Asian Journal of Research in Computer Science

4(4): 1-14, 2019; Article no.AJRCOS.53646

ISSN: 2581-8260

Single and Multiple Error Detection and Correction using Redundant Residue Number System for Cryptographic and Stenographic Schemes

Peter Awon-natemi Agbedemnab[1*] Edward Yellakuor Baagyere[1] and Mohammed Ibrahim Daabo[1]

[1] *Department of Computer Science, University for Development Studies, Navrongo, Ghana.*

Authors' contributions

This work was carried out in collaboration among all authors. Author PAA designed the study, did the preliminary analysis and wrote the first draft of the manuscript. Author EYB re-organized the draft and did further analysis. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2019/v4i430123

Editor(s):

(1) Dr. Hasibun Naher, Associate Professor, Department of Mathematics and Natural Sciences, BRAC University, Dhaka, Bangladesh.

Reviewers:

(1) David Lizcano, Madrid Open University, Spain.

(2) Irshad Ullah, Pakistan.

(3) Sheeraz Ahmed, Iqra National University, Pakistan.

Complete Peer review History: http://www.sdiarticle4.com/review-history/53646

Received: 03 November 2019

Accepted: 09 January 2020

Published: 18 January 2020

Original Research Article

ABSTRACT

The possibility of errors being propagated during the encoding process of cryptographic and steganographic schemes is real due to the introduction of noise by ciphering the data from stage to stage. This real possibility therefore requires that an efficient scheme is proposed such that if after the decoding process the accurate information is not discovered, then it can be employed to detect and correct any errors in the system. The Residue Number System (RNS) by its nature is fault tolerant since an error in one digit position does not affect other digit positions; but the Redundant Residue Number System (RRNS) had been used over the years to effectively detect and correct errors. In this paper, we propose an efficient scheme that can detect and correct both single and multiple errors after and/or during computation and/or transmission provided the redundant moduli are sufficient enough. A theoretical analysis of the performance of the proposed scheme show

**Corresponding author: E-mail: yxiahu@163.com; yxiahu@ncepu.edu.cn;*

it will be a better choice for detecting and correcting computational and transmission errors to existing similar state-of-the-art schemes.

Keywords: Encoding; decoding; steganography; cryptography; residue number system (RNS); redundant residue number system (RRNS).

1 INTRODUCTION

Residue Number System (RNS) belongs to the family of unconventional number systems where numbers specifically, positive integers are represented as vector of remainders based on a defined set of moduli. Mathematically, RNS is defined by a set of relatively prime moduli $\{m_1, m_2, ..., m_n\}$ such that the $\gcd(m_i, m_j) = 1$ for $i \neq j$, and $\gcd$ means greatest common divisor of m_i and m_j; and $M = \prod_{i=1}^{n} m_i$, is the Dynamic Range (DR), [1, 2, 3]. The residues of a number X in conventional representation such as binary or decimal can be obtained as $x_i = |X|_{m_i}$, thus X can be represented in RNS as $X = (x_1, x_2, , x_n)$, $0 \leq x_i \leq m_i$, this representation ought to be unique for any integer $X \in [0, M-1]$, [4, 5]. The modular arithmetic here is independent on the respective moduli, therefore, this number system is capable of supporting parallel arithmetic, as well as carry-free and high speed arithmetic, [6, 7, 8]. This number system also inherently possesses some useful properties such as parallelism, modularity, fault tolerance, and carry-free operations [9, 10]. It is very efficient in performing arithmetic operations such as additions, subtractions and multiplications that predominate in digital signal processing, cryptographic and digital communication systems.

Conversion of data to and fro the RNS is classified into forward and reverse conversions. The *forward conversion* involves converting a binary or decimal number into its RNS equivalent while converting the RNS number back into binary or decimal is *reverse conversion*[10]. Relatively, reverse conversion is more complex. A general structure of a typical RNS processor [11, 12], is shown in Figure .

In Figure , data sets in the form of binary or decimal are forward-converted using a forward converter with a set of moduli sets as its processing units into residues. The residues is converted back into binary or decimal through reverse conversion with a reverse converter.

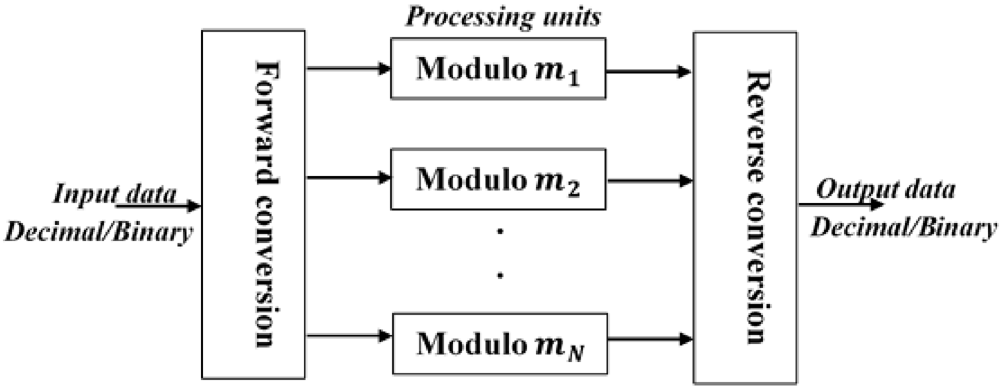

Fig. 1. General structure of an RNS-based processor

Two techniques that have been generally used over the years to perform the reverse conversion process are: Chinese Remainder Theorem (CRT) and Mixed Radix Conversion (MRC); these techniques have recently been modified into other variants such as CRT-I, CRT-II, CRT-III, Core function and

