Kalpa Publications in Computing

Volume 12, 2019, Pages 20–31

Proceedings of 4th International Conference on the Internet, Cyber Security and Information Systems 2019

A Novel Text Encryption and Decryption Scheme using the Genetic Algorithm and Residual Numbers

Peter Awon-natemi Agbedemnab, Edward Yellakuor Baagyere and Mohammed Ibrahim Daabo

Department of Computer Science,
University for Development Studies, Navrongo, Ghana.
pagbedemnab@uds.edu.gh, ybaagyere@uds.edu.gh & mdaabo@uds.edu.gh

Abstract

Data Security is a major concern for both individuals and organisations that are engaged in one form of communication or the other, especially in the cyberspace as a result of the emergence of digital computing and communication. In this paper, we present a novel three-layered text encryption and decryption scheme that is capable of encrypting and decrypting any character or symbol using Genetic Algorithm (GA) and some inherent properties from the Residue Number System (RNS). Simulated results of the proposed scheme shows that it is chaotic by sense of sight, robust with a very wide key space composed at different stages of the scheme and has a good throughput rate as well as being able to encrypt both smaller and larger messages.

Keywords: data *security, encryption, decryption, genetic algorithm, residue number system, cyberspace.*

1 Introduction

Data security involves techniques of communicating in a secret manner in such a way that valuable information is kept sacrosanct from unauthorized users (Phaneendra, 2014; Stinson, 1995). In order to ensure this secret communication, many cryptographic schemes have been developed over the years with different number of keys, and levels of encryption and decryption. The sole objective of such schemes is to make it difficult for untended persons on the line of communication to have access to the information or message and/or if they ever did, then it should be very difficult for such persons to decode the content of the transmitted message. However, the degree of security offered by a cryptographic scheme depends to a larger extent on the type and length of the keys utilised, the levels of encryption to create chaos, the throughput rate of the algorithms as well as the ability of such encryption algorithms to encrypt smaller messages, (Stinson, 1995). Encryption involves translating the

original message (usually in plaintext) into a cypher text using an encryption key. The reverse process of getting back the plaintext (message) from the cypher text (also with a decryption key) is termed decryption, (Abdul-mumin & Gbolagade, 2018; Pakshwar, Trivedi, & Richhariya, 2013).

There are three major types of cryptography namely symmetric, asymmetric and hash functions. Symmetric cryptography uses a single key for both encryption and decryption. The asymmetric cryptography uses two keys; one for encryption (known as the public key), and the other for decryption (known as private key). The final class, Hash Functions use no key but compute a fixed-length mathematical irreversible hash value based on the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered (Kessler, 2018; Kumar & Ghose, 2009). Popular amongst the cryptographic schemes are the Data Encryption Standards (DES) and Advance Encryption Standards (AES) for single key cryptography and the Rivest, Shamir and Adlemnan (RSA) for double key cryptography. However, these schemes have been susceptible to cryptographic attacks such as brute force attacks in the case of DES and AES, and computational attacks on the RSA, (Abdul-mumin & Gbolagade, 2018; Almarimi, Kumar, Almerhag, & Elzoghbi, 2014).

The Genetic Algorithm (GA) is an evolutionary search algorithm that mimics the mechanics of natural genetics, (Goldberg, 1989; Wu & Rulkov, 1993) by enhancing and diversifying the characteristics of individuals in one or more populations into new and improved populations based on desired characteristics . The Residue Number System (RNS) on the other hand, belongs to the family of non-weighted number systems that decomposes larger integers into smaller residues (remainders) uniquely to represent numbers. This system can be applied in areas such as cryptography, Digital Signal Processing systems and any computational intensive systems.

In this paper, a new text encryption and decryption scheme is proposed using the operators of GA (selection, crossover and mutation) at different levels of encryption and decryption, and the inherent features of RNS such as residues and parallelism in a three-layered scheme to encrypt and decrypt text data. The rest of the paper is organized as follows: A background information relating to key concepts of GA and RNS is presented in Section 2 with a review of related existing schemes. Section 3 presents the algorithm and cryptosystem of the proposed scheme. In Section 4, a hardware implementation and realisation is presented with test simulations. The performance of the proposed scheme is evaluated in Section 5, whiles the paper is concluded in Section 6.

2 Background Information and Related Works

The set of GA operators are selection, crossover, and mutation operations. These operators together makes GA a powerful search algorithm. It begins with the selection of fit chromosomes from a population of chromosomes and then goes through mating and recombination of parent chromosomes to result into fitter offspring chromosomes. The operators – crossover and mutation particularly make GA very suitable for data encryption, (Kumar & Ghose, 2009).

