American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

Computer Arithmetic Aided Lempel-Ziv-Welch's Algorithm using the Moduli Set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ for Fast and Secured Transmission of Data via Network Communication Channels

[1]Alhassan Abdul-Barik, [2]Edem Kwedzo Bankas, and [3]Peter Awon-natemi Agbedemnab

[1, 2, 3]Computer Science Department, Faculty of Mathematical Sciences

University for Development Studies

P. O. Box 24 Navrongo, Upper East Region

Ghana, West Africa

Abstract: *Encryption is the process of securing information from unauthorised assess, whiles data compression involves the reduction of data volume for ease of storage or transmission. The reduction should enable the acquisition of the original message or its approximation. Several algorithms have been proposed for compressing or securing data. In this paper, a new efficient scheme is proposed for enhanced data security and compression. Computer arithmetic is efficiently applied to the Lempel-Ziv-Welch's (LZW) algorithm using the moduli set* $\{2^n - 1,\ 2^n,\ 2^n + 1,\ \ 2^{n+1} - 1\}$ *to design encoder and decoder pairs (which works for only even n numbers for enhanced security) for data conversion from one number representation to the other. In the encryption process, the modified LZW algorithm is used to encrypt data using computer arithmetic which results in four (4) residuals. These residuals are then archived or transmitted in a secret order bit stream. In the decryption process, the stored or sent secret order bit stream residual data is then received, reorganised, and converted to decimals using residue to decimal convertor or decoder to decode. The process continues until the original data is acquired back. The traditional LZW algorithm and the proposed scheme are then simulated using MatLab for performance evaluation. The proposed scheme shows better performance than the traditional LZW method and other known state of the art related schemes.*

Keywords: *Compression; Decoder; Encoder; LZW; RNS*

I. Introduction

Compression and encryption is an important aspect of data or information management and security. In a distributed environment, large volumes of data remains a gigantic problem that can easily be dealt with using compression and encryption. A number of researches exist in this area of data management and security. In essence, various compression algorithms exist for compressing different types of file formats including text, image, sound, video or a combination of these. These algorithms are either classified as lossy or lossless, and dictionary or non-dictionary based depending on the nature of the output for a specific input [1]-[4]. In improving the security and storage efficiencies of various algorithms, computer arithmetic has been efficiently applied. In [9], image security has been improved by using Residue Number System (RNS) and the method of Arnold's transforms. Similar research is done in [10] for improving the efficiency of the Huffman's method of data encoding where the frequency of occurrences of each character are used to generate binary codes to reduce data size and enhance security. There exist a lot of literature on improving the Lempel-Ziv-Welch (LZW) algorithm. Alhassan et al. [11] also applied RNS using the traditional moduli set to propose the LZW-RNS (3-Moduli) scheme which shows better performance than the traditional lossless dictionary-based LZW algorithm. Software implementations of the LZW algorithm are often not fast enough, particularly in transmitting data through high-speed media. The research for hardware implementation of binary data compression of the LZW is therefore presented in [3]. The LZW is modified by Kaur and Verma as content based addressable memory (CAM) array which utilises less bits than its ASCII code [12]. In [4] a comparative study of text compression algorithms is done where the LZW is found to be the least performing in terms of bits per compression (BPC). Parthasarathy et al. [13] in a research to determine the existence of secret information hidden within an image, LZW was used to manipulate 128-bit input to achieve encryption and decryption. Mahyar [14] noted that, developing 3G networks using KASUMI Block Cipher performs better that 2G and 2.5G networks. RNS is therefore used to deal with the inherent problem of errors emanating from the use of KASUMI Block Cipher.

This research therefore focuses on applying computer arithmetic using the moduli set $\{2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{n+1} - 1\}$ to the LZW algorithm and also modifying its output for residual archiving or transmission for efficient and secured data compression and encryption.

II. Materials and Methods

Lossless compression algorithms allow for decoding back the original data whiles lossy allows for an approximation of the original data [1]-[4], [12], [13].

A. LZW Algorithm

This is a universal lossless dictionary based data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch in 1984 as an improved implementation of the LZ78 algorithm published by Lempel and Ziv in 1978, which is simple to implement, and has the potential for very high throughput in hardware implementations [1]-[4].

