Study of Multi-Prime RSA

Surinder Kaur*, Pooja Bhardwaj,Shivani Mankotia

Information Technology Bharati Vidyapeeth's College of Engg, New Delhi, India

Emails : kaur.surinder@bharatividyapeeth.edu; bharadwajp@acm.org; mankotias@acm.org
* Correspondence: kaur.surinder@bharatividyapeeth.edu

Abstract

This paper studies and analyses the encryption and decryption times of a popular variant of the RSA algorithm, the multi-prime RSA. This algorithm uses more than two prime numbers for the encryption process. In this paper, 3, 4, and 5 prime RSA algorithms have been implemented and studied. The rate of increase of encryption and decryption times concerning the number of primes used is also illustrated and compared graphically.

Keywords: RSA algorithm; encryption; decryption; n-prime RSA

1. INTRODUCTION

The standard RSA algorithm is an asymmetric-key cryptographic algorithm. It is a popular and proven concept that is easy and feasible to find and multiply substantially large prime numbers. Still, it is extremely difficult to do the exact reversal, i.e., to factor a large number to get to the original large primes. The RSA algorithm's private and public keys are based on very large prime numbers. The point of focus which determines the success or strength of RSA is the selection and generation of the public and private keys.

RSA focuses on Public-key cryptography and Digital Signatures. Public-key cryptography expunges the idea of sending the keys or copy of keys to the receiver through a secure external channel so that one can decrypt the original message and vice versa. The sender's decryption key and digital signature are used to verify the sender's authenticity. A public encryption key can verify digital signatures, and they cannot be forged or changed later; hence a sender cannot repudiate having signed the message in the future [1]. This makes RSA more reliable, secure, and definitive. It is also a collaborative algorithm since it can be used/combined with other encryption algorithms like DES.

In this paper, we will study the standard RSA and one of its modifications; the n-prime RSA. The rest of the paper includes the following; A brief Literature Review of this algorithm and its variations. Section III is about the standard working of the RSA Algorithm. Section IV discusses the n-prime RSA with comparisons of encryption and decryption times with standard RSA. As the value of n increases, the N-prime RSA becomes less feasible, and these drawbacks are discussed in Section V. The paper is concluded in section VI, which also discusses the future scope

Doi : https://doi.org/10.54216/FPA.010105

Received: December 20, 2019 Revised: April 2, 2020 Accepted: April 30, 2020

2. LITERATURE REVIEW

Security of multi-prime RSA was studied for r>=3, with a small prime difference [2]. Multi Prime RSA with imbalanced prime factors was also analyzed [3]. In 2014, distributed RSA key generation approaches were studied, and those were efficient enough to be used with smartphones. Here also, the key generation was performed using more than two primes [4]. Multi-prime RSA is used to enhance the security of user data by generating private and public keys through randomly generated prime numbers. This RSA acts as a middle, secure layer for web services for securely storing data on the cloud. The user can access the web services by logging in and encrypting the data [5].

The basic idea behind multi-prime RSA was to modify the structure of the RSA modulus. One such approach was patented by Compaq [6], with the modulus of the form N=pqr, which is the 3-prime approach. Another approach was put forward by Takagi [7][8], which uses the moduli as $N=p^2q$, and gives an even better decryption speedup using CRT.

3. RSA ALGORITHM AND ITS IMPLEMENTATION

The standard RSA with two large primes was implemented in Java, and an average value of the encryption and decryption time was obtained for a common string, which was used in evaluating multi-prime RSA as well [9],[14]. There are three major steps in the algorithm:

1. Generation of Public and Private key
2. Encryption
3. Decryption

An overview of these three steps is given below:

A. Key Generation

Key generation consists of the following steps:

- Two large prime numbers, 'p' and 'q' are chosen randomly.
- Calculate value of '*n*' as *n=multiply(p,q)*
- Calculate value of 'φ(n)*'* as φ(n) = *(p-1)*(q-1)*
- Select integer '*e*' so that the GCD (greatest common divisor) of φ(n) and *n* evaluates to 1, i.e., $1<e<\varphi(n)$
- Value of 'd' is calculated using (d * e) % φ(n) = 1
- (e,n) → Public Key and (d,n) → Private key.

This encryption process is illustrated in Fig. 1 [12].