The RNS possesses inherently desirable properties such as parallel computation, and carry free arithmetic, which are the major operations in fields like digital signal processing, cryptography, digital communication and image processing, (Agbedemnab & Bankas, 2015; Daabo & Gbolagade, 2012; Omondi & Premkumar, 2007; Sousa, 2015). The RNS is capable of enhancing schemes in these applications by providing fewer hardware resources, improved delays and power profiles in devices that run on batteries. The decomposition of numbers into residues by sight looks chaotic and thus makes the RNS suitable for encryption purposes. RNS is defined by a set of relatively prime moduli $\{m_1, m_2, \dots, m_n\}$ such that the $\gcd(m_i, m_j) = 1$ for $i \neq j$, and gcd means the greatest common divisor of m_i and m_j; and $M = \prod_{i=1}^{n} m_i$, is the Dynamic Range (DR). The residues of a decimal number X can be obtained as $x_i = |X|_{m_i}$, thus X can be represented in RNS as $X = (x_1, x_2, \dots, x_n)$, $0 \leq x_i \leq m_i$ which representation is unique for any integer $X \in [0, M-1]$, this is referred to as *Forward*

Conversion; and arithmetic operations such as addition, subtraction and multiplication are performed totally in parallel on the independent residues, (Younes & Steffan, 2013). *Reverse Conversion* on the other hand, is converting back from RNS notation into conventional notation but is a more complex operation. This has been computed generally, by two techniques: Chinese Remainder Theorem (CRT) and Mixed Radix Conversion (MRC) and their modified variants in the new CRTs I-III, Core function and Modular Weighed Sum Method, (Omondi & Premkumar, 2007).

The CRT is computed as:

$$X = \left| \sum_{i=1}^{n} \ell_i \, |k_i x_i|_{m_i} \right|_M \tag{1}$$

where,

$$M = \prod_{i=1}^{n} m_i \; ; \quad \ell_i = \frac{M}{m_i} \; ; \; |k_i \times \ell_i|_{m_i} = 1$$

Whereas, the CRT is computed using mod-M, the MRC uses mod-m_i for computation and involves sequential operations, which can limit speed (Gbolagade, Chaves, Sousa, & Cotofana, 2010).

Many data encryption and decryption schemes have been proposed by various researchers to ensure the security, integrity, and confidentiality of data by either employing the GA, or a combination of the GA with the RNS, or the RNS with other techniques. In (Kumar & Ghose, 2009), an approach for transmitting secured data using GA with pseudorandom sequence was proposed, to attain a high feasibility for integration into commercial multimedia transmission applications. A scheme proposed by (Jassim & Ali, 2013) utilised genetic algorithm to establish an approach for random key to each letter in text message encoding and decoding by using their ASCII code equivalents as a first step, followed by a random generation of keys for each code according to the cardinality of the ASCII code in binary form. The scheme in (Srikanth, Mehta, Yadav, Singh, & Singhal, 2017) employed genetic algorithm operations such as crossover and mutation functions, binary conversion with pseudorandom functions to encrypt and decrypt data. A power residue generator, $X_{i+1} = X_i.a(mod\ m)$ was used and the value of m was chosen to be as large as possible. These works proposed techniques for encrypting data particularly text, by using either only genetic algorithm (in the case of (Jassim & Ali, 2013)) or a combination of genetic algorithm and pseudorandom number sequence as presented in (Kumar & Ghose, 2009) and (Srikanth et al., 2017). The presence of large modulus m, in the adoption of pseudorandom numbers results in complex schemes which ultimately affects the throughput rate. Decoding of smaller numbers (messages) can also lead to redundancies, which is addressed when RNS, which possesses inherently error correcting capabilities, is employed, (Abdul-mumin, 2016). Other schemes, such as (Navin, Oskuei, Khashandarag, & Mirnia, 2012) employed RNS for encryption, and Huffman coding and Lempel-Ziv-Welch (LZW) compression algorithm used to compress the information. It also employed DES algorithm for high security. In (Salifu, 2017), an RNS Based Fault Tolerant Schemes for Enhancing RSA Encryption Scheme was proposed, by employing a two-layered encryption and decryption approach. Here also, RNS is used with other algorithms and techniques such as Huffman coding, LZW compression algorithm and DES in (Navin et al., 2012); or with the classical RSA, (Salifu, 2017) to encrypt and decrypt data. All these mentioned techniques offer single layer of confusion in addition to the RNS layer. But the degree of any cryptographic scheme depends greatly on the length or stages of confusion that any attacker will have to go through, and this can be achieved when GA is combined with RNS. Also, by the Moore's Law, (Lenstra & Citibank, 2001), computing power increases linearly every 1.5 years and thus hitherto powerful cryptosystems can easily be broken with the use of powerful computing devices, therefore the need for the continuous research into

developing novel multi-layered cryptosystem is necessary to resist security breaches of these systems. Quantum computers are also concern to modern cryptosystems. This paper therefore, attempts to address this problem by proposing a three-layered text encryption and decryption scheme in the subsequent section.

