Application of Residue Number Systems in enhancing the transmission of secured videos

Akinbowale Nathaniel BABATUNDE[1], Abdulkarim Ayopo OLOYEDE[2]

[1] Department of Computer Science, Faculty of Communication and Information Technology, Kwara State University, Malete, Kwara State, Nigeria

[2] Department of Telecommunication, Faculty of Communication and Information Sciences, University of Ilorin, Kwara State, Nigeria

akinbowale.babatunde@kwasu.edu.ng, oloyede.aa@unilorin.edu.ng

Abstract: Full video Encryption has been established as the most suitable technique to guarantee the security of digital video data during transmission but cannot be widely adopted because of its large disk space and long processing time. Therefore, this paper aims at reducing the computational space complexity using Residue Number System (RNS). MPEG IV algorithm and encryption of pixel values using the traditional RNS moduli set were employed for the encryption while a modified reverse converter was used for decryption. JAVA was used for the implementation while an evaluation was performed to compare the disk size of the cipher and deciphered videos. The scheme outperformed the existing evaluated cryptosystems in terms of cipher size and the decrypted videos increased by 49% when compared with the original video. The space computational complexity problem in full video encryption was adequately reduced using RNS. It is recommended that RNS should be used for securing video transmission and storage.

Keywords: Forward converter, Reverse converter, Moduli set, Residual frame, Moving Pictures Experts Group IV (MPEG IV), Video encryption, Residue Number System (RNS).

1. Introduction

A digital video is an electronic representation of moving digital images. It combines a sequence of images to form a moving picture and sends a signal to a screen giving the order and rate at which the screen captures will be drawn. A video is composed of two (2) segments: a codec and a container. Frames are the individual images that make up a video, and each one is very similar to the ones next to it. Frames are individual images composed of a raster of pixels. Pixels are little squares with two properties: color and location (Babatunde et al., 2017; Babatunde, et al., 2016).

In recent years, researchers have shifted attention to securing video in storage and transmission (Babatunde, 2019a; Yogita, 2013). This is due to the widespread use of video in numerous devices and to the transfer of sensitive information such as medical, military, and governmental private information across an open network using multimedia data (internet) (Yogita, 2013; Darshana & Parvinder, 2012). On the internet, transmissions are increasingly susceptible to interception by unscrupulous and unauthorized users (John & Manimurugan, 2012). Over the years, Researchers have proposed various security measures for protecting these videos from unauthorized access including watermarks (Ajay et al., 2013), scrambling (Darshana & Parvinder, 2012), compaction (Shaheen et al., 2017), and steganography (Darshana & Parvinder, 2012; Yogita, 2013). However, video encryption in a compressed format is the most well-established and appropriate technique for addressing these security concerns (Ajay et al., 2013; Mayank et al., 2012). There are two major approaches to video encryption classification, each having various schemes (Babatunde, 2019a; Zhaopin et al., 2012). Video security algorithms can be either total encryption or selective encryption. In the former, the cryptosystem secures the video data in its entirety, frame by frame without taking into consideration any region of interest, while, in the latter, vital or relevant data are encrypted within the video frames (Babatunde et al., 2017; 2016; Wong & Bishop, 2006). Although the selective strategy minimizes computing cost by encrypting only a restricted collection of data, the security and speed of the algorithm are determined by the number of parameters it secures (Babatunde et al., 2017; 2016; Shah & Saxena, 2012).

The full encryption scheme is, however, unsuitable for developing video encryption systems because of high computational complexity and delay during transmission (Abomhara et al., 2010; Puech et al., 2012). This is why most video encryption techniques reported in the literature were developed using either permutation selective or perceptual encryption schemes.

https://doi.org/10.33436/v31i4y202108

Every digital device design is highly dependent on number systems (Babatunde et al., 2019; Abdulbarik, 2016). The conventional (Weighted Number System) and the unconventional (Number System) are the two different kinds of number systems. The weighted number system is used extensively in digital devices (Babatunde, 2019b; Abdulbarik, 2016). The carry propagation chains are the main problem in the weighted number system. In order to increase the performance of processors designed around it in terms of speed and area cost, it is necessary to eliminate the inherent carry propagation chains (Gbolagade, 2010). Residue Numbers System is an alternative here as it splits a large number into smaller ones such that arithmetic operations are performed on smaller numbers, thereby removing carry propagations in additions, subtractions, and multiplications (Babatunde, 2019a).

The Residue Number System (RNS) is an integer system that increases the speed of mathematical operation by separating the numbers into smaller units that are independent of one another. (Alhassan, 2013; Gbolagade & Cotofana, 2009). A binary-to-residue converter, residue arithmetic units, and a residue-to-binary converter are the three primary components of a RNS architecture. The two most crucial components for a successful RNS implementation are data conversion and moduli selection (Babatunde, 2019a; Baagyere et al., 2011).