B. The Residue Number System

Residue Number System (RNS) is defined by a basis consisting of a set of co-prime numbers, called moduli $\{m_1,\ m_2,\ m_3 \ldots m_k\}$, where the Greatest Common Divisor (GCD) between any two moduli is one, that is $\gcd(m_i, m_j) = 1, \forall\ i \neq j$. An integer X is represented by k-tuple $(x_1, x_2, \ldots, x_k)$ in RNS where the residue $x_i = |X|_{m_i}$ for $i = 1, 2, \ldots., k$, and $|X|_{m_i}$ is defined as $X\ mod\ m_i$. The Dynamic range of the RNS is given by $M = \prod_{i=1}^{k} m_i$. Using the Chinese Remainder Theorem (CRT), an integer X can be calculated from its residue digits $(x_1, x_2, \ldots, x_k)$ as follows;

$$X = \left| \left| \sum_{i=1}^{k} M_i^{-1} x_i \right|_{m_i} M_i \right|_M \qquad (1)$$

Where $M = \prod_{i=1}^{k} mi$, $Mi = \frac{M}{m_i}$, and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i.[5]-[8].

C. The Conversion Process

Forward convertor converts from binary/decimal to residue whiles reverse convertor converts from residue to binary/decimal representations respectively [6]-[8].

C.1 Forward Conversion Process for the Moduli Set $\{2^{n+1} - 1, 2^n - 1,\ 2^n,\ 2^n + 1\}$

Given the moduli set $\{2^{n+1} - 1, 2^n - 1,\ 2^n,\ 2^n + 1\}$,

where;

$m_1 = 2^n + 1$, $m_2 = 2^n$, $m_3 = 2^n - 1$ and $m_4 = 2^{n+1} - 1$. $M = \lfloor 0, 2^{4n+1} - 2^{2n+1} - 2^n - 1 \rfloor$ (Where the upper end of the range $(m_1 m_2 m_3 m_4)$, is uniquely defined by the residue set $\{r_1, r_2, r_3, r_4\}$).

X is a $4n$-bit number which can be represented as;

$X = x_{4n-1} x_{4n-2} \ldots x_1 x_0 \qquad (2)$

Since $r_i = |X|_{m_i}$, the $r_i's$ can be computed as follows;

r_2 is the n least significant bit (LSB) of X in binary.

For r_1, r_3 and r_4, we partition X into four (4) n-bit blocks B_1, B_2, B_3 and B_4 where;

$B_1 = \sum_{j=3n}^{4n-1} x_j 2^{j-3n}$, $B_2 = \sum_{j=2n}^{3n-1} x_j 2^{j-2n}$,

$$B_3 = \sum_{j=n}^{2n-1} x_j 2^{j-n} \ and\ B_4 = \sum_{j=0}^{n-1} x_j 2^j \quad \}(3)$$

Which implies;

$X = B_1 2^{3n} + B_2 2^{2n} + B_3 2^n + B_4 \quad (4)$

Therefore,

$r_1 = |X|_{2^n+1} = |B_1 2^{3n} + B_2 2^{2n} + B_3 2^n + B_4|_{2^n+1}$

$= ||B_1 2^{3n}|_{2^n+1} + |B_2 2^{2n}|_{2^n+1} + |B_2 2^n|_{2^n+1} + |B_3|_{2^n+1}|_{2^n+1}$

B_3 is $n - bit < 2^n + 1$ so,

$|2^{3n}|_{2^n+1} = |2^n \times 2^n \times 2^n|_{2^n+1}$

$= |-1 \times -1 \times -1|_{2^n+1} = -1$

$|2^{2n}|_{2^n+1} = |2^n \times 2^n|_{2^n+1}$

$= |-1 \times -1|_{2^n+1} = 1$

and

$|2^n|_{2^n+1} = -1$

Which implies;