3 Proposed Scheme

We propose a new text encryption and decryption scheme called GARN (Genetic Algorithm Residue Numbers), which combines the operators of GA and residues numbers to encrypt and decrypt different kinds of text in a three-layered manner (i.e. RNS data conversion, Crossover and Mutation).

3.1 Algorithm for the Proposed Scheme

Algorithm 1 is the pseudocode for the proposed GARN scheme. In the algorithm, **GenModulus(1^n)** generates the keys for the scheme, **Encsk** and **Decsk** denote the levels of encryption and decryption respectively. Also, C_{ikey} and ρ_{ikey} refer to the crossover and mutation keys respectively as C_i and ρ_i denote crossover and mutation.

Algorithm 1: Proposed Algorithm

```
        // Encryption Process
```

Input: Get ASCII codes or Unicode of all characters from inputted text
Result: GenModulu(1^n) runs GARN(1^n) to obtain m_i, C_{ikey} and ρ_{ikey}, $i = 1, 2, \dots, N$
Input: (Encsk(X))$_1$: m_i, choose $n \in M$; M is the dynamic range, and compute
$x_i = X \bmod m_i$: $\varphi_i = [x_1 x_2 \dots x_N]$
Input: (Encsk(φ_i))$_2$: φ_i,
while $i \leq N_{Blocks}$ **Do**
XOR LSB of φ_i and φ_{i+1}
if $N_{Blocks}\ mod\ 2 \neq 0$ **then**
Append zero chromosome
end
Output: 0 → Two-point Crossover
1 → Multi-point Crossover
end
(Encsk(C_i))$_3$: Mutate C_i using ρ_{ikey} to ρ_i
Result: Store ρ_i as cipher text

```
        // Decryption Process
```

Input: (Decsk(ρ_i))$_1$: ρ_{ikey}; mutate $\rho_i$to get C_i
(Decsk(C_i))$_2$: **while** $i \leq N_{Blocks}$ **Do**
XOR LSB of C_i and C_{i+1} to determine type of crossover
end
Result: $\varphi_i = [x_1, x_2, \dots, x_N]$, ASCII/Unicode character values in RNS representation
Output: (Decsk(C_i))$_3$: m_i, and compute X using CRT

3.2 Proposed Cryptosystem

For all $X_i \in M \in \mathbb{Z}^+$ choose $n \in \mathbb{Z}^+ \ni M = \prod_{i=1}^{N} m_i$ is large enough to contain X_i with a key space $\mathcal{K} = \{m_i, C_{ikey}, \rho_{ikey}\}$ as follows:

$\mathcal{P} = \{X_i \in M\}$
$\mathcal{C} = \{\rho_i \in \mathbb{Z}^+\}$
$\mathcal{E}_1(X)$: $X \equiv x_i \bmod m_i, \varphi_i$
$\mathcal{E}_2(X)$: Crossover, C_i
$\mathcal{E}_3(X)$: $\rho_i = Mutation$
$\mathcal{D}_1(X)$: $C_i = Mutation$
$\mathcal{D}_2(X)$: Crossover, φ_i
$\mathcal{D}_3(X)$: R-B Conversion

Where $\mathcal{P}$ and $\mathcal{C}$ are the ASCII/Unicode values of the plain and cipher texts respectively. Also, $\mathcal{E}$ and $\mathcal{D}$ denote the encryption and decryption rules respectively. $\mathcal{K}$ is the set containing the keys at the different stages of the encryption or decryption processes. Figure 1 is the structure of the proposed cryptosystem; the front and back ends of the system use data conversions in RNS. Whilst the Crossover and Mutation operations come after the forward conversion during encryption, these operations take place before the reverse conversion during decryption.