The paper intends to apply the RNS and MPEG IV algorithm in video security. The work presents a cryptographic scheme for enhancing video transmission using a proposed modification of the traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$. The rest of the paper is structured as follows: a brief review of the literature in section 2, the discussion of the scheme in section 3. Sections 4 and 5 present the discussion of experimental results, conclusion, and challenges for the future, respectively.

2. Literature review

RNS is a set of relative prime moduli $\{m_1, m_2, m_3, \ldots, m_n\}$ such that $gcd(m_i,m_j)=1$ for $i \neq j$, where *gcd* denotes the greatest common divisor m_i and m_j while $M = \prod_{i=1}^{n} m_i$ denotes the dynamic range (Gbolagade, 2010; Chaifali et al., 2001). RNS is capable of uniquely recording any integer X that falls within its dynamic range, that is, $(0 \leq X \leq M)$ where the dynamic range (M) is determined by the multiplication of the moduli sets (Omar, 2011). However, an overflow occurs when the outcome of a calculation exceeds M (Dynamic Range). While forward conversion is the process of transforming a conventional number into its residual form using moduli sets, the reverse conversion is the process of converting a residue representation to a conventional number form using reverse converters. Reverse Conversion in a RNS system which can be performed with either the Chinese Remainder Theorem (CRT) or Mixed Radix Conversion (MRC) is the most difficult element of any RNS architecture (MRC). All other developed or designed algorithms take their basis from these two methods.

Video compression is a technique for changing video signals that aim to retain the initial quality under several constraints taking advantage of the data redundancy within and between successive video frames (Suganya & Mahesh, 2014). Redundancy can be removed either temporally or spatially, and compression can be lossy or lossless (Djordje, 2009). Many video compression and decompression algorithms such as the intel RTV/indeo, MPEG-2 Part 2, H.262, H.263, H.264/AVC, MPEG-4, etc. were reported in the last four decades.

The basic principle behind MPEG (Moving Picture Coding Experts Group) compression is to turn a stream of samples into a bitstream of tokens in order to reduce the amount of required space. MPEG reduces the size of video streams by using the temporal relationship between successive frames (Hosseini, 2012). An example of an MPEG algorithm may be found in MPEG-4, which was introduced in late 1998 and marked a significant improvement over MPEG 2. It was created with a bit rate of 10 kbits/sec to 1 Mbits/sec in mind for use in interactive situations. The MPEG-4 syntactic description language allows the content of a frame to be grouped into objects that can be accessed independently. MPEG has been applied in so many applications in real life, such as in cable, television, direct broadcast, satellite, real-time encoding, computer network, and so on.

Owing to the rapid development of multimedia video reduction techniques and the recent successes achieved in internet technology, the security of continuous images is becoming increasingly vital. These accomplishments have allowed films to be utilized as a medium for storing and transmitting secret information. As a result, video data must be safeguarded against illegal access during transmission and storage. Video encryption is a well-known and secure method of protecting video content. (Babatunde, 2019a; John & Manimurugan, 2012). Completely layered encryption, permutation-based encryption, selective encryption, and perceptual encryption are the four primary categories of encryption that have been identified (Babatunde, 2019a; Zhaopin et al., 2012).

Many video security algorithms such as those discussed in the work of (Aman et al., 2015; Rajagopal & Shenbagavalli, 2013; Ajay et al., 2013; Yogita, 2013; Darshana & Parvinder, 2012; Zhaopin et al., 2012; Saranya & Varalakshimi, 2011; Yan & Main, 2009; Jakimoski & Subbalakshimi, 2008; Zhou et al., 2007; Li et al., 2002; Chiaraluce et al., 2002; Wu & Kuo, 2005;Alattar et al., 1999; Shi & Bhawgava, 1998; Qiao & Nahrstedt, 1997) and so on have been reported in the literature. All these reported authors and many others in the literature worked based on the three other video encryption techniques aside from naïve (full video encryption) and these techniques suffers from security issues (Babatunde, 2019a; Shah & Saxena, 2012).

This research work will address this problem by developing a scheme that protects the whole video content using MPEG IV and a modified Traditional RNS moduli set.

3. Proposed research design

The skeletal framework of the proposed cryptosystem, which is shown in Figure 1, is divided into phases.