$r_1 = |-B_1 + B_2 - B_3 + B_4|_{2^n+1}$

Similarly,

$r_3 = |X|_{2^n-1}$
Therefore $r_3 = |B_1 + B_2 + B_3 + B_4|_{2^n-1}$
Finally,
$r_4 = |X|_{2^{n+1}-1} = |B_1 2^{3n} + B_2 2^{2n} + B_3 2^n + B_4|_{2^{n+1}-1}$
$= ||B_1 2^{3n}|_{2^{n+1}-1} + |B_2 2^{2n}|_{2^{n+1}-1} + |B_2 2^n|_{2^{n+1}-1} + |B_3|_{2^{n+1}-1}|_{2^{n+1}-1}$
Thus, $|2^{3n}|_{2^{n+1}-1} = |2^{3n} - 2^{n+1} + 1|_{2^{n+1}-1} = 2^{n-2}$,
And $|2^{2n}|_{2^{n+1}-1} = |2^{2n} - 2^{n+1} + 1|_{2^{n+1}-1} = 2^{n-1}$
Since $2^n < 2^{n+1} - 1$ implies $|2^n|_{2^{n+1}-1} = 2^n$
Therefore,
$r_4 = |2^{n-2}B_1 + 2^{n-1}B_2 + 2^n B_3 + B_4|_{2^{n+1}-1}$

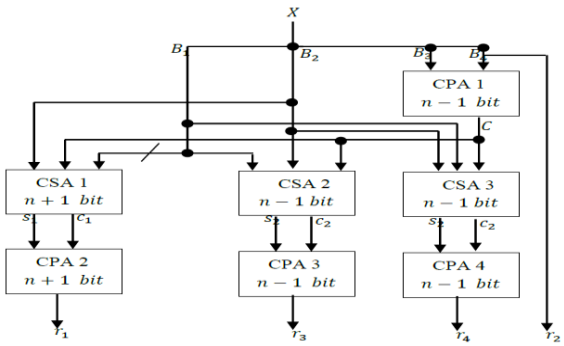
Example 1: Use of the Forward Convertor

Given the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, $n = 2$ and X representing a character of a message to be encoded, with ASCII code $X = 60$. Then the conversion process is;
$60 = 111100 = 00111100$
Thus, $B_1 = 00,\ B_2 = 11, B_3 = 11, and\ B_4 = 00$
Therefore,
$r_1 = |60|_{2^2+1} = |60|_5 = |0 + 0 - 0 + 0|_5 = 0,$
$r_2 = B_4 = 00 = 0$
$r_3 = |60|_{2^2-1} = |60|_3 = |0 + 3 + 3 + 0|_3 = 0.$
and,
$$r_4 = |60|_{2^{2+1}-1} = |60|_7 = ||00|_7 + |2^1(11)|_7 + |2^2(11)|_7 + |00|_7|_7 = |0 + 6 + 12 + 0|_7 = 4$$
And so
$|60|_{5|4|3|7} = (0,0,0,4)_{5|4|3|7}$

III. Hardware Realisation

In order to achieve the hardware realization of the forward converter, we further simplify equation (4) to;
$X = B_1 2^{3n} + B_2 2^{2n} + C$ (5)
Where,
$C = B_3 2^n + B_4$

Figure 1: Forward Converter



A. .2 Reverse Conversion Process for the Moduli Set $\{2^{n+1} - 1, 2^n - 1, 2^n, 2^n + 1\}$

The reverse convertor is designed to convert from RNS to binary/decimal using the moduli set $\{2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{n+1} - 1\}$ for the LZW decoder. Given an RNS number $X = (x_1, x_2, x_3, x_4)$, then the Chinese Remainder Theorem (CRT) is applied as follows;
From Equation (1), if $m_1 = 2^n - 1$, $m_2 = 2^n$, $m_3 = 2^n + 1$, and $m_4 = 2^{n+1} - 1$, we have [5]-[7], [9]:
$M_1 = 2^n(2^n + 1)(2^{n+1} - 1);\ M_2 = (2^{2n} - 1)(2^{n+1} - 1);\ M_3 = 2^n(2^n - 1)(2^{n+1} - 1);$ and $M_4 = 2^n(2^{2n} - 1)$ }(6)
Theorem 1: For the given moduli set, we have
$|M_1^{-1}|_{m_1} = |2^{n-1}|_{m_1}$ (7)
$|M_2^{-1}|_{m_2} = 1$ (8)
$|M_3^{-1}|_{m_3} = |-1|_{m_3}$ (9)
$|M_4^{-1}|_{m_4} = |2^{n-1}|_{m_4}$ (10)
Proof: If it can be demonstrated that $|M_i^{-1} \times M_i|_{m_i} = 1$, then M_i^{-1} is the multiplicative inverse of M_i with respect to m_i. Thus;
For (7), $|(2^{n-1}) \times 2^n(2^n + 1)(2^{n+1} - 1)|_{2^n-1}$
$$= |(2 - 2^n)|_{2^n-1} = |(2 - 1)|_{2^n-1} = 1$$
Hence 2^{n-1} is the multiplicative inverse of M_1 with respect to m_1

