# Image Encryption System based on Length Three Moduli Set

Damilare Peter Oyinloye
Department of Computer Science
College of Information and Communication Technology
Kwara State University, Malete, Nigeria

## ABSTRACT
Digital images have found usage in almost all our everyday applications. These images sometimes contain confidential and intelligible information which need to be protected when stored on memory or transmitted over networks (Intranet or Internet). Many techniques have been proposed to deal with this security issues in the past. This paper proposes a simple scrambling algorithm to encrypt and decrypt the grey level image based on random number generation and Residue Number System (Forward and Reverse Conversion). The image is first encrypted by changing the position of each pixel in the original image without changing the value of grey level. The original image reads row by row, pixel by pixel and each pixel will take a new position in the scrambled image. The new position is chosen based on random number generation from the random number generator. The key will be generated as a matrix during the encryption process and also the key saves the position of each pixel in the encrypted/scrambled image. The encryption layer transforms the scrambled image to moduli images which automatically adds an extra security layer to our data.

The encrypted moduli images is decrypted by decoding the moduli images and converting them back to a single scrambled image (Reverse Conversion) and the single scrambled image back to the Plain and Original Image by using the saved key matrix. This scheme achieves an enhanced image encryption process and a more efficient decryption process without loses of any inherent information of the recovered plain image.

## General Terms
RSA, RNS, Security, Encryption, Decryption

## Keywords
RNS, MRC, Information Security, Encryption, RSA Forward Conversion, Backward Conversion

## 1. INTRODUCTION
The security of information and data (images inclusive) has become a major concern for the past few decades due to the rapid advancement in internet and networking technologies. Images have found usage in diverse areas such as in medicine, military, science, engineering, art, entertainment, advertising, and education. With the increasing use of digital techniques for transmitting and storing images, the fundamental issue of protecting the confidentiality, integrity as well as the authenticity of images has become a major concern. Encryption of images is simply coding or protecting the intelligible element of an image which automatically turns the image to something that cannot be easily recognized or decoded by anyone except the recipient of such images. A lot of image scrambling and encryption techniques have been developed to improve the security level of hidden information [1][2]. Image scrambling techniques generally scramble the pixel location of digital images in such a manner that they become chaotic and Indistinguishable. [3]

There are two types of scrambling methods, one is based on 2D matrix transformation and the other is based on 2D Arnold transformation, [4] Encryption based on 2D Arnold transformation is not suitable for images with different number of rows and columns, it increases the data redundancies which automatically increases the disk size of such images and reduces the quality [4].

This proposed scheme will implement a pixel scrambling algorithm and residue number system (RNS) with moduli sets **{$2^n$-1, $2^n$, $2^n$ +1}** using both forward and reverse conversions) to achieve an enhanced image encryption process and a more efficient decryption process.

## 2. BACKGROUND OF RNS
RNS comprises a set of moduli which are independent of each other. An integer is represented by the residue of each of the modulus and arithmetic operations are based on residues individually. The advantage of using the RNS over the conversional system includes "carry-free" operation, fault tolerance, parallelism and modularity. [5]

These inherent features make RNS to be widely used in Digital Signal Processing (DSP) applications such as digital filtering, convolution, fast Fourier transform and image processing. [4]

Let $\{m_1, m_2, m_{3,\ldots,}m_n\}$ be a set of positive integers all greater 1. $m_i$ is called a modulus, and then n-tuple set $\{m_1,m_2, m_{3,\ldots,}m_n\}$ is called a moduli set. Consider an integer number Y. For each of the modulus in $\{ m_1, m_2, m_{3,\ldots,}m_n \}$, we have $y_i$ = Y mod $m_i$ ,(which will be denoted as $|Y|m_i$ ). Thus the number Y in this system is represented as Y = $(y_1, y_2, y_{3,\ldots,}y_n)$ ,$0 \leq y_i < m_i$ . [6]

Given the moduli set {7,8,9}, the number 150 can be represented in RNS as:

$$y_1 = |Y|_{m1} = |150|_7 = 3$$

$$y_2 = |Y|_{m2} = |150|_8 = 6$$

$$y_3 = |Y|_{m3} = |150|_9 = 6$$

Thus, the RNS representation of 150 is thus; $(3,6,6)_{RHS(7,8,9)}$

To avoid redundancy, the moduli set must be pair wise relatively prime. Thus gcd ($m_i$ ,$m_j$) =1 for i≠j, where gcd means the greatest common divisor of $m_i$ ,$m_j$.

