A Project Report

on

Efficient Variant of RSA Cryptosystem

carried out as part of the course Cryptography and Security (CC 1501)

Submitted by

Kartik Singhal ***Karthik Agarwal***

169104047 ***169104046***

V Semester BTech CCE

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer & Communication Engineering

MANIPAL UNIVERSITY
JAIPUR

Department of Computer & Communication Engineering,
School of Computing and IT,
Manipal University Jaipur,
November 2018

CERTIFICATE

This is to certify that the project entitled " **Efficient Variant of RSA Cryptosystem** " is a bonafide work carried out as part of the course ***Cryptography And Security***, under my guidance by ***Kartik Singhal and Karthik Agarwal***, students of ***BTech V semester*** at the Department of Computer & Communication Engineering , Manipal University Jaipur, during the academic semester ***Five***, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer & Communication Engineering, at MUJ, Jaipur.

Place:

Date: Signature of the Instructor (s)

DECLARATION

I hereby declare that the project entitled **Efficient Variant of RSA Cryptosystem** submitted as part of the partial course requirements for the course ***Cryptography and Security,*** for the award of the degree of Bachelor of Technology in Computer & Communication Engineering at Manipal University Jaipur during the ***V, November 2018*** semester, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

Signature of the Student:

Place:

Date:

Abstract

We implement and analyze the standard RSA algorithm, understanding its deficiency and to further optimize it through designing and combining the RSA algorithm with Chinese Remainder Theorem to speed up RSA decryption by an approximate factor of four.
We then compare and analyse the performance of the variant.
The RSA-CRT variant is backwards compatible in the sense that a system using it can interoperate with systems using standard RSA.

Table of Contents

List of figures

List of tables

1. Introduction

The security of information exchanged over the web is getting more critical because of the ascent in e-business, internet banking, online payments, and so on. It is important to guarantee secure connection while transmitting sensitive data, like card credentials, financial balance, and so forth, over the web. There are a few safety efforts taken to guarantee safe information exchange. One of the methodologies is to apply cryptographic algorithms which include encryption and decryption of data.

Symmetric key encryption utilizes a same key for encryption and decryption. Asymmetric key encryption utilizes both private key and public key. Public key is utilized for encrypting data whereas private key is used for decrypting it.

Out of all the different asymmetric key algorithms, RSA is most vigorous, secure and broadly utilized algorithm.

The issue with RSA algorithm is that RSA decryption is generally slow in contrast with RSA encryption. Chinese Remainder Theorem (CRT), a modulo based mathematical theorem, is proposed by specialists and researchers as an approach to improve the performance of decryption. CRT minimises the numerical computation to vast degree, hence enhancing the speed.

The main reason for using Chinese Remainder Theorem with RSA decryption is that public key algorithms, are secure but generally include complex mathematical operations. The Chinese Remainder Theorem is proven to have the decryption speed increased by a factor of four.

2. Requirement Analysis

After study and research, it was essential to determine the requirements of the project.
In order to attempt a requirement analysis, the technical and behavioral factors in the project development were considered.

2.1 Functional Requirements

This project focuses on implementing and measuring the performance of RSA and RSA-CRT algorithm. The performance factors to be taken into consideration while measuring is time taken and memory consumed to decrypt.

A. Formulation of RSA

We review the basic steps constituting the RSA public key system.

Algorithm

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

- n is known as the *modulus*.
- e is known as the *public exponent*.
- d is known as the *decryption exponent*.

B. Formulation of RSA with CRT

RSA decryption, as discussed earlier, is slower than its encryption. 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.*

Algorithm

Consider two prime numbers ‘p’ and ‘q’ that is used in basic RSA decryption. This ‘p’ and ‘q’ is used in conjunction with ‘n’ and ‘d’, where ‘n’ and ‘d’ is also calculated like RSA decryption. Cipher text ‘c’ is obtained from RSA encryption. Then CRT is applied.

$d_p = d \bmod (p-1)$
$d_q = d \bmod (q-1)$
$Q_{inv} = q^{-1} \bmod p$
Now,
$M_1 = c^{dp} \bmod p$
$M_2 = c^{dq} \bmod q$
$h = Q_{inv} (M_1 - M_2) \bmod p$
Decrypt cipher text M, which is a plain text, by using following equation.
$M = M_2 + hq$

Note that d_p, d_q and Q_{inv} can be precomputed which speeds up the decryptions process.

2.2 Non - Functional Requirements

- The application developed is fast in performance and has quick response time.
- It is reliable, bug-free and usable.
- The code is readable and simple to use.

3. Work Done

3.1 Development Environment

Table 1 below shows the configuration of the system on which the project was initiated and developed fully. All the tests and reviews were also done on the same PC and project was coded in Java.

Table 1. System Configuration.

