ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 1 Issue 3 December 2023 / Page No: 56-62
Paper Id: IJACT-V1I3P106 / Doi: 10.56472/25838628/IJACT-V1I3P106

Original Article

Implementation and Performance Comparison of the Decryption Computation Variants in the Multi-Prime Rivest, Shamir and Adleman (RSA) Algorithm

TOPE-OKE Adebusola. M[1], AWEH Opani[2], BABALOLA Gbemisola[3], DABO Laura[4]
[1,2,3,4]*Computer Science Department, Afe Babalola University, Ado Ekiti, Nigeria.*

Received Date: 28 October 2023 Revised Date: 23 November 2023 Accepted Date: 16 December 2023

Abstract: *This study performed a performance analysis of multi-prime RSA decryption variations' applicability for various system applications. The goal of the research was to determine how suitable the various decryption variations are for different application systems with varying system requirements. The object oriented analysis and design (OOAD) approach was used to create an application that implements the decryption variants, which was then implemented in the Java programming language. In the decryption phase, the application was used to encrypt files using three prime numbers and decrypt separate files using the Aryabhatta Remainder Theorem (ART), Garner method, and Gauss algorithm.*

Keywords: *Algorithm Comparison, Decryption, Multiprime, RSA.*

I. INTRODUCTION

Data security has become known as one of the most essential needs of the information age. Besides, it is a vital subject in guaranteeing safe diffusion of data in the course of the internet [1]. It is the deterrence and protection of workstation assets from unlawful admittance, and other threats while still allowing lawful users to maneuver data without restraint. Cryptography is the subdivision of data security which makes up the study of algorithms and protocols that secure data [2]. The security of data can be achieved through cryptography which serves as the basis for most of the Information Technology (IT) security solutions. Consequently, it can be said that cryptography is an evolving expertise, which is essential for network security and plays an integral role in our society [1].

Rivest, Shamir and Adleman (RSA) algorithm is particularly most commonly used and well-defined cryptographic algorithm [3][4]. Therefore it has attracted keen interest among researchers. Despite the fact that RSA is popularly used, a major weakness identified by a previous study is its decryption mechanism [5]. RSA is grounded on arithmetic modulo of big numbers, which requires large number of computations; however, speedy implementation of RSA has become fundamentally essential for the performance of suits of cryptographic algorithms [6]. The use of large key size had become a challenge especially when the system is working on small device or heavily loaded systems. This had resulted to demand of the cryptosystem with fast computations.

Over the years however, diverse adaptations of RSA have been put forward to micro manage the size of the key and concurrently offer the same level of security as the original RSA. These adaptations include Batch RSA, Rebalanced RSA, Multi-power RSA, Multi-prime RSA and Dual RSA. RSA is based on two prime numbers while multi-prime RSA is based on more than two prime numbers. Multi-prime RSA has been future to speed up RSA implementations; this became necessary due to the speed of computers being manufactured now days [7].

The Chinese Remainder Theorem (CRT) was proposed in the early 1980s to speed up RSA decryption. However, Rao and Yang [10] argued that Aryabhatta Remainder Theorem (ART), the Garner's algorithm and the Gauss' algorithm could give a faster computation time since the rising performance of the computing hardware, the sizes of the keys of the cryptographic algorithms are increasing to provide a more secure data transfer. Data Encryption Standard (DES) was popularly used but is no longer considered a secure algorithm because of its relatively short key size of 56 bits [8]. For RSA algorithm, the key sizes have ranges from 256 to 512, 1024 and 4096 bits even though larger keys are directly proportional to slower decryption. RSA algorithm prompts any two large prime numbers, x and y, and then compute this composition number, n, where n=x*y [3]. It is easy to multiply two huge prime numbers collectively to get the product but computationally hard to do the reverse [9]. The increase of prime numbers will give rise to an increased prime numbers multiplied results, while the number of decomposition in the synthesis of increased difficulty for the prime number [9]. Boneh and Shacham[6] proposed the use of Multi-prime RSA to speed up decryption and the use of Chinese Remainder Theorem. It has been observed that the Chinese Remainder Theorem is not fast enough [10]. However, there are other

residual number theorem such as Aryabhatta Remainder Theorem (ART), Garner's algorithm and Gauss algorithm which are believed to provide faster computation time.