Also, for (8) $|(1)(2^{2n}-1)(2^{n+1}-1)|_{2^n} = |(-1)(-1)|_{2^n} = 1$
Hence 1 is the multiplicative inverse of M_2 with respect to m_2
Similarly, for (9), $|(-1)\times 2^n(2^n-1)(2^{n+1}-1)|_{2^n+1}$
$= |(-1)\times(-1)|_{2^n+1} = 1$
Hence (-1) is the multiplicative inverse of M_3 with respect to m_3.
Finally, for (10), $|(2^{n-1})\times 2^n(2^{2n}-1)|_{2^{n+1}-1}$
$= |(1)\times(1)|_{2^{n+1}-1} = 1$
Hence, (2^{n-1}) is the multiplicative inverse of M_4 with respect to m_4.

Theorem 2: For the given moduli set, any RNS number X can be represented as;

$$X = 2^n\xi + x_2 \qquad (11)$$

where;

$$\xi = \left\lfloor \left| \begin{matrix} |2^n a - a|_{m_1m_3m_4} + |2^{n+1}b - b|_{m_1m_3m_4} \\ +|c - 2^n c|_{m_1m_3m_4} + |2^{n-1}(2^n x_4 - x_4)|_{m_1m_3m_4} \end{matrix} \right|_{m_1m_3m_4} \right\rfloor \qquad (12)$$

and

$$a = 2^n x_1 + x_1 \qquad (13)$$
$$b = 2^{n+1}x_2 - x_2 \qquad (14)$$
$$c = 2^{n+1}x_3 - x_3 \qquad (15)$$

Proof: Substituting equations (6) through to (10) into (1) and factorizing out 2^n, we obtain (11).

i. Hardware Implementation

Equation (11) can further be simplified as follows

$$\xi = |A + B + C + D|_{2^{2n}-1} \qquad (16)$$

Where

$$A = |2^n a - a|_{m_1m_3m_4} \qquad (17)$$
$$B = |2^{n+1}b - b|_{m_1m_3m_4} \qquad (18)$$
$$C = |c - 2^n c|_{m_1m_3m_4} \qquad (19)$$
$$D = |2^{n-1}(2^n x_4 - x_4)|_{m_1m_3m_4} \qquad (20)$$

Equations (13)-(15) can be simplified for implementation as follows;

$$a = 2^n x_1 + x_1 = \underbrace{x_{1,n-1}\dots x_{1,0}\overbrace{0\dots0}^{n\ Bits}}_{2n-bits} \bowtie \underbrace{\overbrace{0\dots0}^{n\ Bits} x_{1,n-1}\dots x_{1,0}}_{2n}$$
$$= \underbrace{a_{2n-1}\dots a_1 a_0}_{2n-bits} \qquad (21)$$

$$b = 2^{n+1}x_2 - x_2 = x_{2,n-1}\dots x_{2,1}x_{2,0}\overbrace{00\dots0}^{n+1} + \overbrace{00\dots0}^{n+1}\overline{x}_{2,n-1}\dots\overline{x}_{2,1}\overline{x}_{2,0}$$
$$= b_{2n}\dots b_1 b_0 \qquad (22)$$

$$c = 2^{n+1}x_3 - x_3 = x_{3,n}\dots x_{3,1}x_{3,0}\overbrace{00\dots0}^{n+1} + \overbrace{00\dots0}^{n+1}\overline{x}_{3,n-1}\dots\overline{x}_{3,1}\overline{x}_{3,0}$$
$$= c_{n+1}\dots c_1 c_0 \qquad (23)$$