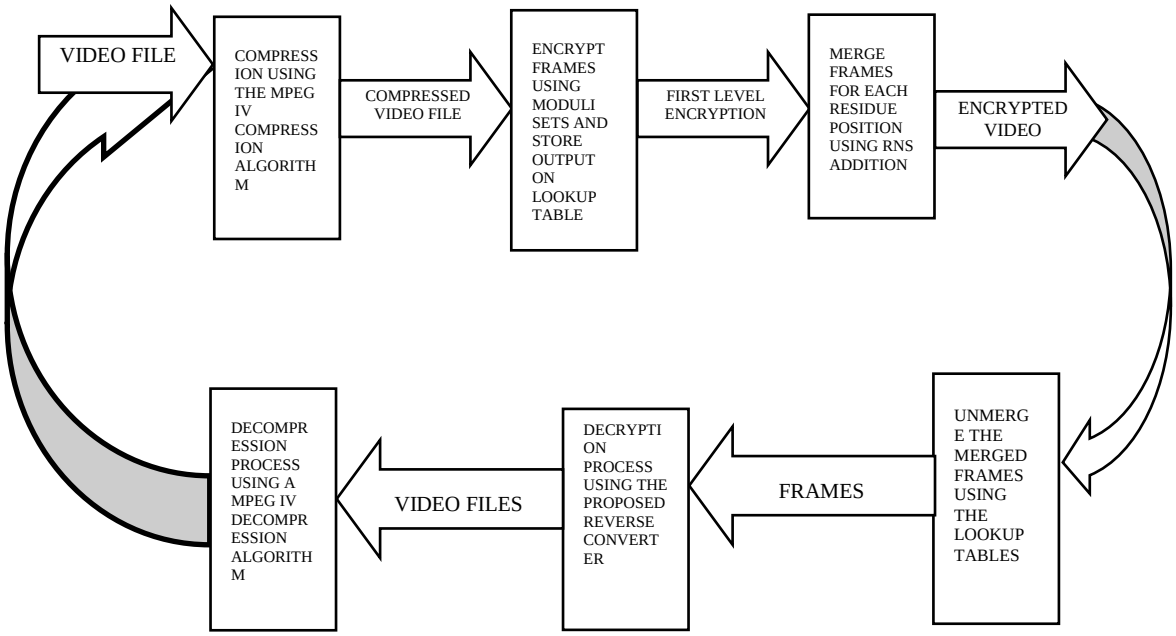

Figure 1. Data flow diagram for the cryptosystems

The Encryption algorithm part was designed using MPEG IV compression algorithm with the forward conversion process of the traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ while the decryption process used the reverse conversion process and MPEG IV decompression algorithm. The forward and reverse conversion processes of the traditional moduli sets $\{2^n+1, 2^n, 2^n-1\}$ were reported in (Abdul-Barik et al., 2015; Mohan, 2002)The two phases were implemented in the JAVA programming language.

The encryption phase of the cryptosystem is made up of the following steps:

i. Send in the video file to the system
ii. Compress the video file using the MPEG IV compression algorithm such that functionality that separates the compressed video into frames is achieved (included).
iii. The pixel values of the frames, which are now digital images, are extracted.
iv. The traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ is applied to the pixel value at each location to obtain the residue of the pixel values in the forward conversion process. The residues are being saved on the lookup table. This forms the first encryption stage of the system.
v. To avoid overflow during the reverse conversion process, a further step that computes the integer division value of each pixel whose value is greater than the dynamic range (DR) was included. The integer division value, which is computed by calculating the quotient from the division of each pixel value by the dynamic range, is saved alongside the residues on the lookup table. This forms the modification to the forward conversion process and an additional process.
vi. To increase the security and reduce the size of the video file, the output of the first stage of encryption is subjected to RNS column addition followed by an RNS row addition. In a bid to further reduce the size of the encrypted video, the result of RNS row addition is followed by scaling (division by several moduli sets).
vii. Merge the residue of each sub-frame of the video file by an RNS operation using the same moduli set for each position.
viii. The result (encrypted video) file and lookup table are however stored or transmitted over the internet.

Step (iv) as reported in (Babatunde, 2019a; Abdul-Abdul-Barik, 2016) alongside the hardware realization and the modification described in (v) was reported in (Babatunde, 2019a).

For the RNS addition and scaling, the easiest means to implement two-operand modular addition for any modulus m, $|A+B|_m$ is to perform two additions modulus 2^k (with $m \leq 2^k$). If the result of A and B exceeds the modulus (m), subtracting the value of the modulus gives the correct result, while if m is less than the addition of A and B, the answer of A and B is the correct result.

$$\text{That is, } \quad |A+B|_m = \begin{cases} A+B & \textit{If } A+B<m \\ A+B-m & \textit{Otherwise} \end{cases}$$

However, to speed up the operation, we can execute two (2) operations in parallel, that is S= (A+B) and S''= (A+B-m). If the three-term addition is negative, it means that the sum (A+B) < m and the modular sum is A+B. Otherwise, addition is the result of the three-term addition.

Figure 2 shows the process flow for the RNS based Video Encryption System for the modified $\{2^n+1, 2^n, 2^n-1\}$.

The Decryption process which uses the MPEG IV decompression algorithm with the modified reverse converters of the traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ is described in the process reported in (Abdul-Abdul-Barik, 2016; Bankas & Gbolagade, 2013).