Modular Weighed Sum Method, [10, 13].
The CRT is computed as:

$$X = \left| \sum_{i=1}^{n} \ell_i \left| k_i x_i \right|_{m_i} \right|_M \tag{1.1}$$

where,

$$M = \prod_{i=1}^{N} m_i; \qquad \ell_i = \frac{M}{m_i}; \qquad \left| k_i \times \ell \right|_{m_i} = 1$$

and the MRC as:

$$X = \sum_{i=2}^{n} \vartheta_i \prod_{j=1}^{i-1} m_j + \vartheta_i \tag{1.2}$$

where $\vartheta_i,\ i = 1, 2, \cdots, n$ are the Mixed Radix Digits (MRDS) and computed as follows:

$$\begin{aligned}
\vartheta_1 &= x_1 \\
\vartheta_2 &= \left| (x_2 - \vartheta_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \\
\vartheta_3 &= \left| \Big((x_3 - \vartheta_1) \left| m_1^{-1} \right|_{m_3} - \vartheta_2 \Big) \left| m_2^{-1} \right|_{m_3} \right|_{m_3} \\
&\vdots \\
\vartheta_n &= \left| \Big(\cdots \Big((x_3 - \vartheta_1) \left| m_1^{-1} \right|_{m_n} - \vartheta_2 \Big) \left| m_2^{-1} \right|_{m_n} - \cdots - \vartheta_{n-1} \Big) \left| m_{n-1}^{-1} \right|_{m_n} \right|_{m_n}
\end{aligned} \tag{1.3}$$

Now in the residue representation of a number, an error in one digit position cannot affect or corrupt the other digit positions. This means that if an error occurs in one of the digit positions, arithmetic computation may continue by excluding the faulty digit position if only the remaining channels are enough to provide a sufficient dynamic range. It is also possible to include extra (redundant) moduli in the system to provide larger and sufficient dynamic ranges to cater for such errors, [1, 14, 15]. Therefore, the use of redundant moduli can facilitate error detection and correction. A Redundant Residue Number System (RRNS) is deemed to be a selected RNS representation with additional redundant moduli. This is usually considered in the design of RNS systems in order to achieve self-checking, and error detection and correction, [16].

Assume as an example, we have $m_1, m_2, , m_N$ as the information moduli, such that the information dynamic range is $\left[0, \prod_{i=1}^{N} m_i\right)$, then we can add R redundant moduli as $m_{N+1}, m_{N+2}, , m_n$ with a dynamic range $M_R = \prod_{j=1}^{n-N} m_{N+j}$. Thus, any operand X has a legitimate range within $[0, M)$ and an illegitimate range within the interval $[M, MM_R)$ but is called for purposes of error detection and correction, [17]. In this paper, we employ the properties of redundancy in RNS by adding two extra moduli to the chosen moduli set $\{2^{n-1} - 1, 2^n, -1, 2^n\}$ for the detection and correction of single errors; the approach laid out in this paper is capable of detecting and correcting any number of errors with a certain number of required redundant moduli. Thus, a forward conversion process is performed first, to represent a given vector of data in its residue equivalent form, from which the error detection and correction can be achieved. That data will then be appreciated if it is brought back into conventional representation through a reverse conversion process, which is undertaken using the CRT. The rest of the paper is organised as follows: In Section 2, review of related works is done, the methodology with some numerical illustration of the proposed scheme is presented in Section 3, with its performance evaluated in Section 4. The paper is finally concluded in Section 5.

2 LITERATURE REVIEW

Over the years, the notion of fault tolerance has been elaborately researched on and achieved through the use of RNS, but the discovery of the redundant moduli in addition to an RNS makes it possible for the detection and correction of errors during transmission. These concepts have been employed extensively in data security; during transmission of a secured data either hidden (steganography) or encrypted (cryptography), where there certainly will be the possibility of errors that can be detected and corrected using RRNS.

The work by [16] discussed the arithmetic of RRNS based codes and reviewed their properties. The paper then proposed a number of application areas for the RRNS codes by demonstrating how it can be employed in areas such communication systems. The proposition in this regard concerned the simplification of the associated systems by unifying an entire encoding and decoding processes across such systems. It is worthy of note that security is a major concern during communication, therefore, even as this work was not applied to security aspect, the RRNS can be used to detect and correct errors when a secured data is communicated or transmitted through a channel. Also, the work failed to show an architecture of the conversion processes to and from RNS.

Earlier, a work by [18] highlighted the need for error detection and correction using RRNS. They opined that, in spite of rapid advances in the design and realization of digital filters, very little attention was paid to the problems of error detection and correction in digital filters. In light of this, the work described a procedure on how to employ the properties of RRNS for that purpose. However, what was presented placed emphasis on overflow detection, and errors in digital filters. This concept could have been extended to cover secure data along a transmission line if time and resources were unlimited. An analysis of the theoretical framework presented in the work show it can only detect single errors with a complex architecture, which has implication for cost. The work by [19], which was on Wireless Sensor Networks (WSNs) but anticipated the possibility of errors that could hinder the ability to monitor and interact with base systems. They emphasised on the fact that the realisation of a fault tolerant operation is very critical to the success of WSNs since the integrity of data could have tremendous effects on performance of such a data acquisition system. Therefore, they proposed the use of RRNS to achieve a fault-tolerant mechanism in base systems because that notion in wireless sensor networks is important due to the construction and deployment characteristics of these low powered sensing devices. It is also a fact that, due to the low computation and communication capabilities of the sensor nodes, the fault-tolerant mechanism should have a low computation overhead which was achieved through RNS when they proposed a low complexity error detection technique through the implementation of low data redundancy and efficient energy consuming in wireless sensor nodes. However, the scheme could not correct more than one errors. Here also, the security of the data that will be transmitted using the sensor nodes was not a concern but it is also very critical in the general network setup.

