Algorithm for Clustering the Moduli of RNS for the Application of Optimization of Time Complexity in Standard Cipher System

Radhakrishna Dodmane, Ganesh Aithal, Surendra Shetty

***Abstract:** Residue number system (RNS) has emerged as a knocking field of research due to its high speed, fault tolerant, carry free and parallel characteristics. Due to these features it has got important role in high performance computing especially with reduced delay. There are various algorithms have been found as a result of the research with respect to RNS. Additionally, since RNS reduces word length due to the modular operations, its computations are faster compared to binary computations. But the major challenges are the selection of moduli sets for the forward (decimal to residue numbers) and reverse (residue numbers to decimal) conversion. RNS performance is purely depending on how efficiently an algorithm computes / chooses the moduli sets [1]-[6].*

This paper proposes new method for selecting the moduli sets and its usage in cryptographic applications based on Schonhage modular factorization. The paper proposes six moduli sets {6qk –1, 6qk+1, 6qk+3, 6qk+5, 6qk+7, 6qk+11} for the RNS conversions but the Schonhage moduli sets are expressed as the exponents that creates a large gap between the moduli's computed. Hence, a new method is proposed to for computing moduli sets that helps in representing all the decomposed values approximately in the same range.

***Keywords:** Residue arithmetic, conversion, moduli set, residue number system (RNS).*

I. INTRODUCTION

In cryptography, computation speed reduces as the value of the data considered for the operation increases. Therefore, the best solution is to reduce the value of the data or may be the key space used, but it becomes more vulnerable to the attacks as go on reduce value. Hence, there is a need of simple and efficient model / system to handle this problem. One of the solutions is Residual Number System (RNS) that could handle the large value as a smaller component which can be operated in parallel and thus helps in increasing the computation speed.

The RNS is widely accepted in cryptographic process because of its knocking characteristics such as modularity, parallelism and the carry free operation [1-3]. In RNS, the set of residues are computed from the given large number by using the moduli sets, identified / determined in advance, this may be termed as forward conversion. All the arithmetic operations are then carried out on these residues as modulo operations by using the corresponding moduli's. Once completing all the operations, these non-weighted residues are converted back as single large number, this process may be termed as reverse conversion. These RNS process is shown in the Fig. 1.

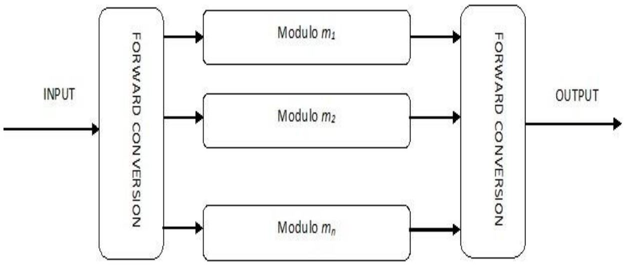

Fig. 1. RNS Processor

Even though RNS have got advantages, the reverse conversion process may influence the performance due to its overhead. Therefore, the moduli's required must be selected carefully and must design a precise algorithm for both forward and reverse process to capture the advantages of RNS. To exploit the parallelism, it is very important to have balanced set of moduli's for the applications [4]-[5]. Additionally, the reverse conversion process itself is complex and slow because it has to combine all the residues computed in the forward conversion [6].

There are many proposals for selecting the moduli sets for the RNS such as three, four and five moduli sets [7]-[9], but these moduli sets are not suitable for applications that operate on large word length. Similarly, there are other set of moduli's with dynamic range are proposed in [10]-[18]. In all the moduli sets proposed, one common finding is the difference between moduli's selected are in terms of 2n (where n ≥ 0). This large difference reveals some statistics about data used in the RNS process, which would be advantage for the cryptanalyst if adopted in cryptographic process. Hence, it has been thought of having balanced set of six moduli in the RNS process in tern adopt cryptographic process. The set of six moduli chosen is based on the Schonhage theory but eliminating the 2n difference among the moduli's. The method of selection of moduli sets and its application are discussed subsequent sections.

Revised Manuscript Received on May 30, 2020.