The decryption phase of the cryptosystem is made up of the following steps:

i. Send in the encrypted video file alongside the lookup table.
ii. Unmerge the merged frames using lookup tables.
iii. Perform reverse conversion for each frame using the modified traditional reverse conversion equation $\left(X = (m_2\left\lfloor \frac{X}{m_2} \right\rfloor + x_2) + (ID * DR)\right)$
where $\left\lfloor \frac{X}{m_2} \right\rfloor = \left| -2^{n-1} m_3 x_1 - m_2 x_2 + 2^{n-1} m_1 x_3 \right|_{m_1 m_3}$.
iv. Merge the frames to form the original video.

v. Decompress the video file using MPEG IV decompression algorithm
vi. Save the new video

The flowchart and pseudocode for the encryption and decryption processes of the developed cryptosystem were presented in (Babatunde, 2019a). The proposed decoder receives the encrypted video and retrieves the subframes using the implemented reverse converters from the lookup table. This retrieval converts the RNS representation into its binary/decimal form. The subframes are merged to form the compressed video file, which is then decompressed to obtain the initial video file.

Testing the reverse converter

Given the moduli set $\{2^n+1, 2^n, 2^n-1\}$ for n=3 where $m_1=\{2^n+1\}$, $m_2=\{2^n\}$ and $m_3=2^n-1\}$. The residue representation for 950 which is representing X is $|950|_{9,8,7}=\{5,6,5\}$ for the forward conversion (Encryption).

For the reverse conversion, using the equation in (iii) in the decryption process. From the encryption process, ID (Quotient) = 1 while the DR (Dynamic range) = 504

$$X=(m_2\left\lfloor\frac{X}{m_2}\right\rfloor+x_2)+(ID*DR)$$

Where $\left\lfloor\frac{X}{m_2}\right\rfloor=|-2^{n-1}m_3x_1-m_2x_2+2^{n-1}m_1x_3|_{m_1m_3}$

$$=|(-2^{3-1}*7*5)-(8*6)+(2^{3-1}*9*5)|_{9*7}$$
$$=|-140-48+180|_{63}=|-8|_{63}=55$$
$$=8*55+6$$
$$=440+6$$
$$=446$$
$$ID*DR=1*504$$
$$=504$$
$$X=446+504$$
$$X=950$$

Hence, the reverse converter worked properly, as it successfully retrieved 950 back.

4. Result presentation and discussion

This section presents the implementation of the proposed scheme. JAVA programming language was used to implement the scheme, and a summary of the analysis conducted mainly in the visual test and the encoding analysis are presented below.

4.1. Visual Test

Six videos of various sizes and formats were used to test the implemented system. Figures 2-6 show the stages involved in a sample testing of a video on the developed system, from the interface as presented on Figure 2 to the point at which the video file is finally encrypted and played alongside the decrypted video. Figure 3 shows the output from the compression by MPEG IV algorithm, Figure 4 displays the counting and separation of compressed video into frames and the extraction of pixel values. The encryption and output information stages are displayed in Figure 5.