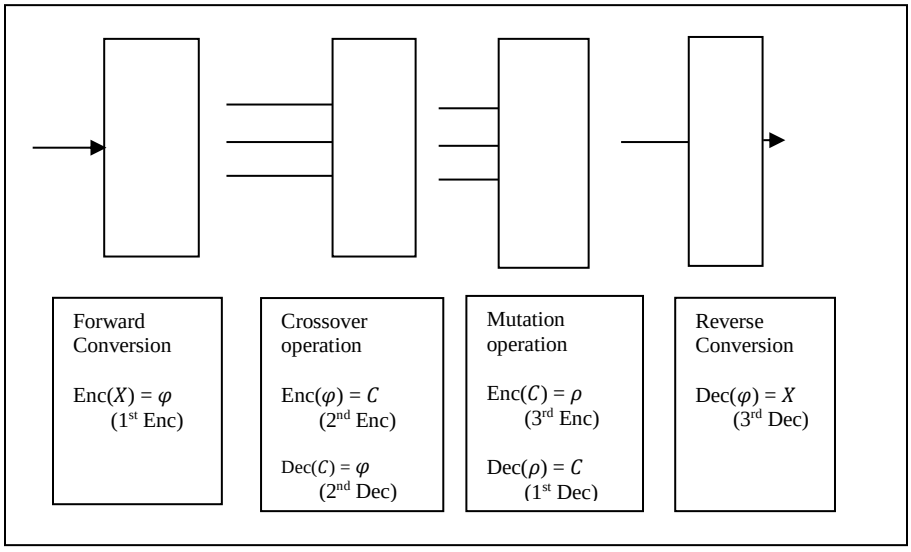

Figure 1: Structure of Proposed Cryptosystem

4 Implementation of the Proposed Scheme

In this paper, the moduli set $\{2^{n-1} - 1, 2^n - 1, 2^n\}$, is used for the implementation of the proposed scheme. This moduli set is well-balanced with efficient properties such as non-derailment of the various channels in order not to impose unnecessary delay on schemes. It is therefore, suitable for the Proposed Scheme as it has the capacity of enhancing the throughput rate.

For the chosen moduli set, any binary number X, which is $(3n - 1)$-bits wide can be partitioned into three sub-blocks for easy implementation as:

$$\underbrace{X_{3n-2}X_{3n-3}\dots X_{2n}}_{B_3-(n-1)\ bits} \mid \underbrace{X_{2n-1}X_{2n-2}\dots X_n}_{B_2-n\ bits} \mid \underbrace{X_{n-1}X_{n-2}\dots X_0}_{B_1-n\ bits} \quad (2)$$

Which can be computed as,

$$X = B_1 + 2^n B_2 + 2^{2n} B_3 \tag{3}$$

Such that,

$$x_1 = |X|_{2^{n-1}-1} = ||B_1|_{2^{n-1}-1} + |2^n B_2|_{2^{n-1}-1} + |2^{2n} B_3|_{2^{n-1}-1}|_{2^{n-1}-1} = |B_1 + B_2 + B_3|_{2^{n-1}-1} \tag{4}$$

$$x_2 = |X|_{2^n-1} = ||B_1|_{2^n-1} + |2^n B_2|_{2^n-1} + |2^{2n} B_3|_{2^n-1}|_{2^n-1} = |B_1 + B_2 + B_3|_{2^n-1} \tag{5}$$

$$x_3 = |X|_{2^n} \quad = B_1 \tag{6}$$

The schematic diagram for the forward converter is shown in Figure 2. The third residue, x_3, needs no computation since it is equal to the value of the first block B_1 after splitting the decimal number X.

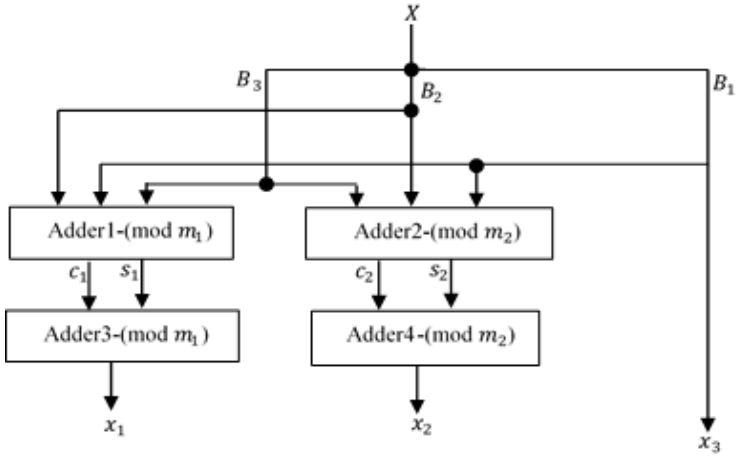

Figure 2: Block diagram of Forward Converter for the Proposed Scheme

The computations of x_1 and x_2 are performed at the same time first by using Adders 1 and 2 respectively, which respective residues and carries computed with Adders 3 and 4. Adders 1 and 2 are Carry Save Adders whilst Adders 4 and 5 are Carry Propagate Adders.

φ_i is a concatenation of the residues as $x_1 x_2 x_3$ for each respective character in the set $\mathcal{P} = \{X_i \in M\}$ and successively paired for a random crossover operation, C_i after XORing the LSBs to determine the crossover rule:

$$0 \rightarrow Two - point\ Crossover$$

$$1 \rightarrow Multi - point\ Crossover$$

After the crossover operations, each chromosome is mutated by a simple n-bits circular left shift of C_i to get ρ_i which is stored as the cipher text.