Also, equations (17) – (20) can be simplified for implementation as follows;

$$A = |2^n a - a|_{m_1m_3m_4}$$
$$= \left| \underbrace{a_{2n-1}\dots a_1 a_0}_{2n-bits}\overbrace{00\dots0}^{n} + \overbrace{00\dots0}^{n}\underbrace{\bar{a}_{2n-1}\dots\bar{a}_1\bar{a}_0}_{2n-bits} \right|_{(2^{2n}-1)(2^{n+1}-1)}$$
$$= \underbrace{A_{3n}\dots A_1 A_0}_{3n+1\ bits} \qquad (24)$$

$$B = |2^{n+1}b - b|_{m_1m_3m_4} = \left| \underbrace{b_{2n-1}\dots b_1 b_0}_{2n\ bits}\overbrace{00\dots0}^{n+1} + \overbrace{00\dots0}^{n}\underbrace{\bar{b}_{2n-1}\dots\bar{b}_1\bar{b}_0}_{2n\ bits} \right|_{(2^{2n}-1)(2^{n+1}-1)}$$
$$= \underbrace{B_{3n}\dots B_1 B_0}_{3n+1\ bits} \qquad (25)$$

$$C = |c - 2^n c|_{m_1m_3m_4} = \left| \overbrace{00\dots0}^{n}\underbrace{c_{2n}\dots c_1 c_0}_{2n+1\ bits} + \overbrace{00\dots0}^{n}\underbrace{\bar{c}_{2n}\dots\bar{c}_1\bar{c}_0}_{2n+1\ bits} \right|_{(2^{2n}-1)(2^{n+1}-1)}$$
$$= \underbrace{C_{3n}\dots C_1 C_0}_{3n+1\ bits} \qquad (26)$$

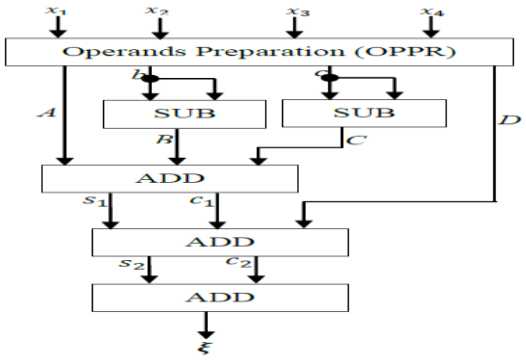
$$D = |2^{n-1}(2^n x_4 - x_4)|_{m_1m_3m_4} = \left| 2^{n-1}\left(\overbrace{00\dots0}^{n}\underbrace{c_{2n}\dots c_1 c_0}_{2n+1\ bits} + \overbrace{00\dots0}^{n}\underbrace{\bar{c}_{2n}\dots\bar{c}_1\bar{c}_0}_{2n+1\ bits} \right) \right|_{(2^{2n}-1)(2^{n+1}-1)}$$

$$= \left| 2^{n-1} \underbrace{D_{3n} \dots D_1 D_0}_{3n+1\ bits} \right|_{(2^{2n}-1)(2^{n+1}-1)}$$

$$= \underbrace{D_{2n} \dots D_0 D_{n-1}}_{3n+1\ bits} \quad (27)$$

Below is the schematic diagram of the proposed scheme.

Figure 2: Reverse Converter



Example 2: Use of the Reverse Converter

Given the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, $n = 2$ and $X = 60$ representing a character and ASCII code to be encoded. Then the conversion process is;

The moduli set is {5, 4, 3, 7};

$$M_1 = 84, \quad M_2 = 105, \quad M_3 = 140, \quad M_4 = 60;$$

$$|M_1^{-1}|_{m_1} = |84^{-1}|_5 = 4,$$

$$|M_2^{-1}|_{m_2} = |105^{-1}|_4 = 1,$$

$$|M_3^{-1}|_{m_3} = |140^{-1}|_3 = 2 \text{ and}$$

$$|M_4^{-1}|_{m_4} = |60^{-1}|_7 = 2$$

Therefore, from equation (1);

$$X = \left| \begin{matrix} |(84 \times 4 \times 0)|_{420} + |(105 \times 1 \times 0)|_{420} \\ +|(140 \times 2 \times 0)|_{420} + |(60 \times 2 \times 4)|_{420} \end{matrix} \right|_{420}$$

$$= |0 + 0 + 0 + 480|_{420} \quad = |60|_{420} = 60$$

These convertors are applied to the LZW algorithm for secured data encoding and decoding. The proposed scheme consists of a modified encoder and decoder pair with enhanced security and compression ratio.