Software Tools: IntelliJ Java IDE.
Operating System: Windows 10 Home
Processor: 2.3 GHz, Intel Core i5
Memory: 8GB DDR3
Graphics: Intel HD 520
Disk storage: SATA 500GB

3.2 Comparison and Experimental Results

The efficiency and advantages of RSA with Chinese Remainder Theorem can be dictated by explanatory correlation among RSA and RSA-CRT decryption algorithms. Execution of RSA-CRT decryption is done from java-based code. A few experiments are made dependent on various situations including distinctive message lengths, and diverse key sizes for message decryption and so on. These experiments are utilized to test and analyse performance of both RSA and RSA-CRT.
Parameters presented in Development environment [Table 1] were used to carry out the test cases.

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 algorithms increases gradually. Figure 1 reveals that RSA-CRT decryption time is almost 3 – 4 times faster as compared to RSA decryption time.

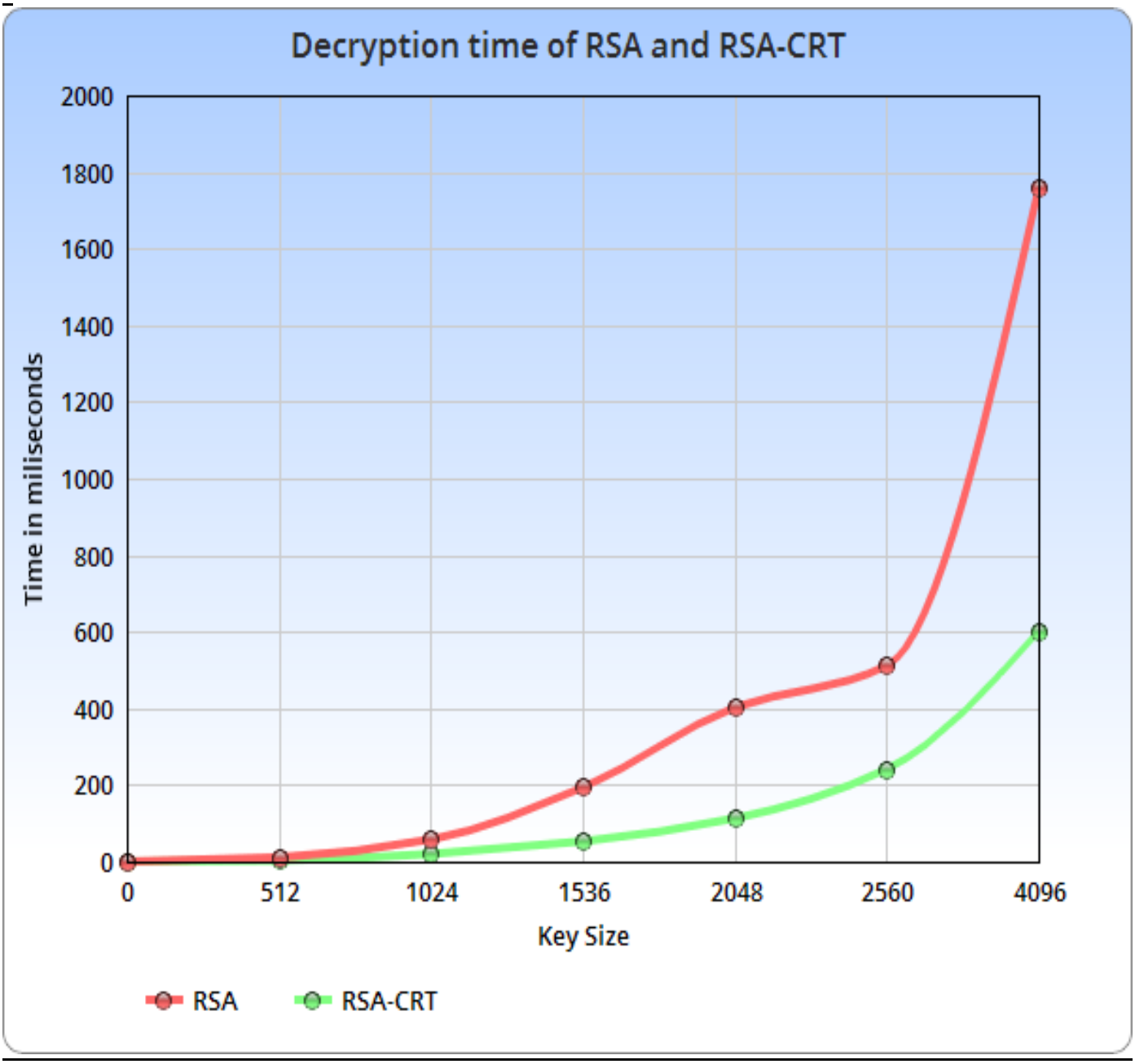

Figure 1. RSA vs RSA-CRT: Decryption time

B. Key Generation time

Overhead time taken to generate the keys. Figure 2 reveals that RSA's overhead is less as compared to RSA-CRT's overhead, when the key length is higher than 1024.