The decryption process involves a reversal of the mutation, followed by the crossover operation and then RNS-Binary conversion. To reverse convert an RNS number with the given moduli set, a simplified CRT is achieved through mathematical manipulation:

Theorem 1: *For the moduli set* $\{2^{n-1} - 1, 2^n - 1, 2^n\}$, *the following holds true from Equation (1):*

$$|k_1|_{m_1} = 2^{n-2} \tag{7}$$

$$|k_2|_{m_3} = -2 \tag{8}$$

$$|k_3|_{m_3} = 2^{n-1} + 1 \tag{9}$$

Proof: If it can be demonstrated that $|\ell_i \times k_i|_{m_i} = 1$, then $|k_i|_{m_i}$ is the multiplicative inverse of ℓ_i mod m_i, for $i = 1,2,3$.

Thus, for Equation (7), $|2^n(2^n-1) \times (2^{n-2})|_{2^{n-1}-1} = |2^{n-1}(2^{n-1}) \times (2^{n-1})|_{2^{n-1}-1}$
$= |(1)(1)(1)|_{2^{n-1}-1} = 1$

Also, for (8) $|2^n(2^{n-1}-1) \times (-2)|_{2^n-1} = |(-1)(2^n-2)|_{2^n-1}$
$= |(-1)(-1)|_{2^n-1} = 1$

Lastly for (9), $|(2^n-1)(2^{n-1}-1) \times (2^{n-1}+1)|_{2^n} = |(-1)(-1)(1)|_{2^n}$
$= 1$ ■

Hence Equations (7) to (9) hold true since:

$$\left.\begin{aligned} \ell_1 &= 2^n(2^n-1) \\ \ell_2 &= 2^n(2^{n-1}-1) \\ \ell_3 &= (2^n-1)(2^{n-1}-1) \end{aligned}\right\} \quad (10)$$

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

$$X = 2^n \mathfrak{Z} + x_3 \quad (11)$$

where,

$$\mathfrak{Z} = |Ax_1 - Bx_2 + Ax_3 + x_3|_{m_1} \quad (12)$$

and

$$A = 2^{n-2}(2^n-1), \quad B = 2(2^{n-1}-1)$$

Proof: Substituting Equations (7) – (9), (10) in Equation (1), factorizing and taking the floor function we obtain Equation (11). ■

Now, to implement Equation (11), we simplify further as:

$$X = \mathfrak{Z}_{2n-2}\mathfrak{Z}_{2n-3} \dots \mathfrak{Z}_1\mathfrak{Z}_0 \overbrace{00 \dots 00}^{n-bits} + x_{3,n-1}x_{3,n-2} \dots x_{3,1}x_{3,0} \quad (13)$$

Since x_3 is an n-bit number, it can easily be joined to $\mathfrak{Z}$ to occupy the n-bit positions of zeros by a process called concatenation without necessarily requiring any hardware as:

$$\begin{aligned} X &= \mathfrak{Z}_{2n-2}\mathfrak{Z}_{2n-3} \dots \mathfrak{Z}_1\mathfrak{Z}_0 \overbrace{00 \dots 00}^{n-bits} \bowtie x_{3,n-1}x_{3,n-2} \dots x_{3,1}x_{3,0} \\ &= \mathfrak{Z}_{2n-2}\mathfrak{Z}_{2n-3} \dots \mathfrak{Z}_1\mathfrak{Z}_0 x_{3,n-1}x_{3,n-2} \dots x_{3,1}x_{3,0} \\ &= X_{3n-2}X_{3n-3} \dots X_1X_0 \end{aligned} \quad (14)$$

where,

$$\begin{aligned} \mathfrak{Z} &= \left|\left|(2^{n-2})(2^n-1)x_1\right|_{\ell_3} + \left|(-2)(2^{n-1}-1)x_2\right|_{\ell_3} + \left|(2^{2n-2}-2^{n-2}-1)x_3\right|_{\ell_3}\right|_{\ell_3} \\ &= \left|\left|Ax_1\right|_{\ell_3} - \left|Bx_2\right|_{\ell_3} + \left|Ax_3\right|_{\ell_3} - x_3\right|_{\ell_3} \\ &= \left|\left|A\left(x_{1,n-2} \dots x_{1,0}\right)\right|_{\ell_3} + \left|\bar{B}\left(\bar{x}_{2,n-1} \dots \bar{x}_{2,0}\right)\right|_{\ell_3} + \left|A\left(x_{3,n-1} \dots x_{3,0}\right)\right|_{\ell_3} + \left(\bar{x}_{3,n-1} \dots \bar{x}_{3,0}\right)\right|_{\ell_3} \end{aligned} \quad (15)$$

and,

$$A = \underbrace{111 \dots 11}_{n-bits}\ \underbrace{00 \dots 0}_{(n-2)-bits} \text{ and } B = \underbrace{111 \dots 11}_{(n-1)-bits}0$$

This requires three fast modulo adders, two of which, will be Carry Save adders (CSAs) and the other, Carry Propagate Adder (CPA) as depicted in Figure 3.