B. Encryption

The plaintext M is encrypted to ciphertext, C using

$C=M^e \bmod n$ M^e *Mod N* $M^e \bmod n$.

C. Decryption

The plaintext is obtained when ciphertext is decrypted using $M= C^d$ *Mod N*. The decryption process is illustrated in Fig. 2 [12].

Doi : https://doi.org/10.54216/FPA.010105

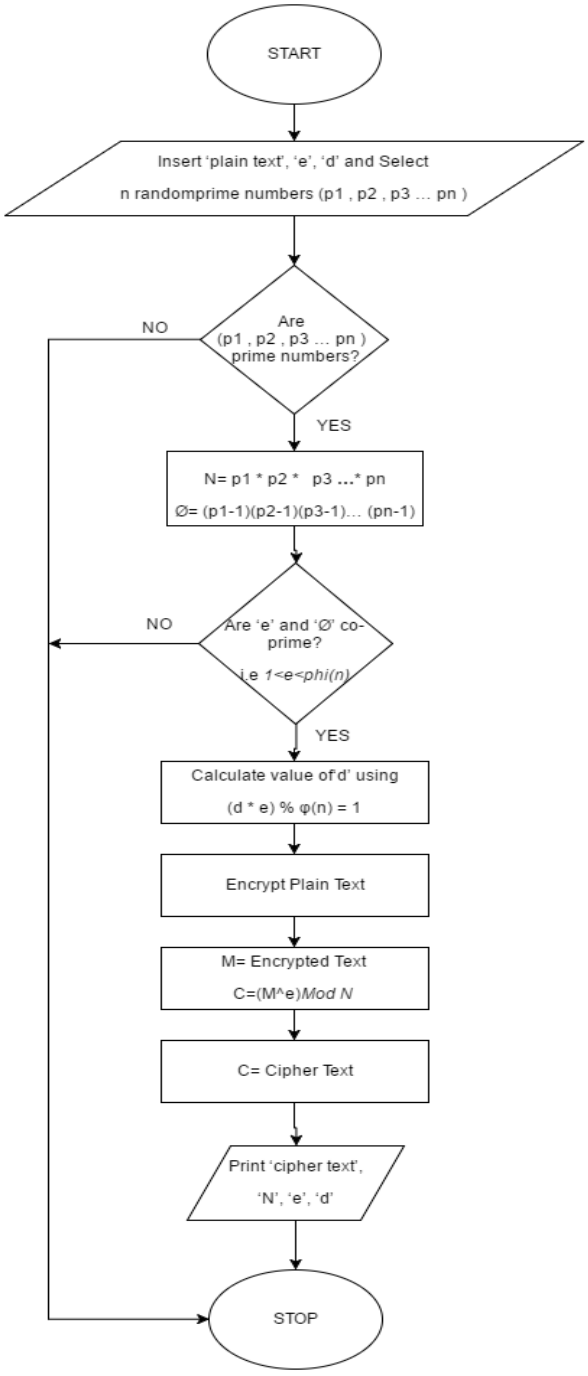

Figure 1: Flowchart for the Multi-prime encryption process.

Doi : https://doi.org/10.54216/FPA.010105

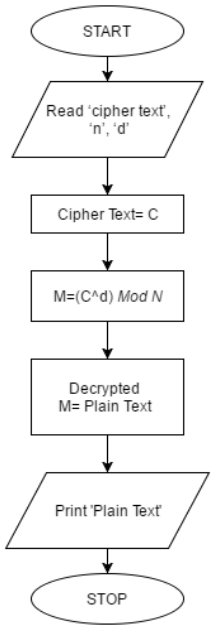

Figure 2: Flowchart for the Multi-prime decryption process.

Results obtained when RSA was implemented in JAVA are average running time for encryption: 11.1 ms, while the average running time for decryption: 23.0 ms

4. MULTI-PRIME RSA

The RSA algorithm can also be implemented using more than two prime numbers. This is referred to as n-prime RSA. Also called the multi-prime RSA, this is simply a general version of the RSA. The following sections cover implementation and the average encryption and decryption times of 3-prime, 4-prime, and 5-prime RSA and differences in cryptanalysis from the standard 2-prime RSA.

The primary aim is to determine if using multi-prime RSA is secure and practical compared to standard RSA. This paper has observed the results generated when 3, 4, and 5 prime numbers are used. We have used a bit length of 1024, and 256 bytes is the block size; the time is recorded in milliseconds (ms).