* Correspondence Author
Radhakrishna Dodmane*, Computer Science and Engineering, NMAM Institute of Technology, Nitte, Karkala, INDIA. Email: rkdodmane@gmail.com
Ganesh Aithal, Vice-principal, SMVITM, Bantakal, INDIA. Email: ganeshaithal@gmail.com
Surendra Shetty, HOD, MCA, NMAM Institute of Technology, Nitte, Karkala, INDIA. Email: hsshetty@nitte.edu.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Rest of the paper is organized as follows. The details of the RNS is discussed in the section background. The proposed method for computing the moduli sets are presented in the section III. The section IV contains the application of the proposed method on the cryptographic standard. Experimental analysis based on the proposed method are presented in section V. finally the paper concluded in the last section.

II. BACKGROUND

A. Residue Number System (RNS):

Residue Number System is developed to speed up the computations by decomposing the large integers into smaller components usually called as residues. These smaller components are generated by performing modular operation on the large integer such that all these residues could be operated independently due to its in-built parallel nature. Since these residues are very small values, computations on these residues becomes easier and are directly proportional to the power required to process. In RNS the moduli's $\{m_1, m_2, \dots, m_n\}$ chosen are relatively pairwise co-prime that is $gcd(m_i, m_j) = 1$ for $i \neq j$.

Any weighted number to be processed using RNS is represented as follows $X = \{x_1, x_2, \dots, x_n\}$, such that,

$$x_i = X \bmod m_i \tag{1}$$

Such a representation is unique for any integer X in the range $[1, M]$, where M the dynamic range of the moduli is set $\{m_1, m_2, \dots, m_n\}$, which is equal to the product of m_i terms $M = \{m_1, m_2, \dots, m_n\}$ [19].

The forward conversion in RNS converts weighted integers into residue. On these residues the arithmetic operations such as multiplication, and additions can be performed in parallel without the carry propagation between the residues, resulting in high speed parallel operations. These moduli sets has to be converted back to weighted number to use outside the RNS using reverse converter as shown in the figure 1.

Similarly, Schonhage has developed a modular arithmetic based factorization of large integer specially for multiplying large integers rapidly, which is discussed in the next section.

B. Schonhage Modular approach:

Schonhage has developed an approach to multiply larger integers rapidly by modular arithmetic [29]. At first it was hard to believe that the choice of moduli could be advantageous over multiplications and the process of conversion of numbers into and out of modular representations. But Schonhage has proved that modular multiplications are faster.

In his theory, he has defined many rules for deciding the number of bits on which the multiplications are carried out. Upon deciding the number of bits, six moduli sets are computed, that are relatively pairwise co-prime numbers. In order to understand Schonhage theory let us consider a special case.

First consider the sequence based on the rules

$$q_0 = 1;\ q_{k+1} = 3q_k - 1 \tag{2}$$

And $q_k = 3^k - 3^{k-1} - \cdots - 1 = \frac{1}{2}(3^k + 1)$ (3)

From equation (3), number of bits are decided as:

$$p_k = (18q_k + 8) \tag{4}$$

Upon deciding the number of bits for multiplication, the moduli sets are computed as per the equation (5).

$$m_1 = 2^{6q_k - 1} - 1;\ m_2 = 2^{6q_k + 1} - 1;\ m_3 = 2^{6q_k + 2} - 1;$$
$$m_4 = 2^{6q_k + 3} - 1; m_5 = 2^{6q_k + 5} - 1; m_6 = 2^{6q_k + 7} - 1 \tag{5}$$

These moduli's chosen are relatively prime, because the exponents of these moduli sets $6q_k - 1, 6q_k + 1, 6q_k + 2, 6q_k + 3, 6q_k + 5\ and\ 6q_k + 7$ are all relatively prime. Upon deciding the moduli's, any number '*U*' less than 2^{p_k} is decomposed into smaller residues {u1, u2, u3, u4, u5, u6} as per the below equation (6).

$$u_1 = U\ mod\ m_1;\ u_2 = U\ mod\ m_2;\ u_3 = U\ mod\ m_3;\ u_4 =$$
$$U\ mod\ m_4;\ u_5 = U\ mod\ m_5\ ;\ u_6 = U\ mod\ m_6 \tag{6}$$

The biggest disadvantage of this method is the large difference between the moduli's chosen, since it is driven by the exponents. This large gap may not suitable in the cryptographic applications because it may reveal statistical relationships to the cryptanalysts. Therefore, a novel method for computing the moduli for RNS method is proposed in this paper that tries to eliminate the large gap between the moduli's, which is discussed in the methodology section.

