Analytical Comparison of RSA and RSA with Chinese Remainder Theorem

Ankur Mantri[1], Abdul Razaque[2], Hiral Makwana[3], Parita Parekh[4], Tariq Rahim Soomro[5]

[1,3,4]*Electrical Engineering & Computer Science Department, Cleveland State University, Cleveland, OH, USA.*
[2]*Computer Science Department, New York Institute of Technology, Nanjing Campus, China.*
[5]*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Dubai, UAE.*

[1]a.g.mantri@vikes.csuohio.edu
[2]arazaque@nyit.edu
[3]hiralmak4eva@vikes.csuohio.edu
[4]paritaparekh@vikes.csuohio.edu
[5]tariq@szabist.ac.ae

Abstract—RSA encryption algorithm is one of the most powerful public key encryption algorithm. The problem with RSA algorithm is that RSA decryption is relatively slow in comparison to RSA encryption. Chinese Remainder Theorem (CRT), a modulo based mathematical theorem, is proposed by researchers as a way to enhance the performance of decryption. CRT minimizes the mathematical computation to large extent, thus improving the speed. CRT is well known for improving RSA's decryption speed, but it has some drawbacks which limits its usage. The problem is that the limitations or drawbacks of CRT are not highlighted. The goal of this research paper is to address disadvantages of CRT when used for RSA decryption. Apart from the social and economic impacts, this paper covers the effects of research on current cryptographic protocols used by different browsers as well as organizations. In order to achieve goal, we are comparing several factors between RSA decryption with RSA-CRT decryption. We are using JAVA programming language to analyze the decryption algorithms. The significance of this research paper is to serve as the foundation for further research on RSA-CRT decryption. In addition, this paper addresses situations where CRT decryption is faster and beneficial to use by stating its advantages and disadvantages.

***Keywords*—**Decryption, RSA, CRT, Performance, Time, Overhead, Key Length

I. INTRODUCTION

The security of data transferred over the internet is getting more significant due to the rise in e-business, internet banking, online bill payment, etc. It is necessary to ensure secure connection while transmitting sensitive information [1], like credit card information, bank account details, etc., over the internet. There are several security measures taken to ensure safe data transfer. One of the approaches is to apply cryptography. Cryptography refers to the data transformation in a way that is not in user readable format, called encryption. After encryption we get cipher text and that can be transmitted over the network securely. At the receiver end, a reverse operation called decryption is performed to get the original message or data sent by the user. There are two types of encryption; [2], symmetric and asymmetric encryption.

Symmetric key encryption uses the same key for encryption and also decryption. Asymmetric key encryption uses both private key and public key. Public key is used for encrypting data whereas private key is used for decrypting data. Out of all the various asymmetric key algorithms, RSA is most robust, secure and widely used algorithm. An alternate approach of Chinese Remainder Theorem (CRT) for RSA decryption can be used [3]. If we look at the history of Chinese Remainder Theorem, its foundation was laid in China by ancients back in 2nd century. Later Suanjing [4] did more research on it and founded the Chinese Remainder Theorem, also known as Sun Zi Theorem.

The main reason for using Chinese Remainder Theorem with RSA decryption is that public key algorithms, as per paper [5], are secure but generally include complex mathematical operations. The Chinese Remainder Theorem explained in [6], is proven to have the decryption speed increased by a factor of four [7]. It is a mathematical theorem, that can be explained in simple words as, it determines a whole number x, which is when divided by given divisors, leaves given remainders as explained in reference [8]. Therefore, there are unique solutions that exists for each pair of coprime numbers. Coprime numbers are the numbers that have 1 as their Greatest Common Divisor (GCD).

The Chinese Remainder Theorem replaces one modular exponentiation with two which are half-size modulus and exponents. The benefit of using Chinese Remainder Theorem is that, each of them is almost eight times faster than any other non-CRT exponentiation.

The motivation for this paper is to determine the faster way to decrypt RSA as it is probably the most widely used public key algorithm and almost everyone who uses the web is affected by it knowingly or unknowingly.