D. 3-Prime RSA

3 large prime numbers p, q, s, are generated randomly and used to calculate the secret key. The Euler's Totient function, in this case, can be calculated as:

phi= (p-1)(q-1)(s-1)

Average running time for encryption: 18.1 ms

Average running time for decryption: 77.8 ms

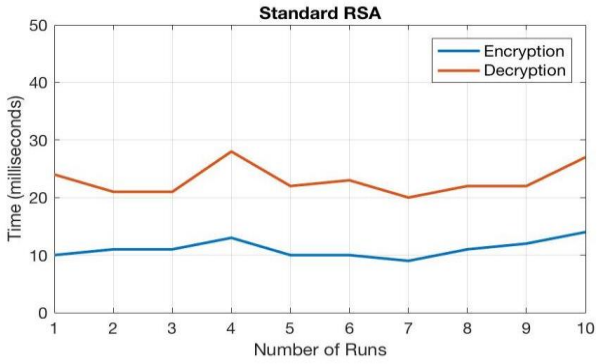

Figure 3: Encryption and decryption times for standard RSA Algorithm.

Doi : https://doi.org/10.54216/FPA.010105

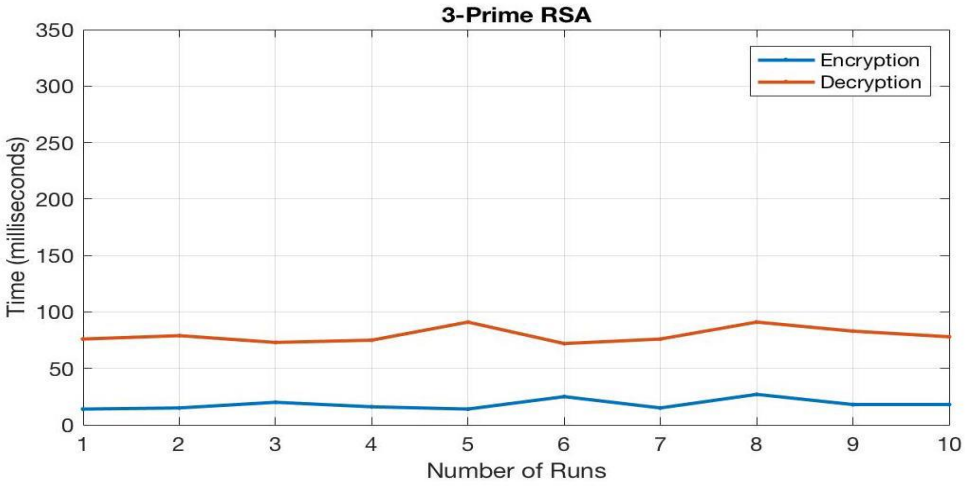

Figure 4: Encryption and decryption times for 3-prime RSA.

E. 4-Prime RSA

4 large prime numbers p, q, s, and t are randomly generated and used to calculate the secret key. The Euler's Totient function, in this case, can be calculated as:
phi=(p-1)(q-1)(s-1)(t-1)

Average running time for encryption: 18.9 ms
Average running time for decryption: 163.7 ms