III. METHODOLOGY

From the study it has been observed that, residue number system (RNS) has considerable advantages over speeding up the arithmetic computations because of its parallel property [21]-[25]. The properties like in built parallel, carry-free and modularity with respect to arithmetic operations of the RNS has greater suitability in some applications such as signal processing, coding theory, cryptography [26]. The idea of speed up the computation using RNS is by the application, which allow the larger integer to represent as combinations of smaller integer. These representations of set of all smaller integer by RNS and their component modular multiplication and addition forms the smaller direct sum of commutative rings. This property of direct sum of commutative rings has greater advantages over fixed radix or decimal arithmetic operation if used in cipher system.

Therefore, this idea of decomposition is thought of applying in cipher system because it also offers the transmission of encrypted data over multiple channels [27] and also helps in hiding the ciphertext. Hence, this work relies on proposing a new model to increase the immunity of the cipher system against side channel attack, algebraic attack and known plaintext attack.

The cipher system based on RNS techniques manages smaller integers that are generated from the decomposition of input integers.

This decomposition of large integers facilitates to process the smaller components in parallel due to its in-built carry free, independent nature, which optimizes the speed of execution without compromising to the security. Therefore, the idea is to develop a new mathematical RNS model that would consider as advantages. With this objective the new RNS mathematical model-based approaches on factorization of integer are discussed below.

Schonhage based computation of prime moduli's for RNS

This proposed design is based on modular arithmetic such that it includes the choice of choosing the moduli and conversion of numbers into and out of modular representations. Even though this is difficult, these operations can be performed quite rapidly.

In this proposed method, choosing of the moduli is based on the number of bits used to represent the integer values and the selection of the positive integer which is the key parameter of all. The mechanism is described by considering the specially defined rules as follows. Let X be a positive integer with p_k number of bits and will be split into n factors

Let, $k = 0,\quad q_0 = 1,\quad q_{k+1} = (2q_k + 1)$ (7)

Such that, $q_k = 2^k + 2^{k-1} + \cdots + 1 = 2^k - 1$ (8)

The procedure for determining the number of bits to denote the range of values is defined as per the equation (9). The number of bits is eventually increased almost a factor of two with respect to q_k.

$$p_k = (2q_k + 2) \tag{9}$$

For the determined p_k-bit value, it has been decided to use six moduli and are determined as per the equation (10).

$$m_1 = 6\left(\frac{(q_k-1)}{2}\right) - 1;\ m_2 = 6\left(\frac{(q_k-1)}{2}\right) + 1;\ m_3 = 6\left(\frac{(q_k-1)}{2}\right) + 3,$$
$$m_4 = 6\left(\frac{(q_k-1)}{2}\right) + 5;\ m_5 = 6\left(\frac{(q_k-1)}{2}\right) + 7;\ m_6 = 6\left(\frac{(q_k-1)}{2}\right) + 11 \tag{10}$$

These six moduli's are relatively prime and can be used to determine the smaller integers as per the equation (11), assuming k> 0 and large integer X.

$$x_1 = X \bmod m_1, x_2 = X \bmod m_2, x_3 = X \bmod m_3, x_4 = X \bmod m_4,$$
$$x_5 = X \bmod m_5, x_6 = X \bmod m_6 \tag{11}$$

An Example: The process is best understood if considered an example. Let us assume k=3, such that, $q_k = 2^k - 1 = q_3 = 2^3 - 1 = 8 - 1 = 7;$ and $p_k = (2q_k + 2) = (2 \times 7 + 2) = 16.$

Now, the moduli's are

$$m_1 = 6\left(\frac{(q_3-1)}{2}\right) - 1 = 6\left(\frac{(7-1)}{2}\right) - 1 = 18 - 1 = 17$$

$$m_2 = 6\left(\frac{(q_3-1)}{2}\right) + 1 = 6\left(\frac{(7-1)}{2}\right) + 1 = 18 + 1 = 19$$

$$m_3 = 6\left(\frac{(q_3-1)}{2}\right) + 3 = 6\left(\frac{(7-1)}{2}\right) + 3 = 18 + 3 = 21$$

$$m_4 = 6\left(\frac{(q_3-1)}{2}\right) + 5 = 6\left(\frac{(7-1)}{2}\right) + 5 = 18 + 5 = 23$$

$$m_5 = 6\left(\frac{(q_3-1)}{2}\right) + 7 = 6\left(\frac{(7-1)}{2}\right) + 7 = 18 + 7 = 25$$

$$m_6 = 6\left(\frac{(q_3-1)}{2}\right) + 11 = 6\left(\frac{(7-1)}{2}\right) + 11 = 18 + 11 = 29$$

Upon computing the residues carefully to capture the advantages of RNS, its required to define the precise method for forward and reverse conversion of RNS system, that is discussed in the next section.