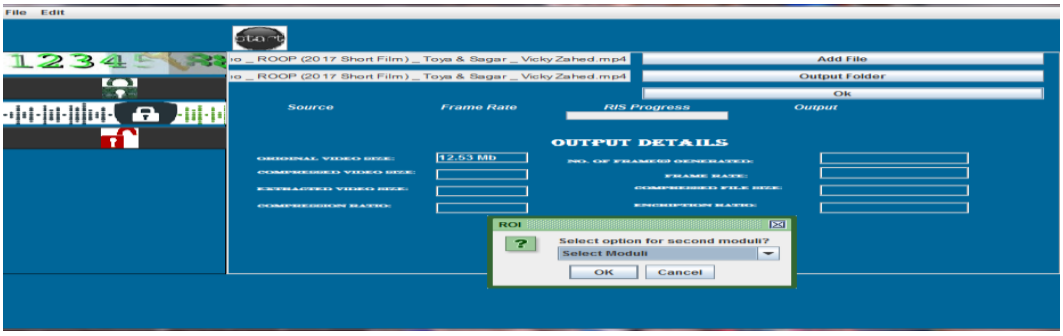

Figure 2. The interface of the developed system

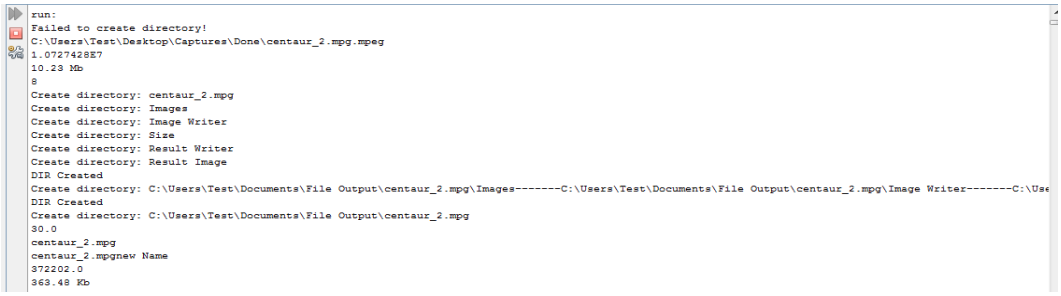

Figure 3. Output from the compression by MPEG IV algorithm

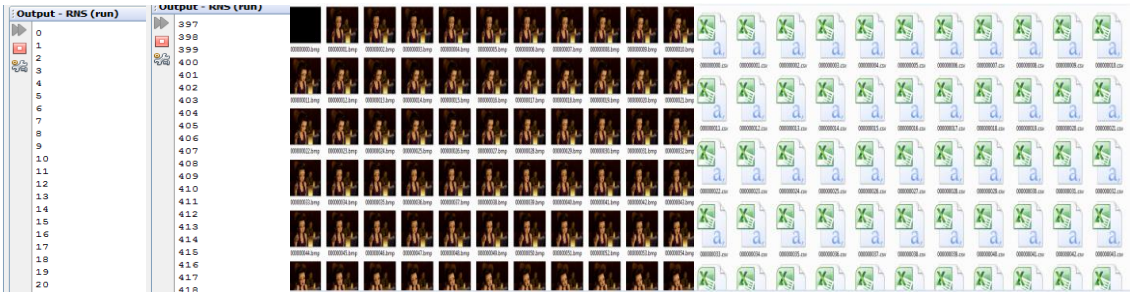

Figure 4. Counting and separation of compressed video into frames and extraction of pixel values

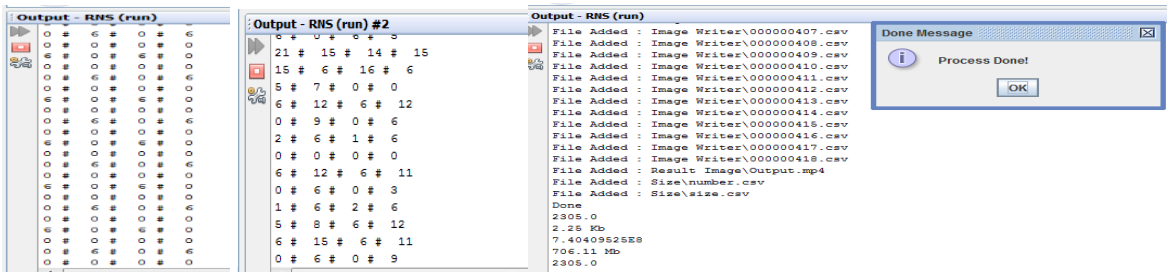

Figure 5. Encryption stages and output information

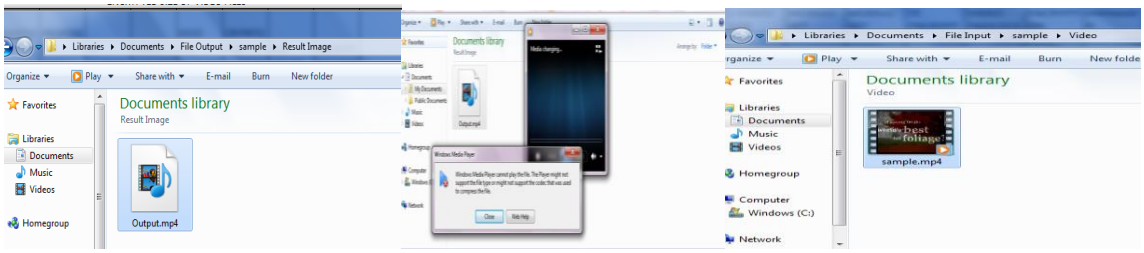

Figure 6. encrypted and decrypted files testing

After testing the application with the sample video, the encrypted video was played using Windows Media Player of windows 8 and 10, and versions 3.0.7 and 3.011 of VLC. None of the two could play the encrypted video file. Besides, owing to the small size of the video cipher, it was converted to a digital image and no meaningful information could be deduced from the image when opened, as shown in Figure 7.

Figure 7. Output when the encrypted file was converted into PNG format

The tested ciphers alongside the lookup table were loaded into the decryption system of the application and the original video with the same format was retrieved. It played the content of the initial video properly, without any information loss or picture quality decrease. However, the size of the decrypted video increased in size by about 70% when compared with the size of the original video, as shown in Table 1.

4.2. Encoding Analysis

The usefulness and efficiency of a data encoding algorithm are determined by how simple it is to implement the compression ratio (reducing disk size), by data transmission speed, and security improvement (Abdul-Barik, 2016; Ammar, et al., 2001). The performance analysis of these RNS-MPEG IV encoding algorithms for secured video transmission is performed on the disk size and execution time. In this study, the system was tested with six video files of various sizes and different formats on two different computers with Window 8 and 10 versions. Three videos were tested by each system and the result is presented in Table 1. From this table, it can be noticed that there is an overall gain in the disk size of the proposed video encoding scheme when compared with the sizes of the initial and compressed videos. The compression and modulus activities have greatly reduced the pixel values of each frame and the overall disk space of the videos tested. While the reduction in size speeds up data transmission across networks, the pixel value reduction can be useful in speeding up computational operations that might be useful in digital signal processing.

Furthermore, from Figure 7, it can be deduced that the media player (VLC and Microsoft Media Player) do not support the present codec of the video cipher, and thus no video stream was generated or shown. Hence, it can be **theoretically** concluded that the video ciphers obtained from the scheme are secured.