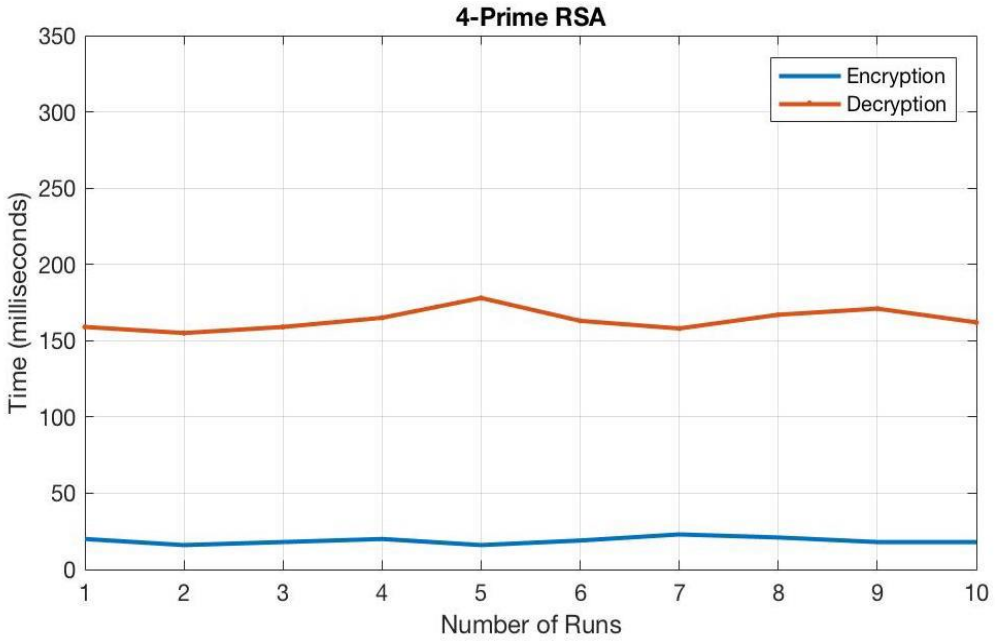

Figure 5: Encryption and decryption times for 4-prime RSA.

Doi : https://doi.org/10.54216/FPA.010105

F. 5-Prime RSA

Five large prime numbers, p, q, s, t, and u, are randomly generated and used to calculate the secret key. The Euler's
phi=(p-1)(q-1)(s-1)(t-1)(u-1)

Average running time for encryption: 24.7 ms
Average running time for decryption: 312.9 ms

TABLE I. AVERAGE ENCRYPTION AND DECRYPTION VALUES OF RSA AND ITS VARIATIONS

RSA type	**Average Encryption Time (ms)**	**Average Decryption Time (ms)**
Standard RSA (2-primes)	11.1	23.0
Multiprime (3-primes)	18.1	77.8
Multiprime (4-primes)	18.9	163.7
Multiprime (5-primes)	24.7	312.9

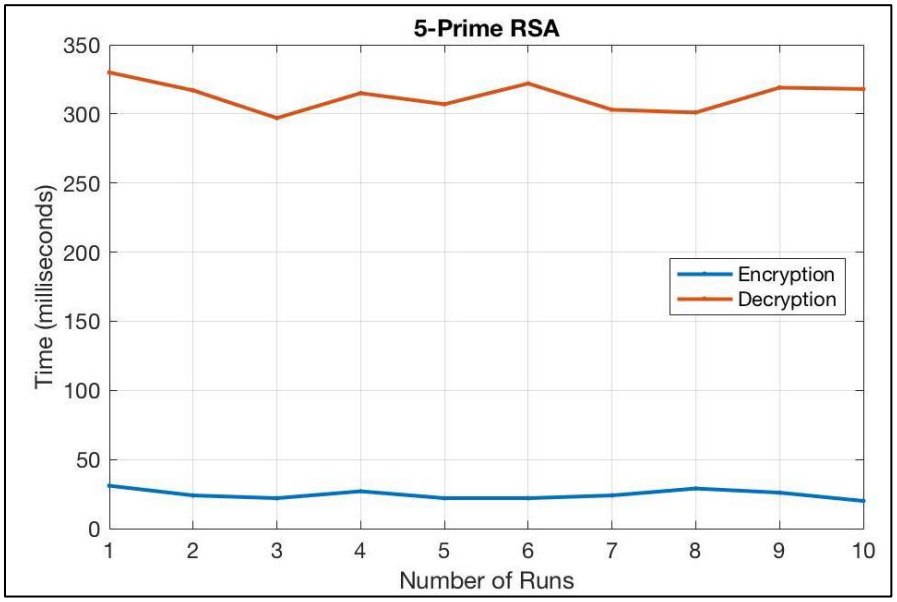

Figure 6: Encryption and decryption times for 5-prime RSA.