Forward and Reverse conversion for the proposed method:

This section explains the simple multiplicative cipher based on RNS. Let M = $\prod_1^n m$ be an integer with $X = \{x_1, x_2, x_3, \dots, x_n\}$ as plain text $A = \{a_1, a_2, a_3, \dots, a_n\}$ as multiplicative key and $C = \{c_1, c_2, c_3, \dots, c_n\}$ as cipher text.

The proposed RNS forward and reverse conversion of is based on CRT method that is defined for the cipher system as follows.

Let X be the integer to be processed in forward conversion into $\{x_1, x_2, x_3, \dots, x_n\}$ by equation (13).

$$x_i = X \bmod m_i;\ \forall\ 1 \le i \le n \tag{13}$$

The cipher text for multiplicative cipher is evaluated by using equation (14). In this equation a_i is used as multiplicative key

$$c_i = (x_i \times a_i) \bmod m_i;\ \ \forall\ \ 1 \le i \le n \tag{14}$$

Such that $(a_i \times a_i^{-1}) \equiv 1 \bmod m_i$. The weighted equivalent of X is computed in the forward conversion by equation (15) as follows:

$$C = (\textstyle\sum_{i=1}^{n}(c_i \times M_i \times M_i^{-1})) \bmod M \tag{15}$$

Where $M_i \times M_i^{-1} \equiv 1 \bmod M_i$, and $M = \prod_{i=1}^{n} m_i;\ \forall\ \ 1 \le i \le n$ and $M_i = \frac{\prod_{i=1}^{n} m_i}{m_i}$.

Similarly, in the reverse conversion the original number are computed as per the equation (16).

$$X = (\textstyle\sum_{i=1}^{n}(x_i \times M_i \times M_i^{-1})) \bmod M \tag{16}$$

Where $x_i = c_i \times m_i^{-1} \bmod m_i;\ \forall 1 \le i \le n$.

From the above observation, it has been observed that the proposed method is best suitable for the arithmetic operations which involves larger operands. Hence, it has been thought of testing the proposed system in the cryptographic standards. In this paper, the proposed system is tested on the symmetric-key technique Blowfish.

IV. EXPERIMENTAL ANALYSIS OF PROPOSED SYSTEM

The RNS system has greater advantages such as high-speed computation, in built parallel nature, immune to power analysis attack and algebraic attack if applied in arithmetic operations involving larger integers [26]-[27]. Therefore, the proposed system is decided to test with respect to the speed of computations on blowfish algorithm. Because in each round of blowfish algorithm it contains two 32-bit XOR and one Feistel function which intern contains two 32-bit addition and two 32-bit XOR operations. That is a total of (16 rounds × (2-XOR+F (2-XOR+2-ADDITION)) +2-XOR) 98 numbers of 32-bit operations for a single 64-bit block of data to encrypt. Therefore, it has been decided to apply the proposed method in the 32-bit addition of Feistel function.

That is 32-bit modulo additions of blowfish algorithms are replaced by the proposed RNS method in which input to the addition modulo are decomposed using the proposed RNS and the smaller components are processed in parallel. The result is analyzed with respect to the time elapsed for the standard blowfish algorithm v/s the proposed method of blowfish algorithm.

The proposed design for the blowfish addition modulo operation is show in the Fig. 2. The Fig. 2 shows the output of two S-boxes are decomposed into smaller 6-bit components using RNS forward conversion. These components (corresponding residues) are added in parallel and later combined based on table driven method to generate 32-bit equivalent number using RNS backward conversion.