Table 1. Size and execution time of the proposed scheme

		Video Information				Size (KB)		Time (Sec.)	
S/NO	Video file	Size (KB)	Format	No of Frames	Compressed size (KB)	Encrypted	Decrypted	Encoding	Decoding
1	Video 1	2020	3GPP	813	679.02	16.9	3353	13.7	9.2
2	Video 2	1130	AVI	269	518.11	6.65	1798	8.2	5.8
3	Video 3	62.36	MKV	73	50.78	6.21	169	2.4	1.9
4	Video 4	4520	MP4	3416	3090	1.9	6467	19.3	14.3
5	Video 5	3940	MPEG	2565	3910	2.1	6737.4	17.2	12.7
6	Video 6	10280	MP4	418	363.48	2.25	17579	27.3	19.7

Three video cryptosystems (two from the best video encryption software for Windows with an algorithm downloaded online and implemented in JAVA programming language) were used to test the same videos the developed cryptosystem tested and the videos were successfully encrypted and decrypted. Tables 2 and 3 shows the results of the evaluation for compression efficiency and execution time, respectively, for the proposed scheme and the three other cryptosystems. From Table 2, it can be noticed that the proposed scheme outperformed the three other cryptosystems in terms of cipher disk size. In contrast, the decrypted cipher size increased in size when compared with the original video by between 42 and 84%.

However, owing to the complexity regarding the operations of the developed cryptosystem, the three tested cryptosystems outperformed the scheme in terms of computational time for the encryption and decryption processes, as shown in Table 3.

Table 2. Comparison of the compression (size) efficiency of the proposed scheme with 3 other encryption algorithms

				RNS BASED		COMPARED VIDEO ENCRYPTION APPLICATIONS					
		Video Information		Proposed sizes (KB)		IDOO (KB)		7-ZIP (KB)		VIDEO ENCRYPTOR (KB)	
S/NO	Video file	Size (KB)	Format	Encrypted	Decrypted	Encrypted	Decrypted	Encrypted	Decrypted	Encrypted	Decrypted
1	Video 1	2020	3GPP	16.9	3353	1900	2020	1950	2020	1940	2020
2	Video 2	1130	AVI	6.65	1798	1110	1130	1090	1130	1310	1130
3	Video 3	62.36	MKV	6.21	169	56.3	62.36	56.1	62.36	58.6	62.36
4	Video 4	4520	MP4	1.9	6467	4120	4520	4180	4520	4230	4520
5	Video 5	3940	MPEG	2.1	6737.4	3510	3940	3560	3940	3610	3940
6	Video 6	10280	MP4	2.25	17579	8660	10280	8720	10280	8690	10280

Table 3. Comparison of the execution time of the proposed scheme with 3 others encryption algorithms

				RNS BASED		COMPARED VIDEO ENCRYPTION APPLICATIONS					
		Video Information		Proposed sizes (SEC)		IDOO (SEC)		7-ZIP (SEC)		VIDEO ENCRYPTOR (S	
S/NO	Video file	Size (KB)	Format	Encrypted	Decrypted	Encrypted	Decrypted	Encrypted	Decrypted	Encrypted	Decrypted
1	Video 1	2020	3GPP	13.7	9.2	5.2	4.6	3.9	3.6	5.4	4.9
2	Video 2	1130	AVI	8.2	5.8	4.3	3.9	2.6	2.2	5.1	4.2
3	Video 3	62.36	MKV	2.4	1.9	1.2	1	0.7	0.7	1.4	1.2
4	Video 4	4520	MP4	19.3	14.3	8	6.2	5.3	4.8	7.6	5.8
5	Video 5	3940	MPEG	17.2	12.7	6.8	4.9	3.2	2.9	6.9	4.8
6	Video 6	10280	MP4	27.3	19.7	12.2	11	9.1	8.3	13.2	11.2

5. Conclusion and future work

In the paper, RNS has been applied as a post-compression technique on the MPEG-IV compression algorithm using a modified traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ which produced a video cryptosystem. The RNS-based video encoder and decoder pair was successfully designed using the forward converter for the encryption phase, residue representation for the video ciphers, and reverse conversion for the decryption process. The designs were implemented in JAVA programming language, and the systems were tested and compared with other video cryptosystems. The systems were able to encrypt and decrypt various formats and sizes of the videos, and they outperformed the available cryptosystems in cipher disk sizes. However, the cryptosystems in the literature outperformed the new systems in terms of encryption and decryption time (computational time). This is as a result of the several processes involved in the encryption and decryption processes of the cryptosystem. The scheme has resulted in the development of a new video encoding and decoding scheme to enhance speed and security during transmission and storage of videos.