Doi : https://doi.org/10.54216/FPA.010105

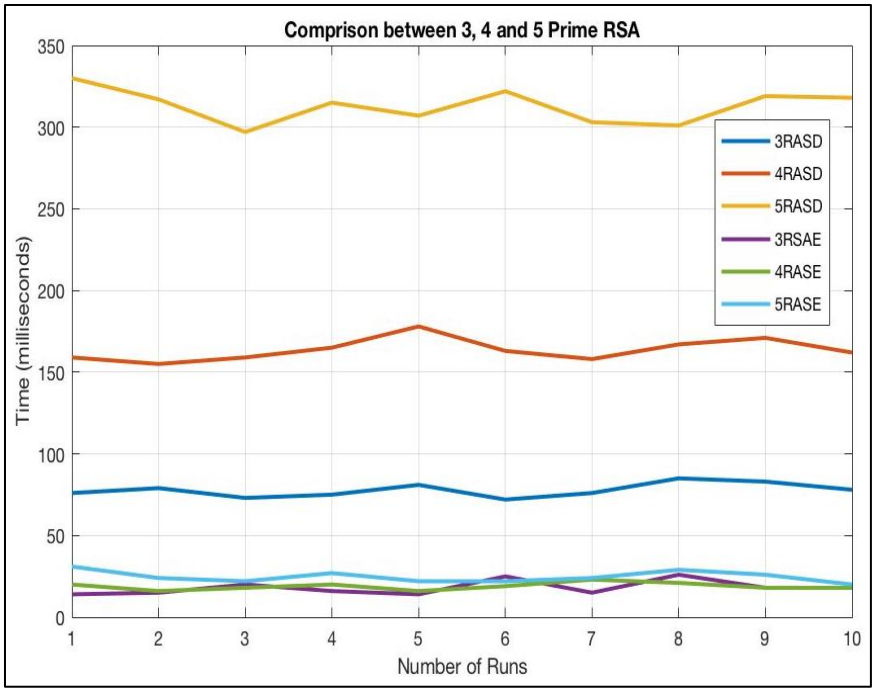

Figure 7: Comparison of Encryption and decryption times of 3, 4, and 5 prime RSA

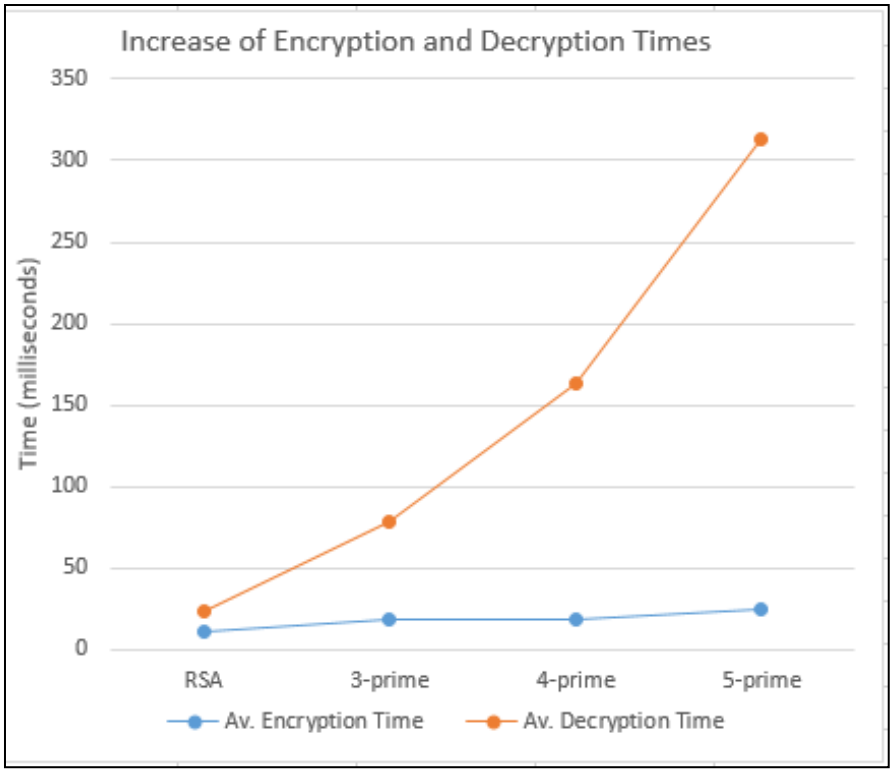

Figure 8: The plot of Avg. Encryption and Decryption time of the various variants of RSA.

The rate of increase of the average encryption and decryption times with respect to the increase in the number of primes used can be illustrated in Fig. 8.

Doi : https://doi.org/10.54216/FPA.010105

It is clear from the above graphs that when the number of primes is increased to calculate the value of N, the decryption time also increases, with little increase in encryption time. This shows that ciphertext is stronger and takes longer to decrypt. It is the advantage of using multiple prime numbers.