This study therefore aims to implement all three algorithms and establish which of these three multi-prime RSA algorithm variants provides better computation time and hence more efficient decryption.

A. Aim and Objectives of the Study

This study's main aim is to determine the most efficient decryption computation for the multi-prime RSA decryption variants. This aim will be achieved through the following objectives:

- To implement each of the variants –. Aryabhatta Remainder Theorem (ART), Garner's algorithm and Gauss algorithm in an object-oriented programming language.
- To test each of these three multi-prime RSA algorithm variants in the same environment under the same conditions.
- To find out the algorithm with the fastest computation time.

B. Review of related Works

Islam et al.[11], proposed an enhanced and modified approach of RSA cryptosystem based on "n" distinct prime number using two different public key and private key generated from the large factor of the variable "N" and performed a double encryption-decryption operation. The proposed modified RSA (MRSA) algorithm was implemented in Java 8, all calculation and performance analysis were performed using four large prime numbers and examined on varying bit sizes of input. Their study revealed that the system is more secure than the standard RSA and the time needed to break the system is high because of the extra complexity added in factoring the components of the public key exponent. The limitation of this approach is that it will not work properly unless "n" distinct prime numbers are considered. Islam et.al [11] approach is similar to the approach in this study since both studies involve more than two prime numbers. The difference between both studies however is that our study adopted the double encryption-decryption operation, that is, the public key exponent and the private key exponent consists of three components (e, f, N) and (d, g, N) respectively, where "e", "f", "d" and "g" were randomly taken.

A comparable study by [12] showed the algorithms that work well with modified multi RSA cryptosystem. These included Euler's Totient, Chinese Remainder Theorem and Fermat's Little Theorem. The study discussed the efficiency and security of using Multi prime RSA cryptosystem. Findings from the study revealed that multi-prime is more efficient in the decryption phase when implemented using CRT and it is more secured since the use of multi prime numbers increase the level of difficulty to break the security of the algorithm.

[13], proposed an alternate method to the existing Chinese Remainder Theorem (CRT) to solve congruity in the decryption stage of Rebalanced RSA and RPrime RSA (which is a combination of Multi-prime RSA and Rebalanced RSA). The method is called the Aryabhatta Remainder theorem, and they implemented the method using Java programming language on a credit card data set provided by Data Trans. Three comparisons were made and the findings reveal that using ART has improved the overall decryption speed of RPrime and Rebalanced RSA and also RPrime RSA exhibits a better speed gain than Rebalanced RSA. The similarities with their work and ours is the use of ART to improve the decryption speed in variants of RSA and a assessment of the performance is made with varying bit length and fixed bit lengths.

[14] However, modified the standard RSA algorithm by using 3 prime numbers instead of the 2 prime numbers used in standard RSA and used Cantor's pairing algorithm to merge multiple data units into a single data unit by generating only a single integer number for messages which are sent to the receiver. [14] Work revealed that this approach provided a better security compared to the standard RSA and reduced the average time taken for sending the data from sender to receiver. This reviewed system is similar to this work since they are both aimed at reducing computation time. Another study by [15] applied the sieve Function methodology for the key generation process and CRT for the decryption phase. The sieve function is used to eliminate the randomly chosen prime candidates having very small factors thereby increases the speed process of prime number generation. The findings reveal that the time needed for decryption decreases with each additional prime in the modulus and the space required decreases with each additional prime added to the modulus. This work is quite similar to the approach in consideration in that both used CRT for a faster decryption phase.