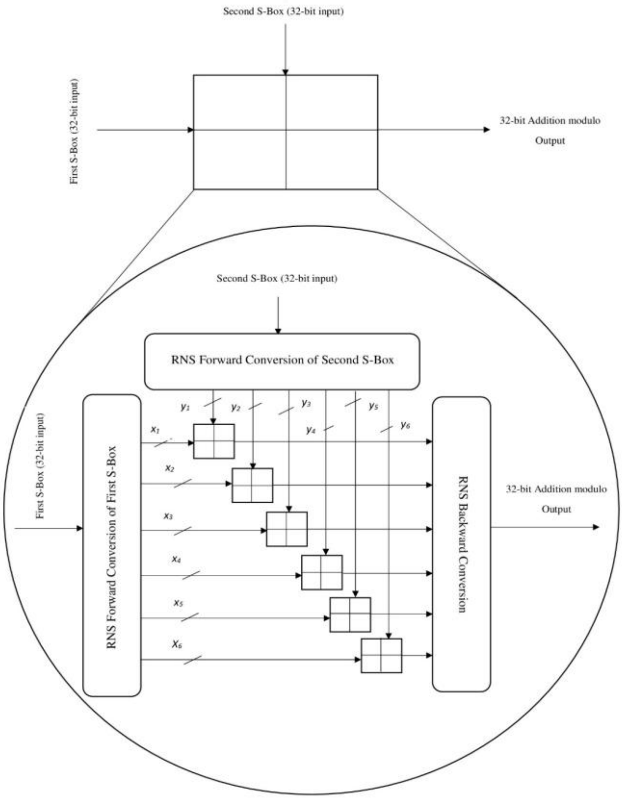
Second S-Box (32-bit input)
First S-Box (32-bit input)
32-bit Addition modulo Output
Second S-Box (32-bit input)
RNS Forward Conversion of Second S-Box
First S-Box (32-bit input)
RNS Forward Conversion of First S-Box
y_1
y_2
y_3
y_4
y_5
y_6
x_1
x_2
x_3
x_4
x_5
x_6
RNS Backward Conversion
32-bit Addition modulo Output

Fig. 2. Proposed model for blowfish addition modulo 32.

It is best understood if considered an example. Therefore, an example is discussed to realize the addition modulo using the proposed model as follows:

Let us assume k=4, such that from equation (7) and (8),

$$q_k = 2^k - 1 = q_4 = 2^4 - 1 = 16 - 1 = 15; \; p_k = (2q_k + 2) = (2 \times 15 + 2) = 32.$$

With respect to the above parameters the six moduli's as per the equation (10) with pair vice relatively prime are

$$m_1 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) - 1 = 6 \times \left(\frac{(15 - 1)}{2}\right) - 1 = 6 \times 7 - 1 = 42 - 1 = 41$$

$$m_2 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) + 1 = 6 \times \left(\frac{(15 - 1)}{2}\right) + 1 = 6 \times 7 + 1 = 42 + 1 = 43$$

$$m_3 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) + 3 = 6 \times \left(\frac{(15 - 1)}{2}\right) + 3 = 6 \times 7 + 3 = 42 + 3 = 45$$

$$m_4 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) + 5 = 6 \times \left(\frac{(15 - 1)}{2}\right) + 5 = 6 \times 7 + 5 = 42 + 5 = 47$$

$$m_5 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) + 7 = 6 \times \left(\frac{(15 - 1)}{2}\right) + 7 = 6 \times 7 + 7 = 42 + 7 = 49$$

$$m_6 = 6 \times \left(\frac{(q_4 - 1)}{2}\right) + 11 = 6 \times \left(\frac{(15 - 1)}{2}\right) + 11 = 6 \times 7 + 11 = 42 + 11 = 53$$

Now let us assume the X=65535 and Y=42153 are the two 32-bit values, these are to be added. The addition is such that the residues of X are computed as per the equation (13) as:

$$x_1 = 65535 \; mod \; 41 = 17; \; x_2 = 65535 \; mod \; 43 = 3; \; x_3 = 65535 \; mod \; 45 = 15;$$
$$x_4 = 65535 \; mod \; 47 = 17; \; x_5 = 65535 \; mod \; 49 = 22; \; x_6 = 65535 \; mod \; 53 = 27$$

Similarly, from equation (13) the residues of Y are computed as:

$$y_1 = 42153 \; mod \; 41 = 5; \; y_2 = 42153 \; mod \; 43 = 13; \; y_3 = 42153 \; mod \; 45 = 33;$$
$$y_4 = 42153 \; mod \; 47 = 41; \; y_5 = 42153 \; mod \; 49 = 13; \; y_6 = 42153 \; mod \; 53 = 18$$

Now the result of the addition modulo of corresponding component moduli x_i and y_i are computed as $c_i = (x_i + y_i) mod \; m_i$ are computed in parallel based on the equation (14) as:

$$c_1 = 17 + 5 \; mod \; 41 = 22; c_2 = (3 + 13) \; mod \; 43 = 16; c_3 = (15 + 33) \; mod \; 45 = 3;$$
$$c_4 = 17 + 41 \; mod \; 47 = 11; \; c_5 = (22 + 13) mod \; 49 = 35; \; c_6 = (27 + 18) \; mod \; 53 = 45$$

