International Journal of Scientific Research in Computer Science and Engineering
Vol.8, Issue.5, pp.16-19, October (2020)

Research Paper

E-ISSN: 2320-7639

Efficient Implementation of Multi-prime RSA using Montgomery Multiplication

Mohammad Esmaeildoust[1*], Vahid Zarei[2], Amer Kaabi[3]

[1,2,3]Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Khorramshahr, Iran

[*]*Corresponding Author: m_doust@kmsu.ac.ir*

Available online at: www.isroset.org

Received: 03/Sept/2020, Accepted: 13/Oct/2020, Online: 31/Oct/2020

Abstract— RSA cryptography is one of the most common algorithm, which exclusively employed in cryptography, digital signature and security systems. By increasing the use of this algorithm, many works are reported to improve the speed of the operation and security levels. Multi-prime RSA is one these improvements over RSA which divides operations over multi prime numbers instead of two in original RSA. In this paper, in order to achieve higher performance, encryption and decryption process of Multi-prime RSA is implemented by using Montgomery multiplication. The implementation results show the noticeable improvement in the speed of the Multi-prime RSA.

Keywords— Cryptography, public key, RSA, Montgomery multiplication, Multi-prime RSA

I. INTRODUCTION

The concept of public key cryptography first was proposed by Diffie and Hellman [1]. There are many type of public key cryptography algorithms such as RSA and ECC [1,2,3]. In order to improve the security and also high computation costs, many works are done by researcher to improve the efficiency of these algorithms [4,5]. RSA cryptography algorithm first was proposed by Rivest, Shamir and Adleman [6]. RSA is widely used in public key cryptography algorithms (PKC) and its applications [7,8,9]. Example of these application is pretty good privacy (PGP) created by Phil Zimmerman which uses RSA for its key transportation [10,11]. Another usage and common operation over internet is RSA signature verification and many protocols such as SSH, OpenPGP, S/MIME and SSL/TLS stands on it [12]. With growth of computation capability, the needs for larger key sizes for RSA is needed in order to provide security. For instance, Google has upgraded the length of all its SSL certificates RSA encryption keys from 1024 to 2048 bits in 2013 [8,9]. Simple process of encryption and decryption, attract a lot of attention and is widely used by applications for secure data transmission [21,22,23].

This paper is organized as follows: section 2 present the related work, in section 3 present methodology. In section 4, the Multi-prime RSA is implemented by using Montgomery multiplication algorithm and the implementation results and comparison will be discussed. Finally, section 5 concluded the paper.

II. RELATED WORK

Security of RSA stand on difficulty of factorization of product of two large prime numbers. The RSA algorithm uses modular multiplication and exponentiation [11,12] and essentially is a slow algorithm and encryption and decryption process in RSA are done with high delay. In order to improve the speed of this algorithm, many works are reported such as Bath RSA [13], Multi-prime [14-15], Multi-power [16], R-prime [17]. Comparison and security analysis of these algorithms are included in [18]. In the work reported in [10], Dual RSA method to reduce memory consumption in the standard RSA cryptosystem has been proposed. Dual RSA generates the same public and private exponents for two distinct RSA instances. In this scheme, the plaintext is divided into two parts and is encrypt in two distinct parts with two separate moduli. Multi-prime RSA is one of these improvements which divide operation on multi prime numbers instead of two in RSA. Using smaller prime numbers results in improvement in speed of the decryption process. Although improvement in Multi-prime, performing operation modulus prime numbers has a lot of cost. Montgomery multiplication is proposed in [19] to performs multiplication in modulus prime number without division. By Employing Montgomery algorithm in the implementation of Multi-prime RSA, faster implementation by eliminating division will be achieved. On the other hands adopting Montgomery algorithm to Multi-prime RSA leads to benefit from both Montgomery method and operation on small moduli in Multi-prime RSA. In this paper, Montgomery multiplication are employed in the implementation of Multi-prime RSA in order to improve the speed of the encryption and decryption process.

III. METHODOLOGY

In this section RSA cryptosystem [6] and its variants will be discussed. Table 1 shows the RSA key generation.

Table 1: key generation in RSA cryptosystem

1. Generate two large primes p and q and compute N=p.q. *2. consider e such that gcd(e, φ(N)) = 1, where φ(N) = (p − 1)(q − 1).* *3. Compute d such that d = e −1 mod φ(N).* *Public Key = (e, N)*

In order to encrypt the plaintext M and considering (*e*, *N*) as public key, ciphertext *C* is calculated as

$$C = M^e \bmod N \quad (1)$$

In order to decrypt the message from ciphertext, considering (*d*, *N*) as private key, we have

$$M = C^d \bmod N \quad (2)$$

A. Multi-Prime RSA

Multi-prime RSA [14-15] consists of *k* prime numbers ($N=p_1.p_2....p_k$) instead of two in RSA which employs two prime number *p* and *q*. Table 2 shows the key generation process of the Multi-prime algorithm.