Agrawal et al. [16], used an approach named modified RSA (MRSA) algorithm and through this improved the speed of decryption procedure of Standard RSA (SRSA) algorithm in which extended Euclidean algorithm is used to calculate the value of decryption key implemented using MATLAB. The findings divulge that there is better execution time and processing time when compared to standard RSA algorithm and its limitation is that plain texts are being represented by numerical

values only. [16] Work is quite similar to the work of Islam et al. [11] in terms of developing better and faster decryption key generation algorithm so that overall decryption time could be reduced.

C. RSA Algorithm

Cormen et al.[17] define an algorithm as some definite computational process that takes a set of values as input and produces a set of values as output. The major characteristics of an algorithm as identified by [18] include finiteness, correctness and efficiency which are determined by time and memory space. Time efficiency points to how fast the algorithm runs, while space efficiency centers on the extra memory the algorithm uses [18]. The analysis of an algorithm focal point is on efficient system resource usage. In this study, Aryabhatta Remainder Theorem (ART), Garner's algorithm and Gauss algorithm will be compared and analysed for secure data transfer in terms of decryption time only.

a) Aryabhatta Remainder Theorem (ART)

Rao and Yang formulated this theorem based on the Aryabhatta algorithms proposed by Pearce and Kak[19][20]. The Aryabhatta algorithm is used in place of CRT given the advantage of lesser number of inverse calculations [10].

Theorem: Let p_1 and p_2 be relatively prime moduli and P= p_1· p_2. Given x mod p_1=x_1 and x mod p_2=x_2, x has one and only one solution in Z_p given by

$$x = ART(x_1, \quad x_2; \ p_1, \quad p_2; \ P)$$
$$= ART(0, \ c; \ p_1, \ p_2; \ P) + x_1, \text{Where } c = (x_2 - x_1) \ mod \ p_2$$
$$= A + x_1 \text{ where } A = p_1[(c \cdot p_1^{-1}) \ mod \ p_2]$$

b) Garner algorithm

Garner constructed an algorithm which is generally used to solve the Chinese Remainder problem to convert the residue code of a number $X = (v_1, v_2, \dots, v_t)$ with respect to relatively prime modulo $m_1, m_2, \dots, m_t$ to a mixed radix number with weight $l, mt, m_{t-l}, m_t, \dots, m_2 m_3 .. m_t$. This algorithm helps in calculating modular multiplicative inverse [21].Garner's algorithm is an efficient method for determining *x*, $0 \leq x < M$, given $v(x) = (v_1, v_2, \dots, v_t)$ the residues of *x* modulo the pair wise co-prime moduli $m_1, m_2, \dots, m_t$. The radix equivalent can be calculated using these weights.

c) Gauss's algorithm

The solution x to the simultaneous congruencies in the Chinese remainder theorem may be computed as $x = \sum_{i=1}^{k} a_i N_i M_i \ mod \ n$, where $N_i = n/n_i$ and $M_i = N_i^{-1} \ mod \ n_i$. These computations can be performed in $O((\lg n)^2)$ bit operations. t inverse operations and a modular reduction with modulo M is requires for CRT when the standard Gauss algorithmis considered. The number of bit operations is $O(t^2 k^2)$ where k is the maximum bit size of the residues [22].

II. MATERIALS AND METHODS

A performance comparison of three decryption algorithms was developed based on one-tier architecture to give a standalone application. It has all the layers in a single software package.