Let $M = \prod_{i=1}^{n} m_i$, then the RNS representation is unique for any integer $Y \in [0, M-1]$ M is called the dynamic range. [7]

A Decimal to Residue (D/R) converter (encoder) is needed in order to convert a decimal number to RNS representation.

# 3. PROPOSED SYSTEM

Our proposed scheme has two (2) levels of encryption; first level is pixel scrambling (scrambling algorithm) and second level of encryption implements forward conversion , while first level decryption is reverse conversion and second level decryption is by pixel unscrambling with moduli sets **{$2^n$-1, $2^n$, $2^n$ +1}**.

## 3.1 Random Number Encryption Algorithm (1st level encryption)

1- Input original image

2- Find the size of the original image (the total number of rows and column)

3- Point to the first pixel in the original image.

4- Let counter equal to 1.

5- Generate new position of the current pixel in the encrypt image by generating two random (n1, n2) numbers, one for row the other for the column.

6- While the new position of encrypt image is generated before go to step 4 otherwise go to step6.

7- Save the value of n1 in the array k1 (counter).

8- Save the value of n2 in the array k2 (counter).

9- The current pixel of the original image will take the position (n1, n2) in encrypt image.

10- While all the pixels of original image finished, go to 13 otherwise go to 10

11- Point the next pixel in the original image. 12- Increment counter by 1

13- Go to step 5

14- End

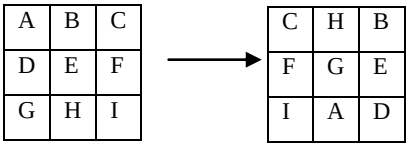


**Fig 1: Scrambling of a 3x 3 matrix image.**

## 3.2 Forward Conversion (2nd level encryption)

Given the moduli set {9,8,7}

$2^n$-1 ,$2^n$ ,$2^n$+1 (n=3)

$m_1$=9 $m_2$=8 $m_3$=7

| | | |
|---|---|---|
| 128 | 108 | 66 |
| 151 | 113 | 179 |
| 139 | 120 | 132 |

**Fig 2: Scrambled 3x3 matrix using the above moduli set.**

## 3.3 Reverse Conversion (1st level decryption)

$Y = \left|\sum_{i=1}^{n} M_i |M_i^{-1} y_i|_{m_i}\right|_M$ **Eq.1**

N=3

$X = |M_1|M_1|_{m_1} x_1 + M_2|M_2|m_2 x_2 + M_3|M_3|m_3 x_3|M$ **Eq.2**

$M_i = \frac{M}{m_i}$ **Eq.3**

M=504

$M_1$= 56 $|M_1^{-1}|m1 = 5$

$M_2$=63 $|M_2^{-1}|m2 = 7$

$M_3$=72 $|M_3^{-1}|m3 = 4$

$X_{11}$=|56X5X2 + 63X7X0 + 72X4X2|$_{504}$

$X_{11}$=128 $X_{12}$=108 $X_{13}$=66

$X_{21}$=151 $X_{22}$=113 $X_{23}$=179

$X_{31}$=139 $X_{32}$=120 $X_{33}$=132

## 3.4 Unscrambling Algorithm (2nd level decryption)

1- Input the encrypt image
2- Input the key k1 & k2
3- Set counter equal to 1
4- Point to the first pixel of the decrypt image
5- n1=k (counter)
6- n2=k2 (counter).
7- Get the value of the position (n1, n2) from the encrypt image and put it in the current position of decrypt image
8- While the counter is not the last position of k1 and k2 go to 9 otherwise go to 13
9- begin
10- Increase counter by one
11- Point to the next pixel of encrypt image
12- Go to 5
13- end

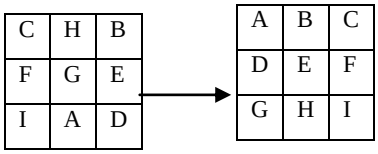


**Fig 3: Unscrambling of a 3x 3 matrix image.**

# 4. PERFORMANCE EVALUATION

The encryption section of our scheme was solely based on two different algorithms, the scrambling algorithm and forward conversion (RNS). The scrambling stage only changes the position of the pixels as seen in Fig 1 and Forward conversation changes the value as seen in Fig 2 which enhances the security of the image.