Finally, the total sum (C) (the addition modulo 32-bit) is computed by combining all the component residues (c_i) by the forward RNS weighted conversion as follows using equation (15).

$$C = (22 \times 236184165 \times 18 + 16 \times 225198855 \times 3 + 3 \times 215190017 \times 8 + 11 \times 206032995 \times 43 + 35$$
$$\times 197623485 \times 26 + 45 \times 182708505 \times 46) mod \; 9683550765$$

$$C = (7.65001E + 11) \; mod \; 9683550765$$
$$C = 107688$$

This is equivalent to addition modulo of 32, that is if added without proposed model

$$(65535 + 42153) mod \; 2^{32} \; = 107688$$

All six moduli's are operated in parallel, this indicates the time taken for the additive operation is, the additive operation for the maximum moduli that is addition modulo 97 (m6). This can be shown experimentally as follows.

To have a practical insight an experimental result is tested on an environment that contains Intel(R) CORE(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz with 4 GB installed RAM. The result is recorded with respect to the time elapsed for processing both proposed method of blowfish and existing standard of blowfish.

The result of the test carried out are recorded and analysed separately for encryption and decryption. For the consistent output, the test is repeated for fifty times and the average of the fifty repetitions are recorded and used for the analysis. The time taken for processing both the method for various file size is shown table-I.

Table-I: Average time between the proposed method and standards for Encryption/Decryption Blowfish

Encryption Time analysis			Decryption Time analysis	
File Size	AVG Time in seconds elapsed using the existing method	AVG Time elapsed using the proposed method	Time elapsed using the existing method	Time elapsed using the proposed method
10MB	2.724338	2.417677	0.002506	0.002322
20 MB	5.375882	4.739947	3.653145	3.351455
30 MB	8.200386	6.40815	5.396083	4.11357
40 MB	10.60201	8.228327	7.537973	5.837602
50 MB	13.50485	9.816842	9.053618	6.657303
60 MB	16.55609	11.630451	11.50914	8.930177
70 MB	20.71033	15.379581	14.36431	11.25126
80 MB	24.36797	17.713692	15.79634	12.26789
90 MB	27.85867	21.046451	17.74162	13.62862
100 MB	31.61525	24.6149229	22.71033	17.85033

From the test and analysis it's noted that the encryption / decryption time for standard blowfish algorithm is more compared to the blowfish algorithm with the proposed RNS. As file size increases the difference in the time also increases. The same results (table-I) are used to show the difference between the encryption time and decryption time for both methods (proposed v/s standard blowfish), that is shown in Fig. 3.

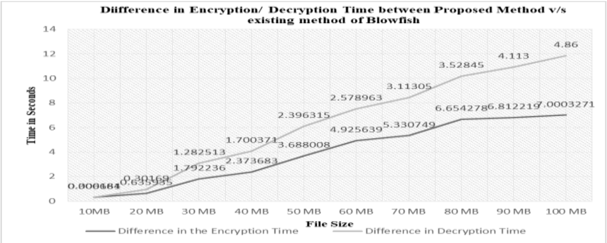

Fig. 3. Difference in time between the proposed method v/s Standard methods for Encryption/Decryption Blowfish

The grey line in the figure 3 represents the difference in the encryption time between the standard blowfish algorithm whereas the dark line denotes the difference in decryption time between the proposed method v/s standard blowfish. The result shows as the file size increase the time difference between both the methods increases, indicating better the time complexity. Therefore, the proposed RNS plays a better role in optimizing the time complexity of the any processing system. This is advantageous in the mission critical systems where time complexity is utmost important. Another importance is its in-built parallel processing capability that leads to increased immunity to side channel attack, algebraic attack and power analysis attack. Because the values are processed in its smaller components in parallel enabling to dissipate actual algebraic relationship between the values and also lowered power consumption. Therefore, the proposed RNS is bests suitable where it's the security is important aspects of data exchange.