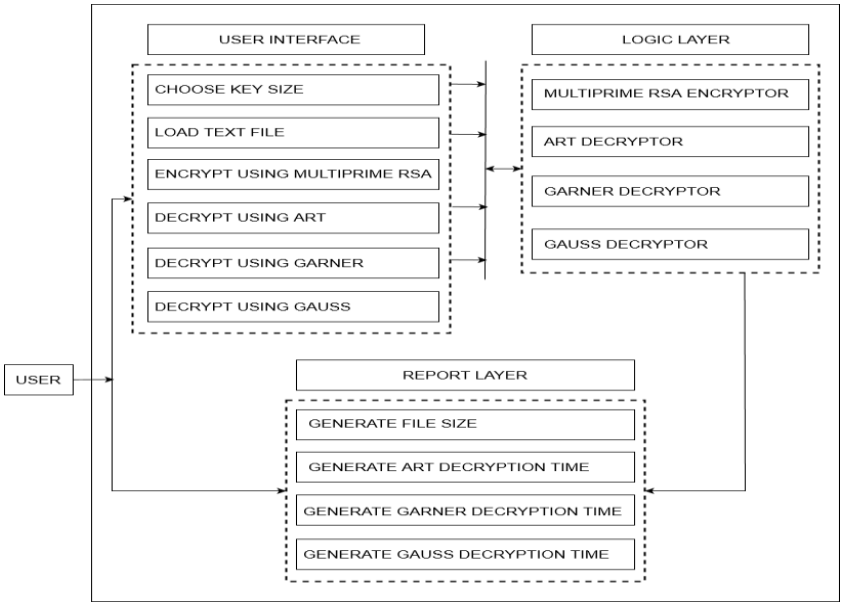

Figure 1: System Architecture

The system has three basic layers: the user interface layer, logic layer and the data layer. The user interface contains the buttons that enable the user to load, encrypt, and decrypt a text file. The logic layer implements the algorithms for RSA algorithm encryption and decryption of text files while the data layer gives information on the size of file and the time taken to decrypt a secret message text. Figure 2.1 summarises the system architecture. Object-Oriented Design (OOD) approach was used to design the system. The conceptual model produced in object oriented analysis took account of the constraints imposed by the system architecture such as transaction throughput, response time, run-time platform, development environment, or programming language. The concepts in the analysis model were mapped onto implementation classes and interfaces. All the algorithms have been implemented in NetBeans Integrated Development Environment (IDE) version 8.2 and graphical interface in JavaFX Scene Builder. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, and Solaris. The system development and testing were carried out simultaneously. However this testing was aimed at ensuring that all the components being coded and put together worked effectively. After all the components had been put together and the entire system was developed. Figure 2.2 explains the way data flows and how each state interacts with each other. It shows the sequence of activities that happens when the user clicks on the decrypt button. The encrypted text file which was converted into array of bytes is converted back to the original message but in byte arrays with the time taken for each operation.

A. The Software Interface

The software interfaces shows what happens when the user loads the text file with the size of the file and the encrypted file clicking the encrypt button with the time taken to encrypt the file. It also shows the decrypted files (that is, the uploaded file) with each of the time taken for each decryption (see figure 2.2).

