	20	14	13	8	9	11	15	16

TABLE 3: Encoded Symbols

Hence the compressed data is *WHOISPRMNG* and a white space.

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.

4.2 Decompression Algorithm

The algorithm, which describes each step in compression process, is listed in the Algorithm_Decompression (input: symbol) below

Algorithm_Decompression (input: symbol)

Step 1: Read the product (α_j), sum (β_j) and the maximum occurrence of the character (h_j) as produced in Table 3.

Step 2: Begin
Step 3: If $C_j = 1$, then output h_j
Step 4: Else
Step 5: Compute C_j factor(s) of $\alpha_j \leq h_j$ such that the factor(s) satisfy the value α_j and β_j
Step 6: Endif
Step 7: If (3) or (5) is true, then output the character S_j at location(s) (ℓ_{ijk})
Step 8: Else goto step 5
Step 9: Repeat steps (1) to (8) for C_j of S_j until $S_j = 0$
Step 10: EndBegin
EndAlgorith_Decompression

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.

Illustrative example 2: To decode the character symbol 'white space':

From the Algorithm_Decompression (input: symbol), the computed value of $C_j = 3$, $\alpha_j = 476$, $\beta_j = 28$, and $h_j = 17$, as depicted in Table 3.

a) We compute the factors of $\alpha_j \leq h_j$, which are 1, 2, 4, 7, 14 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. These are 4, 7 and 17.

Since the value of α_j and h_j are satisfied, a blank space is created at locations 4, 7 and 17.

Taking the second character symbol G: the value $C_j = 1$, $\alpha_j = 16$, $\beta_j = 16$, and $h_j = 1$

c) We compute the factors of $\alpha_j \leq h_j$, which are 1 and 16.
d) From (c) we take C_j factors of $\alpha_j \leq h_j$ that satisfied the value of α_j and h_j and that is $h_j = 16$

Since the value of α_j and h_j are satisfied, a character G is written at locations 16 only.

Similarly, to decode the character symbol "H", we take the value of $C_j = 2$, $\alpha_j = 38$, $\beta_j = 21$, and $h_j = 19$, from the Algorithm_Decompression (input: symbol).

e) We compute the factors of $\alpha_j \leq h_j$, which are 1, 2 and 19
f) From (e) 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 19

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

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

5. RESULTS AND DISCUSSION

From Table 3, and the results of illustrative example 2, showed how efficient and practicable [9] the proposed data compression scheme in Table 1 and decompression algorithms in section 4.2 in handling any kinds of plaintext in an unsecure channel and how it increases storage capacity for faster data processing [3].

The meaningless results in Table 3 is in line with the proposed data compression and decompression algorithms which helped to conceal the content of sensitive information from the terrorist [9,10]. These meant that the proposed algorithms has proved to prevent the hacker from gaining insight of what the content of information was all about while in either transit or store.

The result of the study has also successfully reduced the large volume of the plaintext that contained the same information as the original string in Table 2 but whose length was small as possible [3,4,7]. These proved that it may then become feasible to store the data at a higher, faster, level of the storage hierarchy [5] and reduce the load on the input/output channels of the computer systems which is the main goal of data compression techniques [11,12].

The results of the study also shown that one key is used for encoding the data and another different key were used for decoding of the data. The results revealed that the receiver obtained these key from the sender secretly through either SMS or Fax machine. The strength of cryptography systems in the computer era rests on the secrecy of the key, not on the algorithm [8]. It is clear that our information is not easily visible to an unauthorized person without a decoding key. The concealment of the key rather than the encryption technique is one of the most important features of a strong cryptography system [10].

6. 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.

7. REFERENCES

[1] W.S. Steven.*The Scientist and Engineer's Guide To Digital Signal Processing*. Califonia: Technical Publishing, 2007.

[2] E. B. Guy. "Introduction to Data Compression". Internet: www. eecs.harvard. edu/~michaelm /CS222/ compression.pdf, 2010 [Jan. 24, 2011].

[3] A.G. Fraser. "Data Compression and Automatic Programming". Internet: www.comjnl.oxfordjournals.org, 2010 [Oct. 16, 2010].

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

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

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

[7] E.J.D. Garba and S.E Adewumi. "A Cryptosystems Algorithm Using Systems of Nonlinear Equations". *Iranian Journal of Information Science And Technology,* 2003, 1*(1), pp.* 43-55.

[8] M. Milenkovic. *Operating System: Concepts and Design*, New York: McGrew-Hill, Inc., 1992.

[9] V. Singla, R. Singla, and S. Gupta. “Data Compression Modelling: Huffman and Arithmetic” *International Journal of The Computer, the Internet and Management, 2008, 16(3), pp 64- 68*

[10] *Cryptography FAQ (03/10: Basic Cryptology), "3.5. What are some properties satisfied by every strong cryptosystem?" [cited 23 August 2006]; Available:* *http://www.faqs.org/faqs/cryptographyfaq/part03/index.html*.

[11] A. Hauter, M.V.C., R. Ramanathan. *Compression and Encryption. CSI 801 Project Fall 1995*. December 7, 1995 [cited 10 March2006];

Available: http://www.science.gmu.edu/~mchacko/csi801/proj-ckv.html.

[12] *How does cryptography work*. 12 March 2006 [cited 12 March 2006];
Available: http://www.pgpi.org/doc/pgpintro.