Our research methodology requires setting up an experimental environment to conduct the several tests in order to prove the effectiveness of the RSA algorithm and Chinese remainder theorem.

The rest of the paper is organized as follows: Section 2 explains about problem that is addressed by this paper and its significance. Section 3 describes the related work in the area of research followed by performance analysis of RSA and RSA-CRT in section 4. Section 5 discusses about the experimental results. The section 6 discusses the practical and theoretical implications of the study and finally, the conclusion is given in section 7.

II. PROBLEM IDENTIFICATION AND SIGNIFICANCE

The RSA algorithm includes encryption and decryption that is way more secure and robust than any other similar algorithm. The only flaw in it is that it is very time consuming when it comes to decryption. The decryption in RSA algorithm uses raising exponent to dth power and d is taken as large bit prime number. Thus, larger mathematical calculation will take more time for decryption. As an alternate to this, the Chinese Remainder Theorem is used as a way to reduce the mathematical calculations to a great extent. Despite of its strength it has some drawback when used for RSA decryption. The problem is, these issues are not yet addressed clearly. This research paper will discuss them in detail.

This research has significant social and economic impact. Our research helps the community to have a stronger and faster cryptographic algorithm with fewer resource consumption. It lays out the advantages and disadvantages of CRT, which is a proposed decryption algorithm for RSA in many papers and is still a topic that needs research.

III. RELATED WORK

The Chinese Remainder Theorem is well-known for its use with RSA. The research work has done to determine suitability of Chinese Remainder Theorem for RSA decryption. It is important to understand RSA algorithm and application in the field of cryptography. These concepts are well explained in [2]. The research work done in [4] explains RSA cryptosystem based on CRT with implementation. Their paper shows some computed figure from their implementation but did not show any disadvantages of their design. In order to have a better understanding of this research paper, we have provided some background about previous work. [5] describes the history of Chinese Remainder Theorem that how it was originated in ancient times and came into existence later. One of the major problem for Chinese ancients was the 'remainder' while making calendar. Later that problem was handled and solved by Chinese mathematician Sun Zi Suanjing.

CRT is also named after this mathematician. This paper had no mention of RSA but it is very informative and it helped us understand why we can use CRT to decrypt RSA. Generation of private key and public key is the heart of RSA algorithm. [9-10] provides information about strengths and weakness of private key and public key that could be utilized in further research. Although it didn't mention anything about RSA decryption, it contained useful information about the overall advantages of a public key encryption. [9] talks about fast RSA decryption using CRT. They are comparing RSA with and without CRT with implementation in JAVA. Paper describes comparison of RSA with and without CRT in a nice way leaving out any shortcomings for RSA with CRT. The CRT is all about mathematical modulus functions that was explained with simple numbers for easy understanding in [11-12]. This helps to understand complex mathematical modules in easy way that can be further used in decryption of RSA.

As seen, previous research work in area of CRT covers new implementations to prove that CRT is good alternative to RSA decryption, but it does not touch area where CRT is lacking and is not suited. This research could be used as foundation for further research as it will provide information about the shortcomings of CRT or any drawbacks of CRT when used for RSA decryption. Detailed study, extending existing work and comparison would help us to satisfy the goal of this research paper.

IV. PERFORMANCE ANALYSIS OF RSA DECRYPTION AND RSA-CRT

This research focuses on measuring the performance of RSA and RSA-CRT algorithm. Some of the performance factors to be taken into consideration while measuring are discussed as follows. First factor is the key length value. It is very important factor to measure the performance since length of the key is highly responsible for the performance of an algorithm. Second factor is encryption ratio. The performance of an algorithm highly depends on the amount of data to be encrypted. The encryption ratio factor is used to measure the amount of data to be encrypted in order to measure its performance. Third factor is computational speed. This factor is used to measure the computational speed of an algorithm to check if they are fast enough. Last factor is overhead. It refers to indirect computation time, memory, and other resources required to get the output.

A. Formulation of RSA

There are a few mathematical steps involved to decrypt RSA encryption. This is illustrated in figure 1.