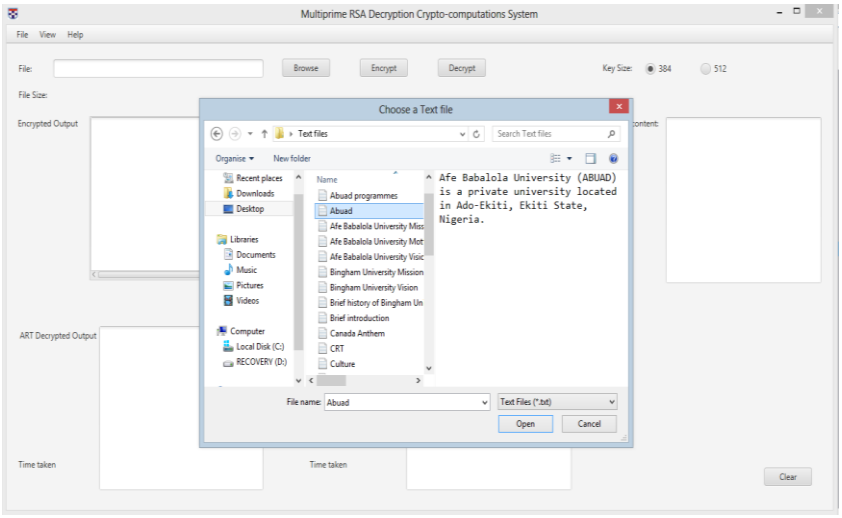

Figure 2: How to Load Text File Using File Chooser Window

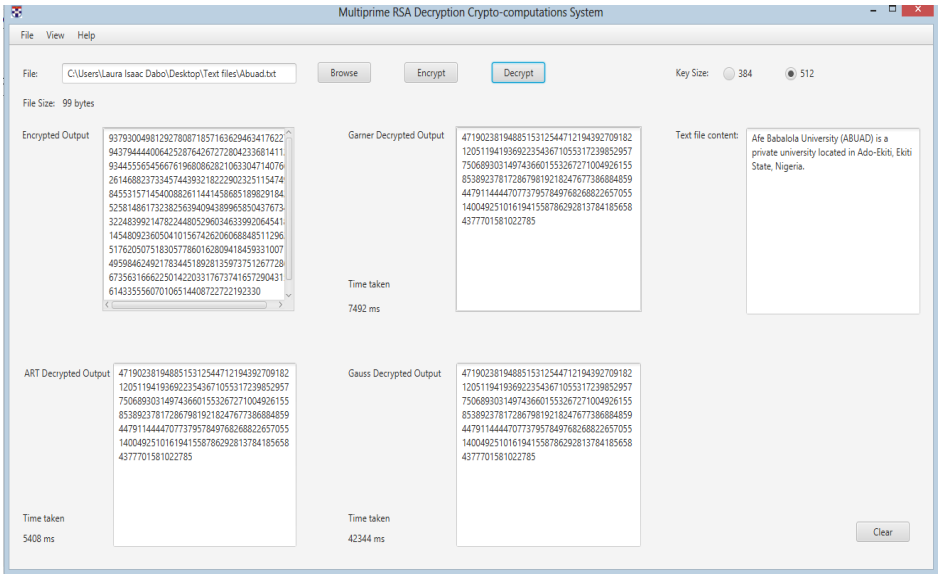

Figure 3: Encrypted and Decrypted Files (512bits)

Figure 3 shows that a text file content that says "Afe Babalola University(Abuad) is a private University located in Ado –Ekiti, Ekiti State, Nigeria" was loaded into the Multiprime RSA decryption crypto computation system. It also displayed the 512 bits Encryption- decryption of all three algorithms (Gauss, Garner and ART) Section 2 shows the result, calculations and performance analysis of the algorithms base on their output.

B. Result: Parameters for Assessment Calculation

The following are the parameters with which the performances of algorithms were calculated: File size, computation time (Encryption), computation time (Decryption), throughput.

a) File size:

The memory space required by each of the three algorithms is determined on the basis of input data size. The smaller the memory used by each algorithm better the task performance.

b) Computation time (Encryption):

The encryption computation time is the time taken for each of the three algorithms to convert a plaintext to a cipher text.

c) Computation time (Decryption):

The decryption computation time is the time taken by the algorithms (ART, Garner and Gauss) to decode the plain text from the cipher text. The decryption time was used to calculate the decryption throughput of the algorithms.

d) Throughput:

Throughput of the decryption algorithms is calculated by dividing the total file sizes in kilobytes on total decryption time for each algorithm in seconds.

$$Throughput = \frac{Total\ file\ sizes}{Total\ decrypt`ion\ execution\ time} = \frac{\sum file\ sizes}{\sum DET}$$

C. Result: Performance Analysis

The timing results shown in tables 1 and 2 were obtained by using the system time. The laptop encrypts different file sizes. The application was developed in Java programming language and methods were written to collect data on some performance metrics: computation time (encryption), computation time (decryption) and file size.

The computation time specifies the speed of encryption and decryption time. This is considered the time that multiprime RSA algorithm takes to produce a cipher text from a plaintext and the time that ART, Garner and Gauss algorithm takes to produce a plaintext from cipher text respectively. Decryption time indicates the speed of decryption and it is used to calculate the throughput of a decryption scheme.