Also, [20] investigated the use of error correction codes (ECCs) to tolerate faults in hybrid memories. The ECCs considered in the work included Hamming, Reed Solomon (RS), and RRNS codes. Concerning the RRNS codes, they proposed a scheme for detection and correction of single errors. The paper proffered that the RRNS codes are example of block codes, where the checkwords are not computed from the dataword but from the input data. The concept actually employed crossbar memories to improve on its reliability. Experimental results were also shown.

Another work by [21] elucidated that errors are inevitable in data communication due to various factors such noise, heat, and interference in the communication channel/ circuits. They therefore proposed a scheme to detect and correct errors using RRNS. This was done using the moduli set $\{2^{n+1} - 1, 2^n + 1, 2^n\}$ and two redundant moduli $2^{n+1} + 1$ and $2^{2n} + 1$ which works for only even numbers. In the opinion of the researchers backed by some demonstrated facts, the number of iterations in the error correction scheme was reduced that in turned reduced the complexities associated with architectural designs, and also reduced propagation delays. But a careful study of the moduli set used will show that it is complex due to the presence of $2^n + 1$, $2^{n+1} + 1$ and $2^{2n} + 1$ moduli; it also only works accurately with even numbers. Consequently, to achieve desired results of simplicity in the architecture of a scheme, a different moduli set can be chosen to exclude the highlighted moduli. The algorithms presented in these two works can not also be expanded to correct more than one errors.

We note from other works by [15], which applied RNS to the LZW data compression algorithm using the moduli set $\{2^n-1, 2^n, 2^n-1, 2^{n+1}-12^{2n}-3, 2^{2n}+1\}$, first to develop a new LZW-RNS compression and encryption scheme and second, to detect and correct errors during the decoding process since the last two moduli were redundant and just added for that purpose. Thus, the scheme was a four-channelled with two extra channels for error detection and correction. However, the choice of the above moduli set meant that, for pairwise number representation in RNS, the scheme could only work for even numbers. The number of iterations to detect a single error presented in this proposed scheme were many and that can consume a lot of time; [22] presented some results on multiple error detection and correction based on the RRNS; an enhanced multiple error detection and correction scheme was also presented by [23] using the RRNS for communication systems. The paper reiterated the fact that in communication systems corruption and hacking of data is unavoidable. It acknowledge also, the fact that RRNS is often used in parallel processing environments and has the ability to increase the robustness of information passing between the processors. It therefore, proposed a multiple error correction scheme that utilizes the CRT together with an algorithm that simplifies the error correcting process for integers. The scheme was applied using the CDMA. This scheme was concerned with general data passing through a communication channel but the encoding and decoding of cipher messages may result in some errors even before it is communicated through the channel. A clear cut forward conversion procedure was not presented by these works, but it is only after the forward conversion processes that error detection and correction can be performed.

In this paper , we propose an efficient scheme that can detect and correct both single and multiple errors during computation and transmission of encoded data provided the redundant moduli are sufficient for cryptographic and steganographic schemes. The proposed scheme will also clearly show both the forward and reverse conversion processes as well as a step by step procedure to detecting and correcting possible errors in cryptographic and steganographic schemes such as the scheme by [24] with emphasis on recovering the exact original data.

3 PROPOSED SCHEME

Given the moduli set $\{2^{n-1}-1, 2^n, -1, 2^n\}$, where $m_1 = 2^{n-1}-1$, $m_2 = 2^n-1$ and $m_3 = 2^n$. Let $m_4 = 2^{2n}-3$ and $m_5 = 2^{2n}+1$ be redundant moduli to be used for detecting and correcting errors during the decoding process. Thus, the residue set corresponding to the information part is $[r_i]_{i=1}^{3}$ and that corresponding to the parity/redundant is r_4 and r_5.

3.1 Forward Conversion

In order to represent an integer (in this case the ASCII/Unicode or image pixel values) in the RNS form from the decimal or binary representation, there must be a forward conversion process using the moduli set $\{2^{n-1}-1, 2^n, -1, 2^n, 2^{2n}-3, 2^{2n}+1\}$. This includes the redundant moduli so that possible errors in the encoding process can be detected and corrected during decoding. For the chosen moduli set, any legitimate binary number X, which is $(3n-1)$-bits wide, [24] can be partitioned into three sub-blocks for easy implementation as

$$\underbrace{X_{3n-2}\cdots X_{2n}}_{B_3,(n-1)}|\underbrace{X_{2n-1}\cdots X_n}_{B_2,n}|\underbrace{X_{n-1}\cdots X_0}_{B_1,n} \tag{3.1}$$

and computed using

$$X = B_1 + 2^n B_2 + 2^{2n} B_3 \tag{3.2}$$

such that,

$$\begin{aligned} x_1 &= |X|_{2^{n-1}-1} \\ &= \left||B_1|_{2^{n-1}-1} + |2^n B_2|_{2^{n-1}-1} + \left|2^{2n} B_3\right|_{2^{n-1}-1}\right|_{2^{n-1}-1} \\ &= \left|B_1 + 2B_2 + 2^2 B_3\right|_{2^{n-1}-1} \end{aligned} \tag{3.3}$$