IV. Results and Discussions

The proposed scheme consists of a modified encoder and decoder pair with enhanced compression ratio, speed, and security. Computer arithmetic is applied to the ASCII character with decimal representation X which is used in the encryption and decryption process using the proposed LZW scheme.

1.1 The Proposed Encoder and Decoder

The initial table/dictionary with single character codes in decimals is created and then converted into its residues in a process termed as forward conversion. The encoding process is continued with the modified algorithm in residues. The compressed or encoded message is then transmitted in four bit stream compartment in a particular secret order. The secret order transmitted four bit stream channel message is received, reorganised by the decoder pair and then converted to its decimal representations through a process known as reverse conversion. The modified LZW decoding process continues until the original message is acquired back.

The average CPU times taken for both the Encoder and Decoder as well as compressed files sizes is shown below.

Table 1: Results of LZW and LZW-RNS Scheme for $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, n=4

Document (kb) / Operations (s)	Original File (kb)	LZW Compression Algorithm			Proposed LZW-RNS(4-Moduli) Encryption Scheme		
		Encoder (s)	Decoder (s)	Compressed File(kb)	Encoder (s)	Decoder (s)	Compressed File(kb)
Document 1	23	6.2969	3.5781	17	5.9219	3.2813	16
Document 2	32	32.1406	8.1406	20	31.9688	8.1125	18
Document 3	36	90.5469	20.6250	27	89.5188	16.7188	20
Document 4	45	105.1836	24.7813	33	97.8990	20.8985	26
Document 5	50	118.7595	27.6458	36	105.3317	23.2206	28
Totals	**186**	**257.9199**	**62.6541**	**133**	**246.3743**	**53.6552**	**106**
Total Execution Time		**320.5741**			**300.0295**		

From Table 1, notice that there is an overall gain in execution time of the Proposed LZW-RNS (4-Moduli set) over the LZW, a gain in compression of over 15% in the LZW compressed file, and over 40% gain to the original file size. The time and compression efficiencies are shown in Figure 3 and Figure 4 respectively.

Figure 3: The Execution Time of LZW and Proposed LZW-RNS Scheme

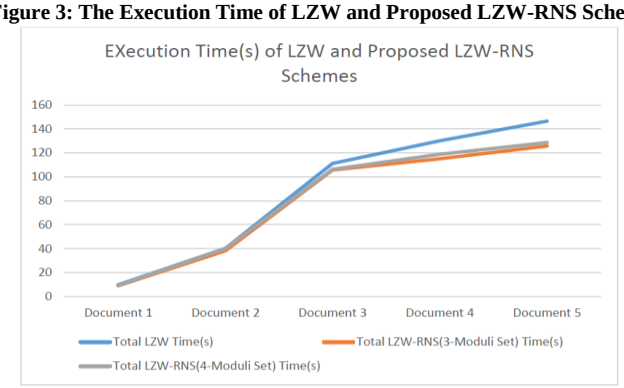
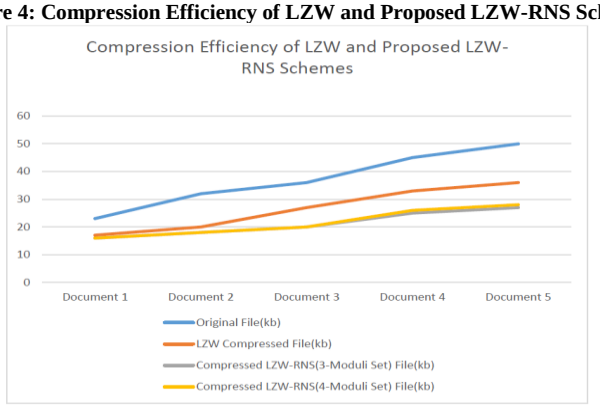


Figure 4: Compression Efficiency of LZW and Proposed LZW-RNS Scheme



1.2 Security and Efficiency of the Proposed Scheme

The purpose of data compression is to reduce data size, enhance speed of transmission, and security. The gain in this research in terms of security and efficiency are; reduced magnitude of computation (a property of RNS that uses residues of numbers that invariably speeds up compression), enhanced security (decoding the message requires knowledge of the coding scheme and the moduli set used), the encoder and decoder pairs are constrained to work for only even ***n*** numbers, and the secret transmission order needs to be known, before the actual residue representation can be determined, and reduction in the number of bits required for archival or storage purposes. special purposes.