Table 1: Timing Results with 384 Bits

	File Size (Bytes)	ART Decryption Time (ms)	Garner Decryption Time (ms)	Gauss Decryption Time (ms)
	28	2,914	3,640	18,044
	76	3,064	3,950	18,424
	95	2,800	3,850	17,835
	99	3,062	4,199	17,631
	130	3,049	3,946	17,386
Total	428	14,889	19,585	89,320

Table 2: Timing Results with 512 Bits

	File Size (bytes)	ART Decryption Time (ms)	Garner Decryption Time (ms)	Gauss Decryption Time (ms)
	28	2,824	3,598	19,882
	76	5,832	7,372	41,323
	95	5,231	7,522	42.560
	99	5,408	7,492	42,344
	130	5,641	7,119	40,729
Total	428	24,936	33,103	186,838

Tables 2.1 and 2.2 represents the five (5) different sizes of files, corresponding encryption execution time in milliseconds and corresponding decryption execution time taken by ART, Garner and Gauss algorithms in milliseconds respectively. By analyzing and comparing table 2.1 and 2.2, it can be concluded that the decryption time taken by ART is relatively faster as compared to Garner while decryption is very slow in Gauss.

2.4 Result: Decryption Throughput

Calculating the decryption throughput we define;

$$Decryption\ throughput = \frac{\sum file\ sizes}{\sum decryption\ execution\ time\ (DET)}$$

For 384 bits,

ART Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/14889$

$= 0.02875 = 2875 \times 10^{-5}$KB/sec.

Garner Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/19585$

$= 0.02185 = 2185 \times 10^{-5}$KB/sec.

Gauss Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/89320$

$= 0.004792 = 479.2 \times 10^{-5}$KB/sec.

For 512 bits,

ART Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/24936$

$= 0.01716 = 1716 \times 10^{-5}$KB/sec.

Garner Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/33103$

$= 0.01293 = 1293 \times 10^{-5}$KB/sec.

Gauss Decryption throughput $= \sum file\ sizes\ / \sum DET$

$= 428/186838 = 0.002291 = 229.1 \times 10^{-5}$KB/sec.

Key Size	Decryption algorithm	Decryption Throughput (KB/sec)
384	ART	2875×10^{-5}
	Garner	2185×10^{-5}
	Gauss	479.2×10^{-5}
512	ART	1716×10^{-5}
	Garner	1293×10^{-5}
	Gauss	229.1×10^{-5}

By examining the throughput result with 384 bits, it shows that the decryption speed of ART is high as compared to Garner and Gauss algorithms. Based on this comparison performed on ART, Garner and Gauss algorithm, the result shows that ART outperformed Garner and Gauss algorithms in throughput which is measured in kilobyte per seconds (KB/Sec). Examining the throughput result with 512 bits, it shows that the decryption speed of ART is high as compared to Garner and Gauss algorithms. Based on this comparison performed on ART, Garner and Gauss algorithm, the result shows that ART outperformed Garner and Gauss algorithms in throughput which is measured in kilobyte per seconds (KB/Sec). Table 2.3 shows the decryption throughput of the three algorithms at a glance.

III. CONCLUSION AND RECOMMENDATION

Cryptographic algorithms are the foundation for a secure data transmission, especially when communicating through an unbounded network like the Internet. Bearing this in mind the implementation and a comparison of the RSA algorithm variants – ART, Garner and Gauss was done. It was found that ART is the most efficient of the three algorithms. The result in this study is therefore aimed at guiding researchers, corporate organizations and government in selecting a data decryption algorithm that is suitable to their data security needs. The suitability of these three algorithms to a particular application or data exchange should be based on the specific security requirements of the system bearing in mind the ever increasing security threats to data and information exchange.

As a suggestion for further studies, this study focused on the computational speed only, it is therefore recommended that new studies should take into consideration the CPU usage and the security threats that may be associated with these algorithms.

IV. REFERENCES

[1] S. Kumar,. Review on Network Security and Cryptography. International Transaction of Electrical and Computer Engineers System, vol 3 No.1, 1–11. 2015. https://doi.org/doi:10.12691/iteces-3-1-1.