$$\begin{aligned} x_2 &= |X|_{2^n-1} \\ &= \left||B_1|_{2^n-1} + |2^n B_2|_{2^n-1} + \left|2^{2n} B_3\right|_{2^n-1}\right|_{2^n-1} \\ &= |B_1 + B_2 + B_3|_{2^n-1} \end{aligned} \tag{3.4}$$

$$x_3 = |X|_{2^n} = B_1 \tag{3.5}$$

and,

$$\begin{aligned} x_4 &= |X|_{2^{2n}-3} \\ &= \left||B_1|_{2^{2n}-3} + |2^n B_2|_{2^{2n}-3} + \left|2^{2n} B_3\right|_{2^n-1}\right|_{2^{2n}-3} \\ &= |C_1 + 3B_3|_{2^{2n}-3} \end{aligned} \tag{3.6}$$

$$\begin{aligned} x_5 &= |X|_{2^{2n}+1} \\ &= \left||B_1|_{2^{2n}+1} + |2^n B_2|_{2^{2n}+1} + \left|2^{2n} B_3\right|_{2^n-1}\right|_{2^{2n}+1} \\ &= |C_1 - B_3|_{2^{2n}+1} \end{aligned} \tag{3.7}$$

where,

$$\begin{aligned} C_1 &= B_1 + 2^n B_2 = B_1 \bowtie B_2 \\ &= C_{1,2n-1} C_{1,2n-2} \cdots C_{1,1} C_{1,0} \end{aligned} \tag{3.8}$$

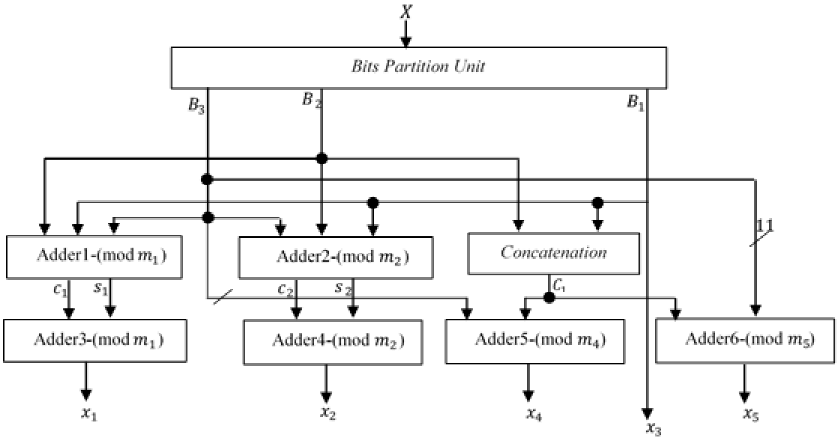

Fig. 2. Block diagram of forward conversion process for the proposed scheme

These mathematical processes/equations are represented diagramatically using a schematic diagram in Figure .

According to the proposed layout of the block diagram in Figure , the binary number X is first passed through a bits partition unit in order to partition the bits based on block sizes as shown in Equation (3.1). We note that the residue x_3 is equivalent to the first block as it is the n least significant bits of the number as shown in Equation (3.5). 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 are computed with Adders 3 and 4. At the same time, the result of the concatenation is added with B_3 using Adders 5 and 6 to get residue 5 and residue 6 respectively. Adders 1 and 2 are Carry Save Adders (CSAs) whilst Adders 3, 4, 5 and 6 are Carry Propagate Adders (CPAs). The deployment of all the CPAs at the same time helps in saving computation time, thereby improving the delays associated with the converter since this will be determined by the modulus with the higher value.

3.2 Reverse Conversion

If a cryptographic or steganogrphic scheme is developed with the concept of RNS, there usually exist a stage in the decoding/ decryption process that values in the RNS representation must be converted back into their binary equivalent forms in order to actually decipher the meaning of the encoded messages. It is at this stage that errors (if there exist any) that may have occurred during the encoding process ought to be detected and corrected to get the true meaning of any message. The CRT is employed for the reverse conversion process; in this manner if any integer X is chosen from the range of $[0, M_N)$, any N residues out of the n residues, where $n > N$ should be sufficient in recovering the original number/integer X.

Given the moduli set $\{2^{n-1} - 1, 2^n, -1, 2^n, 2^{2n} - 3, 2^{2n} + 1\}$, where $m_2 = 2^n - 1$, $m_3 = 2^n$, $m_4 = 2^{2n} - 3$ and $m_5 = 2^{2n} + 1$, then from Equation (1.1) we have:

$$\left.\begin{aligned}
\ell_1 &= 2^n(2^{2n} - 3)(2^{2n} + 1)(2^n - 1)\\
\ell_2 &= 2^n(2^{2n} - 3)(2^{2n} + 1)(2^{n-1} - 1)\\
\ell_3 &= (2^{n-1} - 1)(2^{2n} - 3)(2^{2n} + 1)(2^n - 1)\\
\ell_4 &= 2^n(2^n - 1)(2^{2n} + 1)(2^{n-1} - 1)\\
\ell_5 &= 2^n(2^n - 1)(2^{2n} - 3)(2^{n-1} - 1)
\end{aligned}\right\} \tag{3.9}$$

and,

$$\left.\begin{aligned}
k_1 &= 1\\
k_2 &= 2^{n-1}\\
k_3 &= 1\\
k_4 &= 1\\
k_5 &= 2^{2n} - 2
\end{aligned}\right\} \tag{3.10}$$