On the downside, using more than 3 or 4 prime numbers is not feasible because, firstly, the encryption time also increases as the number of primes increases. It can be observed from Table 1 that the average encryption time of 3 and 4 prime is almost the same, whereas the average encryption time of 5-prime considerably increases, which is not desirable. Secondly, when more than 2 prime numbers are used, the private key can be broken faster using the Chinese remainder theorem (CRT) [13][10]. Also, the prime factors of N should not fall under the elliptic curve method, and since 256 bits are considered to be within the boundary of the elliptic curve method, more than 3 primes should not be used for 1024 bit modulus [11].

"In RSA-CRT, employing the Chinese Remainder Theorem during decryption is a common practice. It results in decryption much faster than modular exponentiation. RSA-CRT differs from the standard RSA in key generation and decryption. The value of d, the secret exponent, cannot be made short as soon as d < N0.292, the RSA system can be totally broken. Keeping this in mind, we make use of the following scheme" [3].

5. Conclusion and Future Scope

RSA is a widely used public-key encryption algorithm. The positive side is the increase in decryption time on an increase in the number of primes. Although 3-prime and 4-prime are still feasible, increasing the primes further leads to increased encryption time, which is not desirable. This can be seen in the drastic increase of decryption time in 5-prime RSA. Further, with more number of primes, RSA-CRT can be useful in breaking the private key faster. Despite this drawback, multi-prime RSA is secure and practical compared to standard RSA. It doesn't allow the data to be decrypted without any effort when the user does not have access to the authentic private key, which is the main crux of encryption algorithms.

References

[1] Milanov, E. (2009). The RSA algorithm. RSA Laboratories, 1-11.

[2] Bahig, H. M., Bhery, A., & Nassr, D. I. (2012, October). Cryptanalysis of multi-prime RSA with small prime difference. In International Conference on Information and Communications Security (pp. 33-44). Springer, Berlin, Heidelberg.

[3] Zheng, M., & Hu, H. (2015). A New Factoring Attack on Multi-Prime RSA with Small Prime Difference. IACR Cryptology ePrint Archive, 2015, 1137

[4] Damgård, I., Mikkelsen, G. L., & Skeltved, T. (2014, December). On the security of distributed multiprime RSA. In International Conference on Information Security and Cryptology (pp. 18-33). Springer, Cham.

[5] Srivenkatesh, M., & Vanitha, K. Implementing Multiprime RSA Algorithm to Enhance the Data Security in Federated Cloud Computing. International Journal of Advanced Research in Computer and Communication EngineeringVol, 4.

[6] Collins, T., Hopkins, D., Langford, S., & Sabin, M. (1998). U.S. Patent No. 5,848,159. Washington, DC: U.S. Patent and Trademark Office.

[7] Takagi, T. (1998, August). Fast RSA-type cryptosystem modulo p k q. In Annual International Cryptology Conference (pp. 318-326). Springer, Berlin, Heidelberg.

[8] Takagi, T., & Naito, S. (2002). U.S. Patent No. 6,396,926. Washington, DC: U.S. Patent and Trademark Office.

[9] Prakash, G. L., Prateek, M., & Singh, I. (2014, July). Data encryption and decryption algorithms using key rotations for data security in cloud system. In 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014) (pp. 624-629). IEEE.

Doi : https://doi.org/10.54216/FPA.010105

[10] Hinek, M. J., Low, M. K., & Teske, E. (2002, August). On some attacks on multi-prime RSA. In International Workshop on Selected Areas in Cryptography (pp. 385-404). Springer, Berlin, Heidelberg.
[11] Xia, Z. Z. Z. On the Variants and Speed Methods of RSA.
[12] Goshwe, N. Y. (2013). Data encryption and decryption using RSA algorithm in a network environment. International Journal of Computer Science and Network Security (IJCSNS), 13(7), 9.
[13] Shinde, G. N., & Fadewar, H. S. (2008, April). Faster RSA algorithm for decryption using Chinese remainder theorem. In International Conference on Computational and Experimental Engineering and Sciences (ICCES) (Vol. 5, No. 4, pp. 255-262).
[14] Boneh, D., & Shacham, H. (2002). Fast variants of RSA. CryptoBytes, 5(1), 1-9.

Doi : https://doi.org/10.54216/FPA.010105