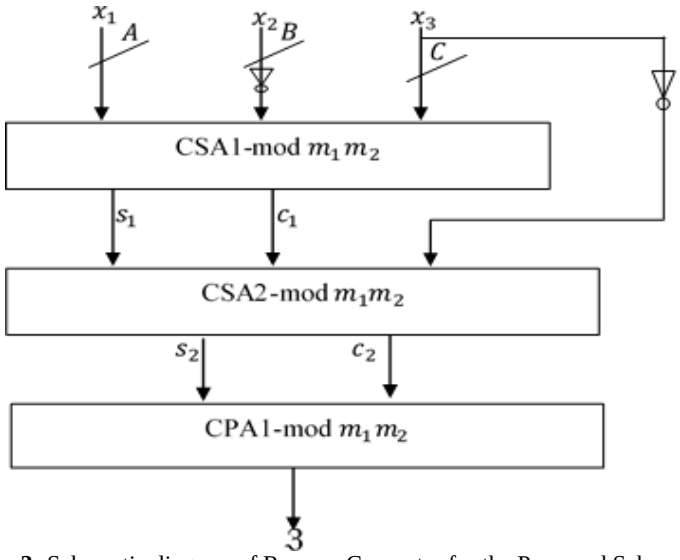

Figure 3: Schematic diagram of Reverse Converter for the Proposed Scheme

Since the values of A and B are known, it will be hardwired to append to the residues as in equation (15) through a $(2n-1)$-bit modulo carry save adder (CSA1). The outputs from this adder will intend be added with x_3 using another $(2n-1)$-bit modulo carry save adder (CSA2). In order to get the value of $\mathfrak{Z}$, a carry propagate adder (CPA1) is used, which is also $(2n-1)$-bit wide and a modulo adder. The binary number X is the result of a concatenation in equation (14) which does not require any hardware resources but append an extra n-bits, thereby increasing the bit length to $(3n-1)$-bits wide.

4.1 Simulation

The Proposed Scheme was tested on different types of files containing letters, numbers and symbol characters. The simulation was done using MatLab® R2017b and a CORE™ i7 microprocessor laptop. The simulated results are shown in Table 1 with their average times of execution.

S/N	***File Name***	***Original Text***	***Encrypted Text***	***Decrypted Text***	***Average Runtime***
1.	A.txt	This is a new text encryption technique using genetic algorithm and residue numbers	? "i0 O04ñ? T°°? ° qÅb? ¢dU? °? E2q? ±% °'é? qM¨ï±ñ ? 0? E0? "ïäb ? ´? u ? â i"® ° õ? ô â k	This is a new text encryption technique using genetic algorithm and residue numbers	**0.0472s**
2.	B. txt	we can also encrypt symbols = *&^%$#@!()_+-/?><}{][|~	ò±0E? u4#ËF´? ñAb(? k* ôÂ §O0´ÂQë? !Ð- - té¡÷Äò¤? æ È1 À	we can also encrypt symbols = *&^%$#@!()_+-/?><}{][|~	**0.0394s**

3.	C. txt	170003467 70087645 180890011 180000334	ªl▯ ▯ "F? ÇN´ªl▯ ▯ ´JÇ? ▯ ´▯ 0▯ ▯ ▯ "" 0▯ 0▯ ▯ ▯ ▯ bb?	170003467 70087645 180890011 180000334	**0.0384s**

Table 1: Simulated Results

5 Performance Evaluation

The encrypted results from Table 1 look very chaotic by sense of sight and do not give any clue whatsoever to the original files. Regarding the robustness of the proposed scheme, an attacker has a $+\infty$ times of guessing the value of n, after which he has $2^{M!} \times N$ (N is number of moduli) chance of getting the moduli set used, $N!$ for the order before the modulo operations to get the residues, (Abdallah & Skavantzos, 1995). The attacker has to continue to obtain the crossover and mutation keys in order to have a complete breakthrough of the proposed scheme at different stages. Thus, for an attacker, without the knowledge of the parameters of the algorithm, it would be an NP-Hard problem which (through a formal proof can be shown) is unfeasible.

The computational complexity of the proposed scheme is defined by the hardware resources that it requires: the Forward Converter (F/C) requires four modulo adders – two $(n-1)$-bits and two n-bits adders; the crossover operation requires just a XOR unit; the mutation operation requires no hardware unit; and the Reverse Converter (R/C) requires three modulo adders of length $(2n-1)$-bits as shown in Table 2. Thus, whiles the scheme is robust at different layers, it requires few hardware units thereby making it computationally efficient. This is corroborated by the times of execution as measured in Table 3.