Therefore, any number X in RNS representation can be converted back to its decimal/binary equivalent form as in Equation (3.11).

$$X = \left|\ell_1 x_1 + 2^{n-1}\ell_2 x_2 + \ell_3 x_3 + \ell_4 x_4 + (2^{2n} - 2)\ell_5 x_5\right|_M \tag{3.11}$$

Equation (3.11) can be further simplified and implemented as follows:

$$X = |A_1 + A_2 + A_3 + A_4 + A_5|_M \tag{3.12}$$

where,

$$A_1 = \ell_1 x_1 = \left|\ell_1\left(x_{1,n-2}x_{1,n-3}\cdots x_{1,1}x_{1,0}\right)\right|_M \quad (3.13)$$

$$A_2 = 2^{n-1}\ell_2 x_2 = \left|\ell_2 A_{21}\right|_M \quad (3.14)$$

$$A_3 = \ell_3 x_3 = \left|\ell_3(x_{3,n-1}x_{3,n-2}\cdots x_{3,1}x_{3,0})\right|_M \quad (3.15)$$

$$A_4 = \ell_4 x_4 = \left|\ell_4(x_{4,2n-3}x_{4,2n-4}\cdots x_{4,1}x_{4,0})\right|_M \quad (3.16)$$

$$A_5 = \left(2^{2n}-2\right)\ell_5 x_5 = \left|\ell_5\left(2^{2n}x_5 - 2x_5\right)\right|_M = \left|\ell_5 A_{51}\right|_M, \quad (3.17)$$

and

$$A_{21} = \left(x_{2,n-1}x_{2,n-2}\cdots x_{2,1}x_{2,0}\overbrace{00\cdots 0}^{n-1}\right) \quad (3.18)$$

$$A_{51} = \left(x_{5,2n}\cdots x_{5,1}x_{5,0}\overbrace{00\cdots 0}^{2n}+\bar{x}_{5,2n}\cdots\bar{x}_{5,0}\right) \quad (3.19)$$

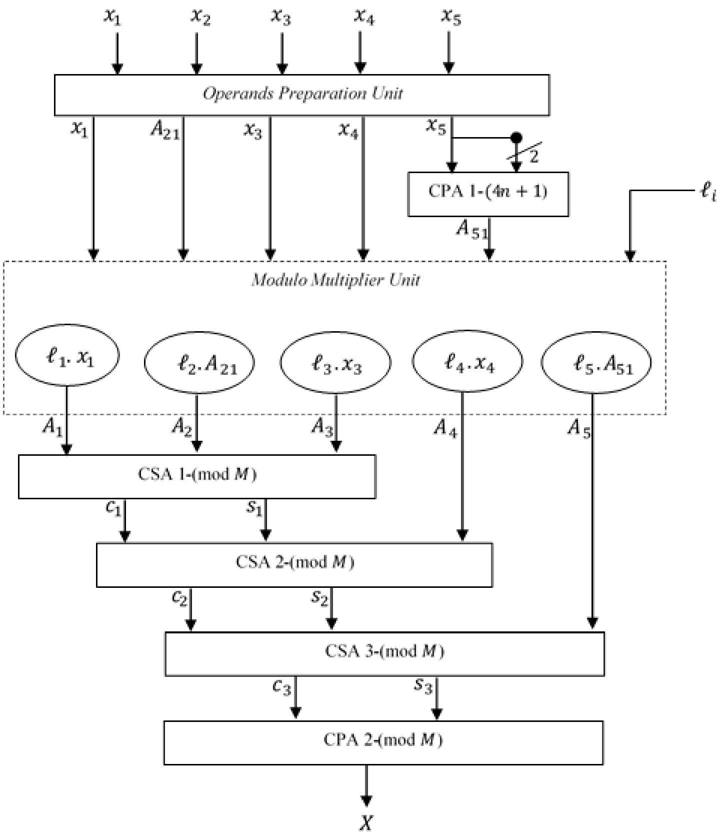

Fig. 3. Schematic reverse converter for the proposed scheme

Figure is the schematic diagram for the proposed reverse converter; the operands operation unit prepares and manipulate an appropriate routing of the residue bits including flipping and extending in some cases. It is worthy of note that, as a result of the M modulo operation and the number of moduli involved, using only adders imply employing a whole lot of them. Therefore, we decided to suggest/use a modulo multiplier to reduce this overhead cost. CPA 1 is a computation of Equation (3.19), which is further used for a multiplication in Equation (3.17). Equations (3.13) - (3.16) are results of a multiplication operation using the modulo multiplier. The parameter so obtained from this operation should then be added in a cascading manner using CSAs 1 3 by ploughing back the respective saves (s) and carries (c) into the following adder. The save (s_3) and carry (c_3) from the final CSA (CSA 3) is now computed using CPA 2 in order to obtain X. It should be noted that this includes the redundant moduli, thus the converter computes for all the residues in the vector (information and redundant).