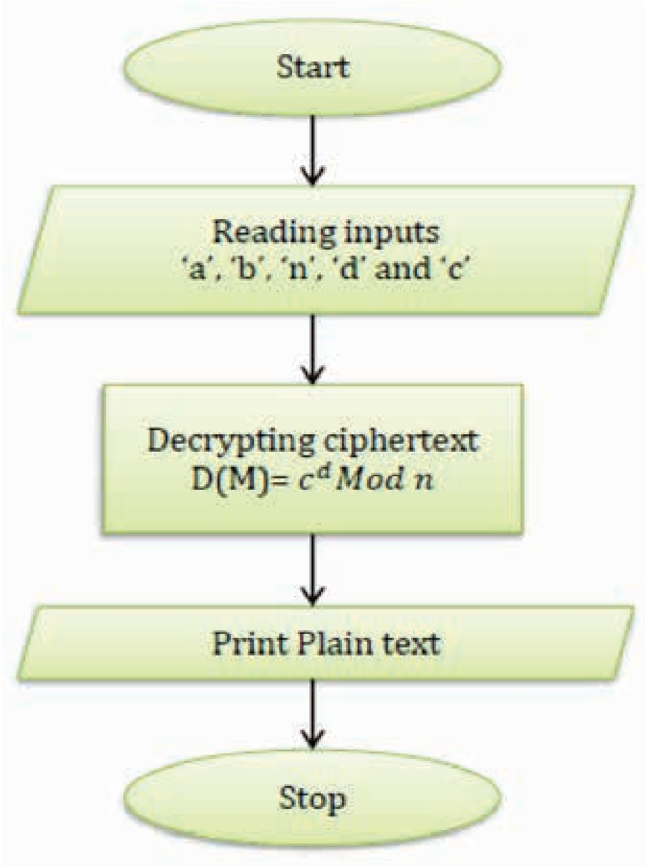

Fig. (1). Decryption flowchart for RSA

Selecting any random numbers a and b, we calculate 'n', that is the multiple of the two prime factors, we assume 'a' and 'b'. Function of n (φ (n)) is calculated to be the product (a-1) and (b-1).

$\phi(n) = (a-1)(b -1)$ (1)

After calculating φ (n), we select any number which is a coprime of φ (n) and is greater than one, (let's say that number is stored in a variable 'f' here). Also, f should be less than φ (n) and greater than or equal to 1.

$1< = f < \varphi (n)$ (2)

'd' can be calculated as the modular multiplicative inverse of (f Mod φ (n)), where the mathematical equation is given as

$f * d \bmod (p-1)(q-1)=1$ (3)

So,

d = multiplicative inverse of (f Mod φ (n))

Ciphertext ('c' here) can be decrypted using 'd' and 'n' as calculated above by the function below, where D(M) is the decrypted text.

$D(M) = cd \text{ Mod } n$ (4)

B. Formulation of RSA-CRT

RSA decryption, as discussed earlier, is slower than its decryption. This can be improved by bringing an alternative to the existing algorithm or by developing similar new algorithm. CRT is an alternate approach to RSA decryption to increase the decryption speed. RSA decryption uses one modular exponentiation which is replaced by two in case of RSA-CRT. Moreover, each replaced modulus and exponents are half in size in CRT and it is eight times. Thus with the use of CRT, RSA decryption speed can be increased up to four times. In addition to this, smaller size of modulus and exponents reduces computational time and resource consumption [9].

To understand basic concept of CRT we assume that x1, x2, ···, xk are positive integers and it is pairwise coprime. Next, let us consider set of integers b1, b2, …, bk. For a set of given integers there is always an integer I that satisfies the simultaneous congruence

I = b1 (mod x1)
I = b2 (mod x2)
...
I = bk (mod xk)

All 'I' have congruent modulo to X, where X = x1 · x2 · ·· ·xk.

If and only if I ≡ z (mod X), we get, I ≡ z (mod xi), for 1 < i < k.

RSA-CRT decryption uses similar inputs as RSA. It is explained here in detail with equations as well as using flowchart in figure 2.