Although this research work has presented the application of RNS in the enhancement of MPEG IV compression algorithm for very fast and easy transmission of video ciphers, it has drawbacks in terms of execution time and increased size of the decrypted cipher. These

disadvantages represent two major challenges the researchers are planning to look into in the nearest future. Furthermore, the researchers would like to investigate so as to confirm the proof of (Yogita, 2013; Darshana and Parvinder, 2012) on the level of security of total video encryption schemes by performing security analysis on the newly developed scheme.

REFERENCES

1. Abdul-Barik, A, Gbolagade, K.A. & Edem, K.B. (2015). *A novel and efficient LZW-RNS Scheme for enhanced information compression and security*. International Journal of advanced research in computer engineering and technology (IJARCET). 4(1). 4015-4019.
2. Abdul-Barik, A. (2016). *Enhancement of the security and compression of the Lempel-Ziv-Welch's algorithm using residue number system for efficient transmission via network communication channels*. Thesis submitted to the Department of Mathematics, Faculty of Mathematical Sciences, University of Development Studies, Ghana.
3. Alattar, A. M., Al-Regib, G. I. & Al-Semari, S. A. (1999). *Improved selective encryption techniques for secure transmission on MPEG video bitstreams.* Proceedings of International Conference on Image Processing (Cat. 99CH36348), vol. 4 (pp. 256-260). DOI: 10.1109/ICIP.1999.819590.
4. Alhassan, S. (2013). *Enhancement of Security of Digital Image Using the Moduli Sets*. Thesis submitted to the Department of Mathematics, Faculty of Mathematical Sciences, University of Development Studies, Ghana.
5. Aman, C., Sushmit, M., Ankit, C., Ravdeep, J. & Roja, M. M. (2015). *Dual-Layer Video Encryption using RSA Algorithm*. International Journal of Computer Application, 116(1), 33-40.
6. Ammar, A., Alkabbany, A., Youssef, M. & Emam, A. (2001). *A secure image coding scheme using RNS.* In Eighteenth National Radio Science Conference, Mansoura University, Eygpt.
7. Baagyere, E. Y., Boateng, K. O. & Gbolagade, K. A. (2011). *Bioinformatics: An important application area of RNS.* Journal of Engineering and Applied Sciences, 6(2),174 -179.
8. Babatunde, A. N. (2019a). *Enhanced Video Transmission Cryptographic Scheme Using Residue Number System*. Ph.D. Thesis submitted to the Department of Computer Science, University of Ilorin.
9. Babatunde, A. N., Jimoh, E. R, Oshodi, O. & Alabi, O. A. (2019). *Performance Analysis of Gray Code Number System in Image Security*. Jurnal Teknologi Dan Sistem Komputer, 7(4), 141-146.
10. Babatunde, A. N. (2019b). *Methodology for Image Cryptosystem Based on Gray Code Number System.* Computing and Information Systems Journal, 23(2), 1-10.
11. Babatunde, A. N., Jimoh, R. G., Abikoye, O. C. & Isiaka, B. Y. (2017). *Survey of Video Encryption Algorithms.* Covenant Journal of Informatics and Communication Technology, 5(1), 65-80.
12. Babatunde, A. N., Jimoh, R. G. & Gbolagade, K. A. (2016). *An Algorithm for a Residue Number System Based Video Encryption System*. Annals. Computer Science Series Journal, 14(2), 136-147.
13. Bankas, E. K. & Gbolagade, K. A. (2013). *An effective new CRT based converter for a novel moduli set* $\{2^{2n+1}-1, 2^{2n}, 2^{2n}-1\}$. In 24th International Conference on Application-specific Systems, Architecture, and Processors (ASAP2013), Washington, USA (pp. 142-146).

14. Chaifali, B. D., Partha, G. & Amitabha, S. (2001). *Estimation of hardware complexity of residue number system for signal processing application*. International Journal of Computer Technology Application, 2(5), 1540-1547.

15. Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P. & Reginelli, M. (2002). *A new chaotic algorithm for video encryption.* IEEE Transactions on Consumer Electronics, 48(4), 833-844.

16. Darshana, H. & Parinder, S. (2012). *A comprehensive survey of video encryption algorithms*. International Journal of Computer Applications, 59(1), 14-19.

17. Gbolagade, K. A. & Sorin, D. C. (2008). *A residue to binary converter for the {2^n+1, 2^n, 2^n-1} moduli set*. In 2008 42nd Asilomar Conference on Signals, Systems and Computers (pp. 1785-1789).

18. Gbolagade, K. A. & Sorin, D. C. (2008). *RNS operands to decimal conversion for 3- moduli sets*. In 51st Midwest Symposium on Circuits and Systems (pp. 791-794). DOI: 10.1109/MWSCAS.2008.4616918.

19. Gbolagade, K. A. & Cotofana, D. (2009). *Residue-to-Decimal converters for moduli sets with common factors.* In 52nd IEEE International Midwest Symposium on Circuits and Systems (pp. 624-627). DOI: 10.1109/MWSCAS.2009.5236017.

20. Gbolagade, K. A (2010). *Effective Reverse Conversion in RNS Processors.* Thesis submitted to Delft University, Netherlands.