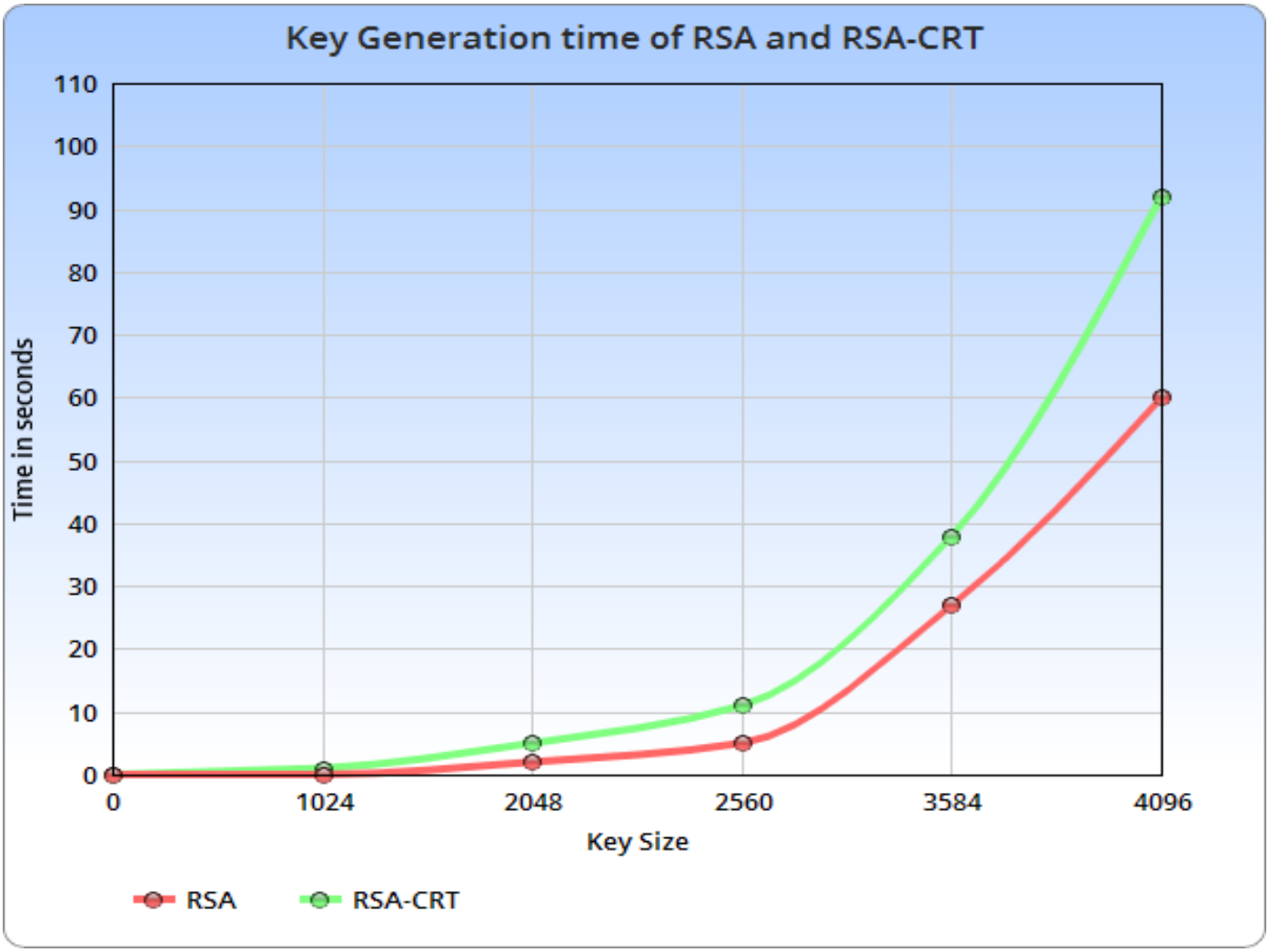

Figure 2. RSA vs RSA-CRT: Key generation time

4.3 Individual Contribution of project members

The RSA implementation was done by Kartik Singhal and CRT implementation was done by Karthik Agarwal. Comparisons and analysis were done by Kartik and, compilations were handled by Karthik. The project was developed together, and all the efforts were made equally.

5. Conclusion and Future

We implemented and analysed the standard and CRT variant of RSA designed to speed up decryption and be backwards compatible with standard RSA.

We found out that RSA decryption using CRT improves the performance of decryption by up to four times. Further analysis shows that it takes more key generation time as compared to RSA when the key length is larger than 1048. The reason for this could be the use of more variables and increase in the computation required during key generation phase of RSA.

In addition to above analysis, this project brought understanding of the Chinese Remainder Theorem along with its usage in RSA.

In future, we would like to create a GUI for the Java code which was coded as a part of the project which will be easier for general users to use.

Further, we would like to survey and research other ways to speed up the RSA public key encryption system.

References

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

2. 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.

3. M. Blumenthal. "Encryption: Strengths and Weaknesses of Public-key Cryptography." CSRS, pp: 1, 2007

4. An efficient variant of the RSA cryptosystem - https://www.ime.usp.br/~capaixao/paper.pdf

5. On the Efficiency of Fast RSA Variants in Modern Mobile Phones-
https://pdfs.semanticscholar.org/0d38/42a5dac514da7aa06ecb4f39 e0b0982b 4cee.pdf

6. RSA- https://simple.wikipedia.org/wiki/RSA_algorithm

7. 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.

(Note – All links last accessed on November 05, 2018.)