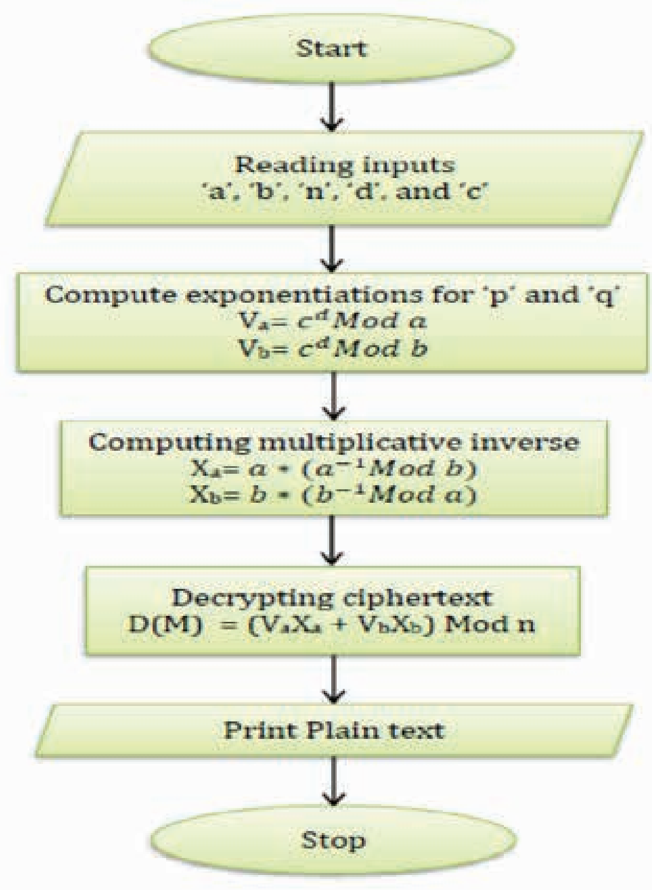
Start
Reading inputs
'a', 'b', 'n', 'd', and 'c'
Compute exponentiations for 'p' and 'q'
$V_a = c^d Mod\ a$
$V_b = c^d Mod\ b$
Computing multiplicative inverse
$X_a = a * (a^{-1} Mod\ b)$
$X_b = b * (b^{-1} Mod\ a)$
Decrypting ciphertext
$D(M) = (V_aX_a + V_bX_b)$ Mod n
Print Plain text
Stop

Fig. (2). Decryption flowchart for RSA with CRT

Let's consider two prime numbers 'a' and 'b' that is used in basic RSA decryption. This 'a' and 'b' is used in conjunction with 'n' and 'd', where 'n' and 'd' is also calculated similar to RSA decryption. Cipher text 'c' is obtained from RSA encryption. Then CRT is applied, we get two equations for two prime numbers.

$$Va = cd \text{ Mod } a \tag{5}$$

$$Vb = cd \text{ Mod } b \tag{6}$$

Then we compute X_b and X_a as a part of CRT,

$$Xa = a * (a - 1 \text{ Mod } b) \tag{7}$$

$$Xb = b * (b - 1 \text{ Mod } a) \tag{8}$$

Above computed V_a, V_b, X_a, and X_b is used to decrypt cipher text D(M), which is a plain text, by using following equation.

$$D\,(M) = (VaXa + VbXb) \text{ Mod } n \tag{9}$$

By using above stated steps for decryption, computational speed can be significantly reduced. Other performance factors can also be improved which are discussed in detail in experimental results section 5.

V. EXPERIMENTAL RESULTS

The limitations and drawbacks of RSA using Chinese Remainder Theorem can be determined by Analytical comparison between RSA and RSA-CRT decryption algorithms. Implementation of RSA-CRT decryption is done from java based code in [6]. Several test cases are created based on different scenarios including different message lengths, and different key sizes for message decryption etc. These test cases are used to test and compare performance and security factors of both RSA and RSA-CRT. Parameters shown in table 1 were used to carry out the test cases.

Table 1. System Configuration.