3.3 Error Handling

The following lemmas, [22, 23] are useful in relation to RRNS for error detection/correction:

Lemma 3.1. *A code Ω based on an RRNS has the minimum nonzero Hamming weight $wt_{min} \geq r+1$ and minimum distance $d_{min} \geq r+1$*

Lemma 3.2. *A code Ω based on an RRNS can correct up to t errors, $t \leq \lfloor r/2 \rfloor$, r is the number of redundant moduli.*

Assume an integer X is selected from a range $[0, M_N)$ with the corresponding residue set $[x_1, x_2, x_N, x_{N+1} x_n]$, where N and n are chosen such that **Lemma** 3.2 holds. If this integer is passed through a noisy system such as the encoding/encryption process, it is possible that errors may be introduced into the residue vector. Let the new vector with errors be

$$\delta_i = x_i + e_j,\ i = 1, 2n \quad \& \quad j = 1, 2, t \tag{3.20}$$

Now at the decoding end, when δ_i is received, it validity is tested by checking for possible errors using the CRT in Equation (1.1). If the recovered vector, δ is within the legitimate range, then it is valid and no further action will be needed. But if it is outside the legitimate range, then it can be concluded that δ contain errors in its residues. The relationship between X and δ will be

$$X = \left|\delta_i - E\right|_M, \quad 0 \leq E \leq M \tag{3.21}$$

Where, E is the amount of error that is propagated into X; its magnitude can be evaluated in a likewise manner using Equation (1.1) as

$$E = \left|\sum_{j=1}^{t} \ell_j \delta_j e_j\right|_M \tag{3.22}$$

Let M in (3.22) be expressed as

$$M = \prod_{i=1}^{n} m_i = \prod_{\alpha=1}^{t} m_\alpha . \prod_{\beta=l_1}^{l_{n-t}} m_\beta \tag{3.23}$$

where, $[\alpha_i]_{i=1}^{t}$ are the positions of residues with errors and $[\beta_i]_{i=l_1}^{l_{n-t}}$ are the remaining positions without errors inside the vector δ. Therefore, let

$$\hat{M} = \prod_{\beta=l_1}^{l_{n-t}} m_\beta \tag{3.24}$$

be the product of all moduli corresponding to residues without errors in δ. This will make it possible for us to detect and correct any number of errors if and only if **Lemma 3.2** is satisfied.

Theorem 3.3. *For an RRNS code with proper amount of redundancies r, such that the number of errors that occur in a received vector, δ is $t \leq \lfloor r/2 \rfloor$, the original integer X can be found by evaluating the equation*

$$X = |\delta|_{\hat{M}} \tag{3.25}$$

Proof. Equation (3.25) is employed iteratively to find the one combination of $\hat{M}$ (in $u = {}^{n}C_r$) which produces the integer X that falls within the legitimate range. This will imply a maximum of u possible combinations since the position(s) of error(s) cannot be determined apriori. □

These processes can be simplified into an algorithm as follows:

Algorithm for the proposed error detection and correction

I. Compute δ from the received vector δ_i using Equation (1.1)

II. If δ is in the legitimate range, stop and output δ

III. Compute iteratively starting from $i = 1$ using Equation (3.25) to obtain X. If X is in the legitimate range, stop and output X. Otherwise, increment i and repeat Step III for $i \leq u$

IV. Compute $X \mod e_i$, i is the error position where X is in the legitimate range.

3.4 Numerical Illustration

Let us demonstrate how this technique can be used to detect and correct errors (in this case a single error but applicable to any number of errors provided the redundant moduli are sufficient to satisfy Lemma 3.2). For the chosen moduli set $\{2^{n-1} - 1, 2^n, -1, 2^n, 2^{2n} - 3, 2^{2n} + 1\}$, if $n = 3$, then we have the set $\{3, 7, 8, 61, 65\}$. Since the moduli set contains $r = 2$ redundant moduli, implies it can correct $t = \lfloor 2/2 \rfloor = 1$ error. Also the legitimate range is $[0, 168)$ while the illegitimate range is $[168, 666120)$. Now, let $X = 97$ and the equivalent residue vector $x = [1, 6, 1, 36, 32]$ such that during transmission an error is propagated at the first position (i.e. $e_1 = 2$), the received vector will be $\delta = [2, 6, 1, 36, 32]$. Thus,

$$x_i = [1, 6, 1, 36, 32]$$
$$\delta_i = [\underline{\mathbf{2}}, 6, 1, 36, 32]$$

To obtain δ using Equation (1.1), we substitute the residue vector of δ and all necessary parameters of the equation to get

$$\delta = \left| \sum_{i=1}^{5} \ell_i \left| k_i \delta_i \right|_{m_i} \right|_M$$
$$= 222137$$

The respective ℓ_i and multiplicative inverses, k_i are:
$\ell_i = [222040, 95160, 83265, 10920, 10248]$ and
$k_i = [1, 4, 1, 1, 62]$.
Since the result for $\delta = 222137$ falls within the illegitimate range, it can be concluded that an error has occurred; similarly, we notice that when the errors occurred at positions 2 and/or 5 (i.e. $e_2 = 3$, $e_5 = 12$), the calculated results fall in the illegitimate range, and so we have to further process by employing the proposed technique. The results are shown in Table by performing all the possible combinations. From Table , it is observed that every iteration for X yields an illegitimate value except at the position where the error is introduced. This sole legitimate number is turned out in all the cases to be the correct integer that was transmitted. Now to get the respective residue in error, a modulo operation of that number by its modulo is performed. That is, $97 \mod 3$ equals 1; likewise the other examples for channels 2 and 5.