V. CONCLUSION

In this paper a new approach of RNS to decompose a large integer into smaller components were proposed. The forward conversion of the proposed RNS is created based on the six moduli sets $6\left(\frac{(q_k-1)}{2}\right)-1, 6\left(\frac{(q_k-1)}{2}\right)+1, 6\left(\frac{(q_k-1)}{2}\right)+3, 6\left(\frac{(q_k-1)}{2}\right)+5, 6\left(\frac{(q_k-1)}{2}\right)+7, 6\left(\frac{(q_k-1)}{2}\right)+11$ and the RNS for the backward conversion is proposed. In-built nature of the proposed RNS design leads to the higher computation speed and is tested using a standard blowfish algorithm. When proposed RNS system is used, then the total time required for the modular arithmetic operation is the maximum time required to process the largest moduli of the RNS (i.e. $6\left(\frac{(q_k-1)}{2}\right)+11$). Hence, the total time of operations (modular arithmetic operations) is directly proportional to the largest residue computed in the forward conversion. The experimental analysis shows the optimization of the time complexity by the application of proposed RNS. The proposed RNS system provides immunity to the side channel attack, algebraic attack and power analysis attack because of parallel processing of the components residues.

REFERENCES

1. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "Efficient Reverse Converter Designs for the New 4-Moduli Sets 2𝑛−1,2𝑛,2𝑛+1,22𝑛+1−1 and 2𝑛−1,2𝑛+1,2𝑛,22𝑛+1 Based on New CRTs," IEEE Trans. Circuits Syst. I Regul. Pap., vol. 57, no. 4, pp. 823–835, Apr. 2010.
2. E. K. Bankas and K. A. Gbolagade, "A residue to binary converter for a balanced moduli set 22𝑛+1−1,22𝑛,22𝑛−1", in: Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on, 2013, pp. 211–216.
3. K. A. Gbolagade, R. Chaves, L. Sousa, and S. D. Cotofana, "Residue-to-binary converters for the moduli set 22𝑛+1, 22𝑛, 2𝑛−1," in 2009 2nd International Conference on Adaptive Science Technology (ICAST), 2009, pp. 26–33.
4. Y. Liu and E. Lai, "Moduli set selection and cost estimation for RNS based FIR filter and filter bank design," Trans. Des. Aut. Embed. Syst., vol. 9, no. 2, pp. 123–139, 2004.
5. S. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," IEEE Trans. Computers, vol. 43, no. 1, pp. 68–77, Jan. 1994.
6. Y. Wang, "Residue-to-binary converters based on new chinese remainder theorems," IEEE Trans. Circuits Syst. II, Analog Digital Signal Process,, vol. 47, no. 3, pp. 197–205, Mar. 2000.
7. Y. Wang, X. Song, M. Aboulhamid, and H. Shen, "Adder based residue to binary number converters for (2/sup n/-1, 2/sup n/, 2/sup n/+1)," IEEE Trans. Signal Process., vol. 50, no. 7, pp. 1772–1779, Jul. 2002.
8. W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y. Wang, "A high-speed residue-to-binary converter and a scheme for its VLSI implementation," in Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on, 1999, vol. 6, pp. 330–333 vol.6.
9. A. Hiasat and A. Sweidan, "Residue number system to binary converter for the moduli set (2n- 1, 2n- 1, 2n+ 1)," J. Syst. Archit., vol. 49, no. 1–2, pp. 53–58, 2003.
10. A. Hariri,K.Navi, and R. Rastegar, "A newhigh dynamic range moduli set with efficient reverse converter," Trans. Comput. Math. Appl., pp. 660–668, Apr. 2008.