Software Tools:	Java Eclipse Kepler.
Operating System	MAC OS X El Capitan; Version 10.11.1
RAM	16GB
Processor	2.8 GHz, Intel Core i7
Memory	16GB 133MHz DDR3
Graphics	Intel HD 3000 512 MB
Disk storage	SATA 750GB

In experimental results, focus on the following parameters.

- Decryption time
- Overhead for key generation

Remaining part of the section describes these parameters with graphs.

A. Decryption time

Decryption time is a time that an algorithm takes to decrypt the cipher text. As key length increases decryption time taken by both algorithm increases gradually. Figure 3, reveals that RSA-CRT decryption time is almost 3 times faster as compared to RSA decryption time.

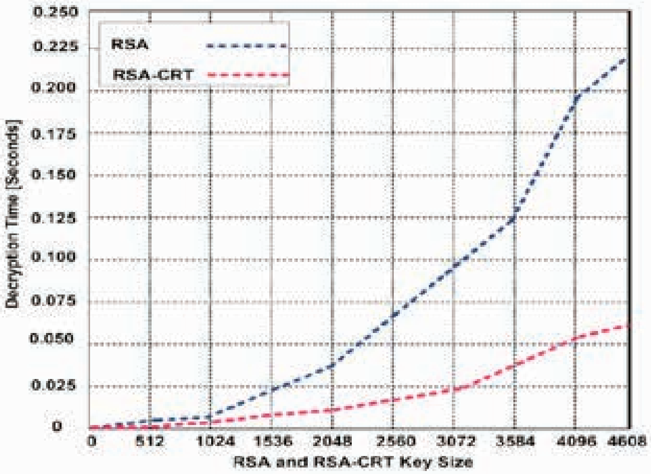

Fig. (3). decryption time of RSA and RSA-CRT

B. Overhead for key generation

Overhead is the time taken to generate the keys. Figure 4 reveals that RSA's overhead are less as compared to RSA-CRT's overhead, when the key length is less than 2048. When the key length is larger than 2048, RSA-CRT's overhead is less than RSA's overhead.

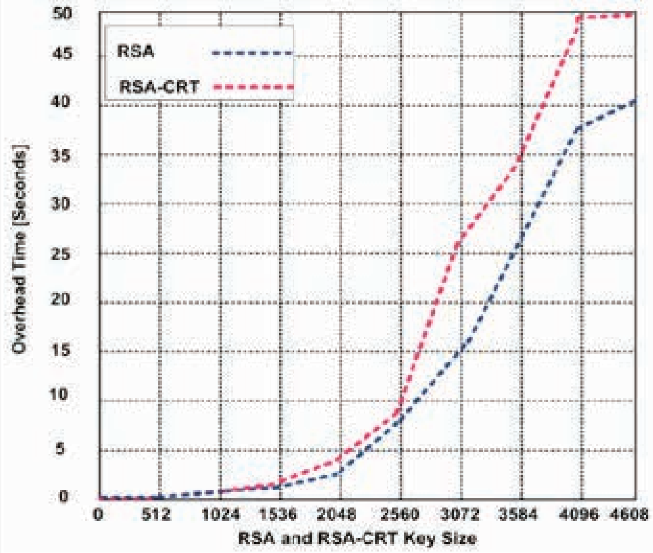

Fig. (4). Overhead time of RSA and RSA-CRT

Table 2 shows comparison of RSA and RSA-CRT after combining the decryption time with the overhead and shows the total time (in seconds) taken by both implementations (RSA and RSA-CRT). The results show that the RSA-CRT implementation is faster when the key length is 1024 and 2048, whereas, RSA implementation is faster for key lengths 3072 and 4096.

Table 2. Comparison Table.

Total time by RSA	Total Time by RSA-CRT	Key Length
0.4298	0.412	1024
4.5438	3.64	2048
20.556	26.424	3072
37.3486	49.369	4096

VI. PRACTICAL AND THEORITICAL IMPLICATION OF STUDY

Having presented the comparison's major statistical discoveries in the previous section, the following paragraph serves to critically discuss the discoveries and originates the theoretical and practical implications. The major analytical comparison results are briefly discussed in the previous section in order to augment the readability and understanding. In this study, we attempted to respond the number of research questions related to how secure data transmission is understood, and performed. The secure data transmission is one the biggest challenges over the Internet particularly for business purposes. We have highlighted the strength and weakness of the RSA and RSA-CRT. Furthermore, we addressed in which conditions, either RSA or RSA-CRT should be used for secure data transmission. Our main goal in this study is to address the overhead time and decryption time for RSA or RSA-CRT.