Table 1. Results of error detection and correction at selected positions

No. of Iterations, i	$\hat{M}$	$X = \lvert\delta\rvert_{\hat{M}}$	$X = \lvert\delta\rvert_{\hat{M}}$	$X = \lvert\delta\rvert_{\hat{M}}$
		$e_1 = 2, \quad \delta = 222137$	$e_2 = 3, \quad \delta = 190417$	$e_5 = 12, \quad \delta = 369025$
1	**222044**	**97**	**190417**	**1469885**
2	**95160**	**31817**	**97**	**83545**
3	**83265**	**55607**	**23887**	**35965**
4	**10920**	**3737**	**4777**	**8665**
5	**10248**	**6929**	**5953**	**97**

4 PERFORMANCE EVALUATION

The results on the performance of the proposed scheme were analysed theoretically in terms of the hardware requirements, which has an implication for cost; the delay imposed by the various units/stages, which is a factor of the speed of the proposed scheme; and finally compared with existing similar existing schemes on key parameters.

4.1 Hardware Requirements

Regarding the hardware requirements of the proposed scheme, the forward converter is made up of six modulo adders comprising of two CSAs and two CPAs to compute the first and second residues respectively; these are $(n-1)$-bits and n-bits wide. The third residue is the n-least significant bits of the binary number X and so, does not require any computation but the fourth and fifth residues are also respectively computed using two CPAs; these are also $(2n-1)$-bits and $(2n+1)$-bits wide. Therefore, the estimated area for the forward converter is $(8n-2)\Delta_{FA}$. The reverse converter requires two CPAs of $(4n+1)$-bits wide (for CPA 1) and modulo M (i.e. CPA2). There are also three modulo M CSAs implemented in a cascading fashion. Finally, a multiplier (a Booth encoded) unit, [25, 26, 27] to do five different multiplication operations. These will require a total area of $(5n^2+32n-2)\Delta_{FA}$. According to the proposed architecture for implementing the forward converter, there are only two stages due to the parallel implementation. Therefore the estimated delay for the forward converter is $D_{Adder2}+D_{Adder6} = (4n+3)\Delta_{FA}$ and the estimated delay for the revere converter is $(5n^2+22n+1)\Delta_{FA}$. Finally, the number of iterations during the error detection and correction is $u = {}^{n}C_t$, thus ${}^{5}C_1 = 5$ iterations.

4.2 Comparison of Proposed Scheme with Existing Schemes

Next, we compared the performance of the proposed scheme with existing similar schemes on some parameters as shown in Table .

Table 2. Performance comparison

Scheme	*Iterations*	*Error Correction*		*Architecture*	
		Single	*Multiple*	*F/C*	*R/C*
[22]	${}^{n}C_t$	NO	YES	NO	NO
[21]	$2\times{}^{n}C_t$	YES	NO	YES	NO
[23]	${}^{n}C_t$	NO	YES	NO	NO
[15]	$2\times{}^{n}C_t$	YES	NO	YES	NO
Proposed	${}^{n}C_t$	YES	YES	YES	YES

From the table, it is seen that the schemes presented by [21] and [15] have a high number of iterations $(i.e. 2\times{}^{n}C_t)$ in order to detect and correct just a single error during transmission. These schemes presented architectures for the forward conversion (F/C) process but in doing so, they failed to present a combined converter for both the information and redundant parts but did do separately. This approach is not good for hardware optimisation. These schemes however, failed to present architectures for the reverse conversion (R/C) process. The schemes by [22] and [23] on the other hand, did not show any architecture for both the forward conversion and reverse conversion processes. However, these schemes detected and corrected multiple errors with ${}^{n}C_t$ number of iterations (i.e. half the number of iterations for the schemes by [21] and [15]. A comparison using the above parameters with the proposed scheme puts it a notch higher. Thus, whiles the number of iterations are lower (i.e. $^{n}C_{t}$), it detects both single and multiple errors during transmission. The scheme has also shown the architectures for both the forward conversion (combined information and redundant parts) process and the reverse conversion process.

5 CONCLUSION

The paper presented an error detection and correction technique using RRNS with two redundant moduli in addition to the moduli set $\{2^{n-1}-1, 2^{n}, -1, 2^{n}\}$; this is necessary because during the encoding process of security systems such as cryptographic and stenographic schemes there is the likelihood of errors being propagated. In that case, the errors need to be detected and corrected during the decoding process in order to ensure that whatever is transmitted at the senders end is received accurately at the receivers end. The proposed error detection and correction scheme demonstrated that depending on the number of redundant moduli, it can detect and correct both single and multiple errors. Hardware realisation of the scheme at both ends – binary to RNS and RNS to binary, was also presented. Finally, a theoretical analysis show that it outperforms existing similar state of the art schemes.

ACKNOWLEDGMENT

The authors thank all the reviewers for their valuable contributions.

COMPETING INTERESTS

The authors declare that they have no competing interests.

REFERENCES

[1] Amos Omondi, Benjamin Premkumar. Residue number systems: theory and implementation, volume 2. Published by Imperial College Press and Distributed by World Scientific Publishing Co.; 2007.
ISBN 978-1-86094-866-4, 978-1-86094-867-1.
Available:http://www.worldscientific.com/-worldscibooks/10.1142/p523

[2] Molahosseini AS, Navi K. New Arithmetic residue to binary converters. International Journal of Computer Sciences and Engineering Systems. 2007;1(4):295299.

[3] Gbolagade KA. Effective reverse conversion in residue number system processors. PhD Thesis, The Netherlands; 2010.

[4] Agbedemnab PA. Overflow detection and correction techniques in RNS arithmetic comutations. PhD Thesis, University for Development Studies, Tamale; 2015.

[5] Dina Younes. Residue number based building blocks for applicaitons in digital signal processing. PhD Thesis, Czech Republic; 2013.