11. B. Cao, C.-H. Chang, and T. Srikanthan, “An efficient reverse converter for the 4-moduli set based on the new Chinese remainder theorem,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 10, pp. 1296–1303, Oct. 2003.
12. L. Sousa and S. Antão, “MRC-based RNS reverse converters for the four-moduli sets and,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 59, no. 4, pp. 244–248, Apr. 2012.
13. A. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, “Efficient reverse converter designs for the new 4-moduli sets and based on new CRTs,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 823–835, 2010.
14. A. Skavantzos and T. Stouraitis, “Grouped-moduli residue number systems for fast signal processing,” in Proc. Int. Symp. Circuit Syst., 1999, vol. 3, pp. 478–483.
15. B. Cao, C.-H. Chang, and T. Srikanthan, “A residue-to-binary converter for a new five-moduli set,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 5, pp. 1041–1049, May 2007.
16. A. Skavantzos, M. Abdallah, T. Stouraitis, and D. Schinianakis, “Design of a balanced 8-modulus RNS,” inProc. IEEE16th Int. Conf. Electronics, Circuits, Syst., 2009, pp. 61–64.
17. A. Skavantzos, “An efficient residue to weighted converter for a new residue number system,” in Proc. Great Lakes Symp. VLSI, 1998, pp. 185–191.
18. A. Hiasat, “VLSI implementation of new arithmetic residue to binary decoders,” IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 13, no. 1, pp. 153–158, Jan. 2005.
19. F. J. Taylor, “Residue arithmetic: A tutorial with examples,” Computer, vol. 17, pp. 50–62, May 1984.
20. Li Xiao and Xiang-Gen Xia, “Error Correction in Polynomial Remainder Codes with Non-Pairwise Coprime Moduli and Robust Chinese Remainder Theorem for Polynomials”, IEEE Transactions on Communications, DOI: 10.1109/TCOMM.2015.2400997.
21. N. B. Chakraborti, John S. Soundararajan, and A. L. Reddy, “An Implementation of Mixed-Radix Conversion for Residue Number Applications”, IEEE Transactions on Computers, Vol. C-35, No. 8, August 1986.
22. G. A. Jullien et al., "Hardware realization of digital signal processing elements using the residue number system," Presented at IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Hartford, CT, May 9-11, 1977.
23. W. K. Jenkins and B. J. Leon, "The use of residue number systems in the design of FIR digital filters," IEEE Trans. Circuits Syst., vol. CAS-24, pp. 191-201, Apr. 1977.
24. W. K. Jenkins, "Techniques for residue-to-analog conversion for residue-encoded digital filters-," IEEE Trans. Circuits Syst., vol. CAS-25, pp. 555-562, July 1978.
25. A. Z. Baraniecka and G. A. Jullien, "Residue number system implementation of number theoretic transforms in complex residue rings," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-28, pp. 285-291, June 1981.
26. Ganesh Aithal, K.N. Hari Bhat and U.Sripathi, “Implementation of Stream Cipher System Based on Representation of Integers In Residue Number System”, 2010 IEEE 2nd International Advance Computing Conference, pp. 210-217, 2010.
27. Abdelhamid S. Abdelhamid and Ahmed A. Belel, “Secure Transmission of Sensitive Data Using Multiple Channels” IEEE international Conference on Computer System Application 3rd ACS 6th January 2005.
28. NEAL KOBLITZ “A Course in Number theory and Cryptography” Springer Verlog New York 1987 page 11.
29. Donald E Knuth, “The art of computer programming-semi numerical algorithms”, published by Dorling Kindersley (India) Pvt. Ltd., licensees of Pearson Education in south asia, 3rd edition, volume 2.

AUTHORS PROFILE

Mr. Radhakrishna Dodmane, Associate Professor,Dept. of CSE NMAM Institute of Technology, Nitte. He has completed his BE in CSE, masters in CSE and pursuing his PhD in the area “Cryptography and Network Security” from the University of VTU karnataka. His areas of interests are security and data communication. He has about 16 years of teaching experience. Published around nine papers in the international journals and conferences. Received the best paper award for one of his paper presented in the international conference organized in association with springer.

Dr. Ganesh Aithal, Professor and Vice-Principal, SMVI Technology and Management. Had completed his BE in Electrical Power, masters in Digital Electronics (M.Tech) and PhD in Electronics and Communication. He has published more than 30 papers in various international journal and conferences. His areas of interests are Cryptography and Network Security, Data Mining and Business Analytics. He has guided two PhD students and currently five students are pursuing under his guidance. He has book chapter under his name. He has about 32 years of teaching and research experience.

Dr. Surendra Shetty, Professor and Head, Dept. Of MCA, had completed his B.Sc. in 2001 and Master of Computer Applications during 2004. Dr. Surendra Shetty had been awarded his doctoral degree for his research work “Audio Data Mining Using Machine Learning Techniques” in 2013 from university of Mangalore. He has published more than 25 research papers in different international journals and conferences. He is currently guiding six research scholars. Dr. Surendra Shetty authored two book chapters in different publications entitled “Machine Learning Approach for Carnatic Music Analysis” and “Applications of Unsupervised Techniques for Clustering of Audio Data”. He has received research grant of 20 lakhs from VGST (GoK) for carrying out research on “Automatic Natural Language Processing and Speech Disorder Problems in Kannada Language”. He has 15 years of teaching experience. The Research areas of interest are Cryptography, Data mining, Pattern Recognition, Speech Recognition, MIS, Software Engineering and Testing.