[2] Dent, A. W. (2006). Fundamental problems in provable security and cryptography, Information Security Group. Retrieved from https://eprint.iacr.org/2006/278.pdf

[3] Rivest, R. L., Shamir, A., andn Adleman, L. . A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, vol.26, No1,1983, pp 96–99. https://doi.org/10.1145/357980.358017

[4] S.Robinson,. Safe and secure: data encryption for embedded systems. EDN Europe, vol. 53 No. 6,.2008, PP 24-33.

[5] D. Calloway,. Introduction to cryptography and its role in network security: Principles and practices. 2009. Retrieved from http://www.dancalloway.com/.

[6] D. Boneh,. and H. Shacham, . Fast variants of RSA, Cryptobytes, vol 5. No.1, 2002,1-9. RSA Laboratories.

[7] N. Ojha, and S. Padhye, . Cryptanalysis of Multiprime RSA with Secret Key Greater than Public key, International Journal of Network Security, vol.16 No.1, 2014 ,pp53-57.

[8] J. Andress, (2011) . The Basics of Information Security., https://doi.org/10.1016/c2010-0- 68336-2

[9] RSA Laboratories . RSA Laboratories" Frequently Asked Questions About Today"s Cryptography, Version 4.1, 2000, RSA Security Inc.

[10] T.R.N.Rao, & Yang, C.H. (2006) Aryabhatta Remainder Theorem: Relevance to Public-Key Crypto-Algorithms, Circuits, Systems and Signal Processing, vol.25 No.1, 2006, pp1-15. https://doi.org/10.1007/s00034-005-1123-6

[11] M.A. Islam, Md. A. Islam, N. Islam, and B. Shabnam. A modified and secured RSA public key cryptosystem based on "n" prime numbers. JCC Vol.6, 2018, 78-90. https://doi.org/10.4236/jcc.2018.63006

[12] M. G. Kamardan, N. Aminudin, N. Che-Him, S. Sufahani, K. Khalid, and R. Roslan. Modified Multi Prime RSA Cryptosystem. in Journal of Physics: Conference Series, 995, 012030. 2018 https://doi.org/10.1088/1742- 6596/995/1/012030

[13] C. Padmaja, V. Bhagavan, and B. Srinivas, using Aryabhatta Remainder Theorem to Decrypt a Message with RPrime and Rebalanced RSA. International Journal of Engineering & Technology, vol.7 No. 2.7, 2018, pp758-762. https://doi.org/10.14419/ijet.v7i2.7.10940

[14] H. Sahay . Modified RSA Cryptosystem. International Journal of Innovative Science, Engineering and Technology, vol. 4 No.1, 2017 pp249-253. Retrieved from http://www. http://ijiset.com/vol4/v4s1/IJISET_V4_I01_33.pdf

[15] A.Tariq, A. Bharti, D. Kumawat, and S. Naz. A framework for generating multi prime RSA using sieve function, International Journal of Engineering Technology Science and Research, 4(11), 2017. Retrieved from https://www.ijetsr.com/

[16] M. Agrawal, B. Pal, and R. Maheshwari. Improvement over public key cryptosystem RSA by implementing new decryption key generation algorithm. International Journal of Engineering and Management Research, 5(6), 2015, pp300-304. Retrieved from http://www.ijemr.net

[17] T. Cormen, C. ,Leiserson, R. Rivest, and C. Stein. Introduction to algorithms. Cambridge, Massachusetts: MIT Press. 2009.

[18] L. Anany. Introduction to the design & analysis of algorithms. Boston ; Montr al Pearson Addison-Wesley 2012.

[19] I Pearce,., cited: Indian Mathematics: Redressing the Balance. 2002. Retrieved October 26, 2019, from mathshistory.st-andrews.ac.uk website: http://wwwhistory.mcs.st-andrews.ac.uk/history/Projects/Pearce/index.html

[21] H. Garner. The residue number system. IRE Transactions on Electronic Computers, vol. 8 No. 2, 1959 PP140– 147.

[22] A. J Menezes,., and S.A.Vanstone. Handbook of applied cryptography. Boca Raton: Crc Press.2001.