V. Conclusions

In this paper, computer arithmetic has been applied to the Lempel-Ziv-Welch's (LZW) algorithm resulting in an encoder and decoder pair as well as modification of the output of the algorithm. The output modification allows for secret order bit stream transmission or storage of data. The issue of the unbounded growth of the dictionary has also been improved by the use of residues of the larger numbers. The transmission speed and storage space has also been enhanced.

The proposed scheme is lower in execution speed as by Alhassan et al. (2015), it is however high in security because of the larger channel that will have to be hacked, and the deliberate imposition on both the encoder and decoder pair to work or execute for only even ***n*** numbers.

VI. Future Research Work

Undoubtedly, the proposed system is efficient in terms of security. Notwithstanding the error correction capability of RNS, Redundant Residue Number System (RRNS) which is a trait of RNS will be applied for the purposes of error detection and correction.

References

[1] J. Amit, "Comparative Study of Dictionary Based Compression Algorithms on Text Data". International Journal of Computer Engineering and Applications, Vol I, Issue II, Pg.1-11, India.

[2] J. Trivedi, "A Survey on Different Compression Techniques Algorithm for Data Compression", International Journal of Advanced Research in Computer Science and Technology, Volume II, Issue III, Pg1-5, 2014.

[3] Md. Arif Sobhan, Md. Mamun, A. B. Ahmad Ashrif, and H. Hafizah, "Hardware Approach of Lempel-Ziv-Welch Algorithm for Binary Data Compression", World Applied Sciences Journal, Vol. 22(1), Pp 133-139, 2013.

[4] S. Shammugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithm", International Journal of Wisdom Based Computing, Vol. 1(3), India, Pp 68-76, 2012.

[5] K. A. Gbolagade, "Effective Reverse Conversion in Residue Number System Processors", PhD thesis, Delft University of Technology (TU-Delft), The Netherlands, Pp 1-187, 2010.

[6] A. Omondi and B. Pumkumar, "Residue Number Systems: Theory and Implementation", Imperial College Press, 57 Shelton street, Covent Garden, London WC2H 9HE, Pp 5-900, 2007.

[7] B. Pahami, "Computer Arithmetic Algorithms and Hardware Design", Department of Electrical and Computer Engineering, University of California, Santa Barabara, Oxford University Press, New York, Pp 2-200, 2000.

[8] P. V. A. Mohan and A. B. Preemkumar, (2007): RNS-to-Binary Converter for Four-Moduli Set $\{2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{n+1} - 1\}$, IEEE Transaction Circuits Systems, Reg. Papers, Vol. 54, pp 1245 – 1254.

[9] S. Alhassan, and K. A Gbolagade, "Enhancement of the Security of a Digital Image using the Moduli set $\{2^n\text{-}1,\ 2^n,\ 2^n\text{+}1\}$", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Vol. (2), Issue (7), Pp 2223-2229, 2013.

[10] A. Alhassan, I. Saeed, and P.A. Agbedemnab, "The Huffman's Method of Secured Data Encoding and Error Correction using Residue Number System (RNS)", Communication on Applied Electronics (CAE) Journal, Foundation of Computer Science (FCS), New York, USA, 2015.

[11] A. Alhassan, K. A. Gbolagade, and E. K. Bankas, "A Novel and Efficient LZW-RNS Scheme for Enhanced Information Compression and Security", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Vol4 (11), November, 2015. ISSN: 2278-1323. Pp 4015-4019.

[12] S. Kaur, and S. Verma, "Design and Implementation of LZW Data Compression Algorithm", International Journal of Information Sciences and Techniques (IJIST), Vol. 2(4), Pp 71-81, 2012.

[13] C. Parthasaraty, G. Kalpana, and V. Gnanachandran, "LZW Data Compression for ISP Algorithm", International Journal of Advanced Information Technology, Vol. 2(5), Pp 25-36, 2012.

[14] H. Mahyar, "Reliable and High Speed KASUMI Block Cipher by Residue Number System Code", World Applied Sciences Journal, Vol.17(9), Pp 1149-1158, 2012.