21. Hosseini, 2012, M. (2012). *A survey of data compression algorithms and their applications*. Network Systems Laboratory, School of Computing Science, Simon Fraser University, BC, Canada.

22. Jakimoski, G. & Subbalakshimi, K. P. (2008). *Cryptanalysis of some multimedia encryption schemes*. IEEE Transactions on Multimedia, 10(3), 330-338.

23. John, J. & Mamimurugan, S. (2012). *A survey of various encryption techniques*. International Journal of Soft Computing and Engineering (IJSCE), 2(1), 322-325.

24. Li, S., Zheng, X., Mou, X. & Cai, Y. (2002). *Chaotic Encryption Scheme for Real-time Digital Video*. In Proceedings of SPIE, Electronic Imaging, Real-time Imaging, 4666, pp. 149-160.

25. Mayank, A. C., Ravindra, P. & Navin, R. (2012). *A novel approach to digital video encryption.* International Journal of Computer Applications, 49(4), 38-42.

26. Mohan, P. V. A. (2002*). Residue Number Systems: Algorithm and Architectures*. Kluwer Academic Publishers, Dordrecht.

27. Puech, W., Erkin, Z., Barni, M., Rane, S. & Lagendijk, R. L. (2012). *Emerging cryptographic challenges in image and video processing*. In *ICIP:* International Conference on Image Processing, Orlando, Florida, United States, pp. 2629-2632. DOI: 10.1109/ICIP.2012.6467438.

28. Qiao, L. & Nahrstedt, K. (1997). *A new algorithm for MPEG video encryption*. In Proceedings of the First International Conference on Imaging Science, Systems, and Technology (CISST'97), Las Vegas, Nevada, pp. 21-29.

29. Rajagopal, S. & Shenbagavalli, A. (2013). *A survey of video encryption algorithm implemented in various stages of compression*, International Journal of Engineering Research and Technology (IJERT), 2(2), 1-12.

30. Rooju, G. C. (2008). *RNS Enhancement for Programmable Processors.* Thesis submitted to Arizona State University.

31. Saranya, P. & Varalakshmi, L. M. (2011). *H.264 based selective video encryption for mobile applications*. International Journal of Computer Applications, 17(4), 13-20.

32. Shah, J. & Saxena, V. (2012). *Video encryption: A Survey*. International Journal of Computer Science Issues. 8(2), 525-534.

33. Shi, C. & Bhargava, B. (1998). *A fast MPEG video encryption*. In Proceedings of the 6th ACM International Conference on Multimedia, New York, USA, pp. 81-88.

34. Suganya, G. & Mahesh, K. (2014). *A survey of various techniques of video compression*. International Journal of Engineering Trends and Technology (IJERT), 7(1), 10-12.

35. Szabo, N. & Tanko, Y. (1967). *Residue arithmetic and its Application to computer technology*. Mc-Graw Hill.

36. Wong, A. & Bishop, W. (2006). *An efficient parallel multi-key encryption of compressed video streams.* In 8th IASTED International Conference on Signal and Image Processing, vol. 2 (pp. 69-74).

37. Wu, C. P., Kuo, C. C. (2005). *Design of integrated multimedia compression and encryption systems*. IEEE Transaction of Multimedia, 7(5), 828-839.

38. Yan, L. & Main, C. (2009). *H.264-Based multiple security levels net video encryption scheme*. In International Conference on Electronic Computer Technology, pp. 8-11.

39. Yogita, N. (2013). *A survey of video encryption techniques*. International Journal of Emerging Technology and Advanced Engineering, 3(4), 234-237.

40. Zhaopin, S., Guofu, Z. & Jianguo, J. (2012). *Multimedia security, a survey of chaos-based encryption technology multimedia*. Multimedia - A Multidisciplinary Approach to Complex Issues, 99-124. School of Computer and Information, Hefei University of Technology, China.

41. Zhou, Z., Liang, Z., Chen, Y. & Au, C.O. (2007). *Security analysis of multimedia encryption schemes based on multiple Huffman tables*. IEEE Signal Processing Letters, 14(5), 201-204.

Akinbowale Nathaniel BABATUNDE received B.Sc, M.Sc and Ph.D. degrees from the Computer Science Department, University of Ilorin. He joined Kwara State University, Malete as a Lecturer in the Department of Computer Science, in 2013. He has authored and co-authored several publications from both international and national journals. He is a member of Computer Professionals of Nigeria, Institute of Electrical and Electronics Engineers (IEEE) and of the Internet Society. His research interest include information security, computer arithmetic and natural language processing.

Abdulkarim Ayopo OLOYEDE received his B.Sc degree from Bayero University in Kano, in 2008, and his M.Sc and Ph.D. degrees from The Department of Electronics Engineering University of York in 2011 and in 2015, respectively. He teaches and conducts research activity in the Department of Telecommunication, University of Ilorin. He is the Vice-Chairman of ITU-D Study for the Technology Development Advisory Group (TDAG) and has published over 60 articles.