From Tab 1, our scheme has lower number of elements and lower computational time compared to [4] proposed. Both the encryption and decryption algorithms takes lesser time computational time and lower number of elements compared to [4]. Fig: 4 and 5, (histogram analysis), an hacker cannot easily guess or deduce the encrypted image from the original image. The two layer encryption this scheme offers (i.e Random scrambling, Fig: 1 and Forward Conversion Fig: 2 using Residue Number System) has changed the position of the image pixels and also the value of the pixels which makes it difficult for hackers to guess the original image from the encrypted. The scheme provides a better security than [4]

**Tab 1: Time Complexity for the Proposed Scheme**

| **Image details** | | **Encryption algorithm** | | **Decryption algorithm** | |
|---|---|---|---|---|---|
| **Image** | **No. of elements** | **Scrambling** (seconds) | **Encoding** (seconds) | **Decoding** (seconds) | **Unscrambling** (seconds) |
| Lena(512 x512) | 262144 | 0.1100 | 7.5218 | 9.4065 | 0.1010 |
| Checkerboard(256 x256) | 196608 | 0.2181 | 4.6122 | 4.8484 | 0.2050 |
| Koala(448 x336) | 45184 | 0.1160 | 9.8514 | 11.4281 | 0.1847 |

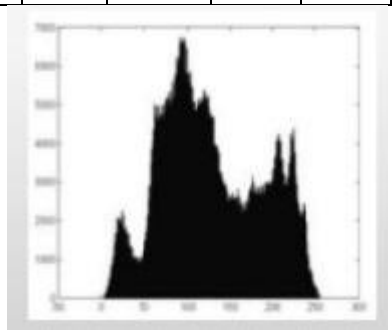

**Fig 4: Histogram of original image**

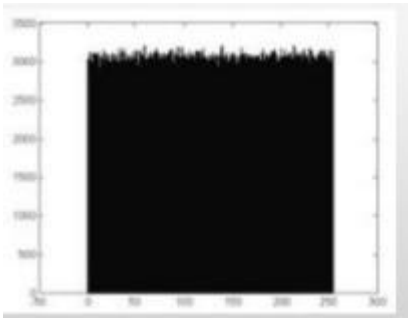

**Fig 5: Histogram of encrypted image**

## 5. CONCLUSION

This paper proposed a simple encryption system to encrypt and decrypt the images based on random number generation and RNS (Forward and Reverse Conversion). The image is first encrypted by changing the position of each pixel in the original image.The original image is then read row by row, pixel by pixel and each pixel will take a new position in the scrambled image. The new position is chosen based on random number generation from the random number generator. The key will be generated as a matrix during the encryption process and also the key saves the position of each pixel in the encrypted/scrambled image. The second layer of encryption then transforms the scrambled image to moduli images (Forward Conversion) by changing the pixel value of the scrambled image as shown in Fig:2 which automatically adds an extra security layer to our data (Image).

## 6. REFERENCES

[1] Wang .D, Chang C, Liu .Y, Song .G, and Liu .L (2015)," Digital Image Scrambling Algorithm Based on Chaoti Sequence and Decomposition and Recombination of Pixel Values" , International Journal of Network Security, vol.17, PP.322-327.

[2] Liping S,Qin, Liu Bo .Z, Jun Q, Huan L (2008)" Image Scrambling Algorithm Based on Random Shuffling Strategy" 3rd IEEE Conference on Industrial Electronics and Applications, pp. 2278 – 2283.

[3] Radu. B, Cristina D, Iustin .P and Cristina F. (2014) "A New Fast Chaos-Based Image Scrambling Algorithm" 10th international conference on communication, pp 1-4.

[4] Alhassan S, Gbolagade K.A (2013)" Enhancement of the Security of a Digital Image using the Moduli Set $2^n$-1 ,$2^n$ ,$2^n$+1 International Journal of Advanced Research in Computer Engineering & Technology , Volume 2, Issue 7.

[5] Gbolagade K.A, Cotofana S.D (2008) "A Residue to Binary Converter for the {2n+2,2n+1,2n} Moduli Set", Asilomar Conference on Signals, Systems, and Computers, pp. 1785-1789, California, USA.

[6] Siewobr H, Gbolagade K.A, and Cotofana S.D (2014) "An Efficient Residue-to-Binary Converter for the New Moduli Set $\{2^{n/2}\pm1,2^{2n+1},2^n+1\}$". International Symposium on Integrated Circuits (ISIC 2014), (to appear), Singapore.

[7] Siewobr H, Gbolagade K.A. (2012) "An Area Efficient RNS-to-Binary Converter for the Moduli Set $\{2^n,2^{2n+1}-1,2^n-1\}$". 4th IEEE International Conference on Adaptive Science and Technology, pp. 104-107, Kumasi, Ghana.

[8]