[6] Bhardwaj M, T Srikanthan, Clarke CT. A reverse converter for the 4 moduli super set $\{2^{n}-1, 2^{n}, 2^{n}+1, 2^{n+1}+1\}$. IEEE Conference on Computer Arithmetic; 1999.

[7] Srikanth P, Abhinav Mehta, Neha Yadav, Sahil Singh, Shubham Singhal. Encryption and decryption using genetic algorithm operations and pseudorandom number. Computer Science and Network. 2017;6(3):455459.

[8] Chang CH, Low J, Yung S. Simple, fast, and exact RNS scaler for the three-moduli set $\{2^{n}-1, 2^{n}, 2^{n}+1\}$. IEEE Transaction on Circuits and Systems I: Regular Papers. 2011;58(11):26862697.

[9] Gbolagade KA, Chaves R, Sousa L, Cotofana SD. An improved reverse converter for $\{2^{2n+1}-1, 2^{n}, 2^{n}-1 moduliset\}$. IEEE International Symposium on Circuits and Systems (ISCAS 2010). 2010;21032106.

[10] Bankas EK, Gbolagade KA. A residue to binary converter for a balanced moduli set $\{2^{2n+1}, 2^{2n}, 2^{2n}-1\}$. In Awareness Science and Technology and Ubi-Media Computing (iCASTUMEDIA). 2013;21216.

[11] Agbedemnab PA, Bankas EK. 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. 2015;110(16):3034. ISSN 09758887. DOI: 10.5120/19403-0925 Available:http://research.ijcaonline.org/volume110/number16/pxc3900925.pdf

[12] Leonel Sousa, Paulo Marins. Sign detection and number comparison on rns augumented 3-moduli sets $\{2^n - 1, 2^{n+x}, 2^n + 1\}$. IEEE Transactions on Very Large Scale Integration Systems: Journal of Latex Class Files. 2014;20(2).

[13] Mohmmad Abdallah, Alexander Skavantzos. A systematic approach for selecting practical moduli sets for residue number systems. Proceedings of the 27th Southeastern Symposium on System Theory, SSST. 1995;445449. DOI:10.1109/SSST.1995.390542

[14] Bankas EK, Gbolagade KA. A speed efficient RNS to binary converter for the moduli set $\{2^n, 2^n + 1, 2^n - 1\}$. Journal of Computing, 2012;4(5):8388.

[15] Abdul-Barik Alhassan, Kazeem Alagbe Gbolagade, Edem Kwedzo Bankas. New lempel- ziv-welch fault tolerant data compression and encryption scheme. International Journal of Advanced Studies in Engineering and Scientific Inventions (IJASESI). 2017;4(1):2536.

[16] Lie-liang Yang, Lajos Hanzo. Reductant residue number system based error correction codes. IEEE Transaction on Circuits and Systems I: Regular Papers. 2001;15.

[17] Lie-liang Yang, Lajos Hanzo. Coding theory and performance of redundant residue number system codes coding theory and performance of redundant residue number system codes. IEEE Transaction on Information Theory.1999;(0):039.

[18] Etzel M, W Jenkins. Redundant residue number systems for error detection and correction in digital filters. IEEE Transactions on Acoustics, Speech and Signal Processing. 1980;28(5):538545. ISSN 0096-3518. DOI: 10.1109/TASSP.1980.1163442 Available: http://files/38/login. html

[19] Roshanzadeh M, Saqaeeyan S. Error detection & correction in wireless sensor networks by using residue number systems. International Journal of Computer Network and Information Security. 2012;4(2):2935. ISSN 20749090. DOI:10.5815/ijcnis.2012.02.05

[20] Nor Zaidi Haron, Said Hamdioui, Zaiyan Ahyadi. ECC design for fault-tolerant crossbar memories: A case study. In IDT10 - 2010 5th International Design and Test Workshop, Proceedings. 2010;6166. ISBN 9781612842929. DOI:10.1109/IDT.2010.5724409

[21] Abdul-Mumin Salifu, Kazeem Alagbe Gbolagade. An improved redundant residue number system based error detection and correction scheme for the moduli set. 2016;2(1):1114. DOI: 10.11648/j.awcn.20160201.12

[22] Vik Tor Goh, Mohammad Umar Siddiqi. Multiple error detection and correction based on redundant residue number systems. IEEE Transactions on Communications. 2008;56(3):325330. ISSN 00906778. DOI: 10.1109/TCOMM.2008.050401

[23] Karthik G, Mohan Raj R, Karthik B. RRNS based error detection and correction in CDMA using chinese remainder theorem. 2016;2(2):338341.

[24] Peter Awon-Natemi Agbedemnab, Edward Yellakuor Baagyere, Mohammed Ibrahim Daabo. A novel text encryption and decryption scheme using the genetic algorithm and residual numbers. In Kennedy Njenga, editor, Proceedings of 4th International Conference on the Internet, Cyber Security and Information Systems 2019, volume 12 of Kalpa Publications in Computing. 2019;2031. EasyChair

DOI: 10.29007/zd9h
Available: https://easychair.org/publications/-paper/kzG7

[25] Gary W. Bewick. Fast multiplication : Algorithms and implementation. PhD Thesis; 1994.

[26] Sheu M, Lin SH, Chen C, Yang SW. An efficient VLSI design for a residue to binary converter for general balance moduli set $\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$. IEEE Transactions on Circuits and Systems II. 2004;51(3):152155.

[27] Kumar Ch. Harish. Implementation and analysis of power, area and delay. International Journal of Scientific and Research Publications. 2013;3(1):15.

©2019 Agbedemnab et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/53646