S/N	***Component***	***No. of Hardware Units***	***Area***-(Δ_{FA})	***Delay***-(Δ_{FA})
1.	Forward Converter (F/C)	4 Full Adders	$4n-2$	$2n$
2.	Crossover	1 XOR	1	1
3.	Mutation	Nil	Nil	Nil
4.	Reverse Converter (R/C)	3 Full Adders	$6n-3$	$2n$
Total		***8 units***	$\mathbf{10n-4}$	$\mathbf{4n+1}$

Table 2: Hardware Requirements of the Proposed Scheme

The runtime performance of the scheme was compared to similar existing schemes by (Srikanth et al., 2017) and (Salifu, 2017); this because, the scheme by (Srikanth et al., 2017) employed GA with some other techniques whilst that by (Salifu, 2017) used the RNS and the CRT techique as in this paper. However, the results showed that the proposed scheme performs favourably better. The runtimes of three different files are shown in Table 3:

File	***Scheme***	***Average Runtime***
A.txt	Srikanth *et al.*, (2017)	**0.0415s**
	Salifu, (2017)	**0.2731s**
	Proposed	**0.0472s**
B.txt	Srikanth *et al.*, (2017)	**0.0284s**
	Salifu, (2017)	**0.3012s**
	Proposed	**0.0394s**
C.txt	Srikanth *et al.*, (2017)	**0.0312s**
	Salifu, (2017)	**0.2913s**
	Proposed	**0.0384s**

Table 3: Evaluation of Runtime by a Comparison of other existing Schemes

As observed from Figure 4 the scheme by (Salifu, 2017) has the highest runtime due to the RSA component of the scheme. The scheme by (Srikanth et al., 2017) has the lowest runtime as seen in both Table 3 and Figure 4; however, it does not achieve optimum results when the message to be encrypted is small. It also has only two layers of encryption. Even though, the proposed scheme has three layers of encryption and decryption, its runtime rises minimally above the scheme by (Srikanth et al., 2017) but outperforms the scheme by (Salifu, 2017). Thus, the throughput rate of the proposed scheme competes favourably when compared to similar existing schemes.

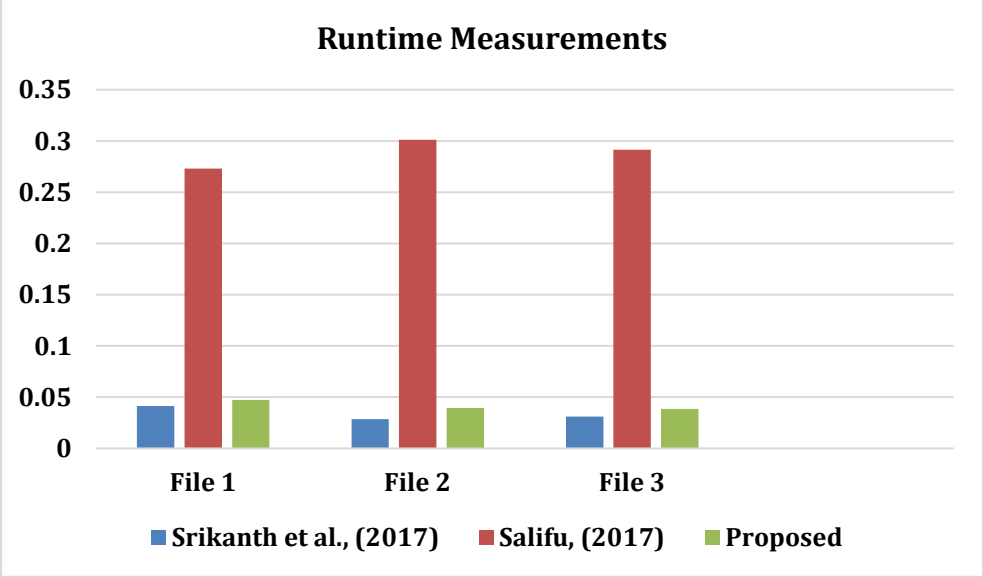

Figure 4: Bar chart showing runtimes of compared schemes

6 Conclusion

Security is a very topical issue currently in digital computing. In this paper, a new text encryption and decryption technique using GA and Residue Numbers (GARN) was presented with its hardware realization and simulation. The simulated results show that the proposed scheme is very chaotic and robust and the throughput rate of the scheme (runtime) competes very favourably with existing similar schemes. In future, this work will be scaled up to include other multimedia elements such as images, videos and audio which are predominantly being used on the Internet.