Many test cases are generated based on different scenarios including varying message lengths, key sizes for message decryption etc. As, these test cases are used to validate and compare performance and security features of both RSA and RSA-CRT. The experimental results confirmed that RSA-CRT takes minimum decryption time and overhead time as compared with RSA. Based on obtained results, we recommend to replace the RSA with RSA-CRT for reducing the decryption and overhead times. As, the results obtained through this practical implication will make the faster and secure data communication.

VII. CONCLUSION

Analytical comparison of RSA and RSA-CRT decryption algorithms is designed and implemented to determine the shortcomings of RSA-CRT decryption algorithm. Initial results showed RSA decryption using CRT definitely improves the performance of decryption by up to 3 times. Further analysis shows that it has more overhead as compared to RSA when the key length is larger than 2048. The reason for this overhead could be the use of more variables, as seen in figure 2, therefore memory utilization by RSA-CRT is greater than RSA. In addition to above analysis, this research brought deeper understanding of working mechanism of existing Chinese Remainder Theorem along with its usage in RSA. By identifying drawbacks of RSA-CRT, this research will serve as a base requiring for future research work to improve the RSA decryption speed. Further investigations and experiments could result in to betterment of the current internet world in terms of adding more robustness without affecting its security.

REFERENCES

[1] A. Razaque and S. S. Rizvi. "Triangular data privacy-preserving model for authenticating all key stakeholders in a cloud environment." *Computers & Security*, vol. 62, pp: 328-347, 2016.

[2] S. Burnett and S. Paine. *The RSA Security's Official Guide to Cryptography*. McGraw-Hill, Inc., 2001.

[3] J. Großsch¨adl. "The Chinese remainder theorem and its application in a high-speed RSA crypto chip." In *Proceedings of 16th Annual Conference Computer Security Applications (ACSAC'00),* 2000, pp. 384-393.

[4] C-H. Wu, J-H. Hong and C-W. Wu. "RSA cryptosystem design based on the Chinese remainder theorem." In *Proceedings of the 2001 Asia and South Pacific Design Automation Conference,* 2001, pp: 391-395.

[5] S. Kangsheng. "Historical development of the Chinese remainder theorem." *Archive for history of exact sciences,* vol.38, no. 4, pp: 285-305, 1988.

[6] S. Singh and G. Agarwal. "Use of Chinese Remainder Theorem to generate random numbers for cryptography." *International Journal of Applied Engineering Research,* vol. 1, no. 2, pp: 115-123, 2010.

[7] S. Iftene and F. Chelaru. "The general Chinese remainder theorem." *International Journal of Computing,* vol. 6, no. 1, pp: 44-50, 2014.

[8] S. Lee, D. Choi and Y. Choi. "Improved Shamir's CRT-RSA Algorithm: Revisit with the Modulus Chaining Method." *ETRI Journal,* vol. 36, no. 3, pp: 469-478, 2014.

[9] M. Blumenthal. "Encryption: Strengths and Weaknesses of Public-key Cryptography." *CSRS*, pp: 1, 2007.

[10] G. N. Shinde and H. S. Fadewar. "Faster RSA algorithm for decryption using Chinese remainder theorem." In *Proceedings of International Conference on Computational & Experimental Engineering and Sciences,* 2008, vol. 5, no. 4, pp: 255-262.

[11] S. Iftene. "General secret sharing based on the Chinese remainder theorem with applications in e-voting." *Electronic Notes in Theoretical Computer Science,* vol. 186, pp: 67-84, 2007.

[12] B. Lynn. *The Chinese Remainder Theorem* [Online]. Available: https://crypto.stanford.edu/pbc/notes/numbertheory/crt.html