References

Abdallah, M., & Skavantzos, A. (1995). A systematic approach for selecting practical moduli sets for residue number systems. *Proceedings of the 27th Southeastern Symposium on System Theory, SSST 1995*, 445–449. https://doi.org/10.1109/SSST.1995.390542

Abdul-mumin, S. (2016). Mixed Radix Conversion based RSA Encryption System. *International*

Journal of Computer Applications (0975 – 8887), *150*(1), 43–47.

Abdul-mumin, S., & Gbolagade, K. A. (2018). Rivest Shamir Adleman Encryption Scheme Based on the Chinese Remainder Theorem. *Advances in Networks*, *SciencePG*, *6*(1), 40–47. https://doi.org/10.11648/j.net.20180601.14

Agbedemnab, P. A., & Bankas, E. K. (2015). A Novel RNS Overflow Detection and Correction Algorithm for the Moduli Set {2^n-1,2^n,2^n+1}. *International Journal of Computer Applications*, *110*(16), 30–34. https://doi.org/10.5120/19403-0925

Almarimi, A., Kumar, A., Almerhag, I., & Elzoghbi, N. (2014). a New Approach for Data Encryption, (August).

Daabo, M. I., & Gbolagade, K. A. (2012). RNS Overflow Detection Scheme for the Moduli set {M − 1, M}. *Journal of Computing*, *4*(8), 39–44.

Gbolagade, K. A., Chaves, R., Sousa, L., & Cotofana, S. D. (2010). An improved reverse converter for 2^(2n+1)-1,2^n,2^n-1 moduli set. *IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, 2103–2106.

Goldberg, D. E. (1989). *Genetic algorithm in search optimization and machine learning.* Boston: Addison-Wesley Publishing Company, Inc.

Jassim, A., & Ali, M. (2013). Randomly Encryption Using Genetic Algorithm. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, *2*(8), 242–246.

Kessler, G. C. (2018). *An Overview of Cryptography*.

Kumar, A., & Ghose, M. K. (2009). Overview of information security using genetic algorithm and chaos. *Information Security Journal*, *18*(6), 306–315. https://doi.org/10.1080/19393550903327558

Lenstra, A. K., & Citibank, N. A. (2001). Selecting Cryptographic Key Sizes. *Journal of Cryptology*, *14*, 255–293. https://doi.org/10.1007/s00145-001-0009-4

Navin, A. H., Oskuei, A. R., Khashandarag, A. S., & Mirnia, M. (2012). A Novel Approach Cryptography by using Residue Number System. In *6th International Conference on Computer Sciences and Convergence Information Technology* (pp. 636–639).

Omondi, A., & Premkumar, B. (2007). *Residue Number Systems: Theory and Implementation*. *Advances in Computer Science and Engineering: Texts* (Vol. 2). Published By Imperial College Press and Distributed by World Scientific Publishing Co. Retrieved from http://www.worldscientific.com/worldscibooks/10.1142/p523

Pakshwar, R., Trivedi, V. K., & Richhariya, V. (2013). A Survey On Different Image Encryption and Decryption Techniques. *(IJCSIT) International Journal of Computer Science and Information Technologies*, *4*(1), 113–116.

Phaneendra, H. D. (2014). Identity-Based Cryptography and Comparison with traditional Public key Encryption : A Survey. *International Journal of Computer Science and Information Technologies*, *5*(4), 5521–5525.

Salifu, A.-M. (2017). *Residue Number System Based Fault TolerantT Scheme for enhancing Rivest Shamir Adleman Encryption Scheme*. Kwara State University, Malete.

Sousa, L. (2015). 2^n RNS Scalers for Extended 4-Moduli Sets. *IEEE Transactions on Computers*, *PP*(99), 1–14. https://doi.org/10.1109/TC.2015.2401026

Srikanth, P., Mehta, A., Yadav, N., Singh, S., & Singhal, S. (2017). Encryption and Decryption Using Genetic Algorithm Operations and Pseudorandom Number. *Computer Science and Network*, *6*(3), 455–459.

Stinson, D. (1995). *Cryptography: Theory and Practice*. CRC Press LLC.

Wu, C. W., & Rulkov, N. F. (1993). Studying chaos via 1-Dmaps - a tutorial. In *IEEE Transaction on Circuits and Systems-I: Fundamental Theory and Applications*, (pp. 707–721).

Younes, D., & Steffan, P. (2013). Universal Approaches for Overflow and Sign Detection in Residue Number System Based on {2n – 1, 2n, 2n + 1}. In *ICONS 2013, The Eighth International Conference on Systems* (pp. 77–81). Retrieved from

http://www.thinkmind.org/index.php?view=article&articleid=icons_2013_4_20_20091