Table 2: key generation in Multi-prime RSA cryptosystem

1. consider k distinct primes p1, ..., pk and $N = \prod_{i=1}^{k} p_i$. *2 - Compute e and d such that d = e −1 mod φ(N), where gcd (e, φ(N)) = 1 and* $\varphi(N) = \prod_{i=1}^{k}(p_i - 1)$. *3 – For* $1 \leq i \leq k$*, compute di = d mod (pi − 1).* *Public key = (N, e)* *Private key = (N, d1, d2, ...dk)*

Table 3 shows the encryption and decryption process of Multi-prime RSA.

Table 3: Encryption and decryption in Multi-prime RSA cryptosystem

Encryption: Same as RSA cryptosystem $C = M^e \bmod N$ *Decryption:* *Calculate* $Mi = C^{di} \bmod p_i$ *for each i* *apply the Chinese reminder Theorem (CRT) to the Mi 's to get M = Cd mod N*

From table 3 it can be seen that in the decryption process modular exponentiation is reduced by operating on smaller prime numbers. In order to calculate plaintext *M* from M_i's in table 3, CRT can be used as follows:

$$M = \left|\sum_{i=1}^{k} |M_i N_i|_{P_i} Z_i\right|_N \quad (3)$$

Where $N = P_1 \times P_2 \times ... \times P_k$, $Z_i = {}^{N}/_{P_i}$ and $N_i = |Z_i^{-1}|_{P_i}$ is the multiplicative inverse of Z_i modulus p_i [20].

B. Montgomery Multiplication

Montgomery multiplication is a method for computing fast modular multiplication [19]. Montgomery multiplication performs $x \times y$ mod *N* for positive integers *x*, *y* and *N*. It reduces execution time when there are a large number of multiplications to be done. The difficulty in computing $x \times y$ mod *N* is in the reductions modulus *N*, which are, essentially division operations. Divisions are costly in execution time. If one ignores the modulus operation to the end, then the products will grow to very large numbers. This results in multiplications with large delay and also slow down the final modulus operation. Table 4 shows the details of the Montgomery multiplication algorithm.

Table 4. Montgomery Multiplication algorithm

Montgomery Multiplication *Input: two positive integer x, y* *Outpout: :U = xy mod N* *Calculate x′ = xR-1 mod N and y′ = yR-1 mod N* *Calculate T = x′×y′* *m = T×N′ mod R where (RR-1-NN′=1)* *U = (T+m×N)/R* *If U ≥ N then U – N else U*

IV. RESULTS AND DISCUSSION

In this section, Multi-prime RSA detailed in section 2 is implemented using Montgomery multiplication. As discussed in section 2, encryption process is same as standard RSA and for plaintext M, $C = M^e$ mod *N* must be calculated. In order to speed up the encryption process, Montgomery multiplication is employed and for different key sizes has been implemented on computer with Core i5 CPU and 4 GBRAM using Matlab software. The results are shown in table 5.

Table 5. Encryption delay for different key sizes

Key Size	1024	2048	4096
Encryption	0.486 ms	0.918 ms	1.807 ms

It can be concluded from table 5 that using Montgomery multiplication in the implementation of Multi-prime RSA algorithm, noticeable improvement in the speed has been achieved.

As discussed in section 2, decryption process divided into *k* prime numbers. For realize decryption each part is

implemented in parallel using Montgomery multiplication. The results are included in table 6. Because of using Multi prime number instead of two, operation of decryption process are done by using smaller moduli. Therefore operation of decryption employs both advantages of using Montgomery reduction as well as operation over smaller moduli and the faster implementation compared to encryption process is achieved.

Table 6. Multi-prime decryption time

Key size	1024	2048	4096
Decryption	0.381 ms	0.848 ms	1.672 ms

In order to make comparison, the original RSA algorithm and also Multi-prime RSA has been implemented using Matlab software. Comparisons are included in table 7. For ease of comparison, the results are included in figure 1. It can be seen that by using Montgomery multiplication, noticeable improvement in speed of encryption has been achieved.

Table 7. Delay of Encryption comparison

Key size	1024	2048	4096
Mutli-Prime Using Montgomery Multiplication	0.381 ms	0.848 ms	1.672 ms
Multi-prime	0.486 ms	0.918 ms	1.807 ms

The comparison of Multi-prime algorithm implemented with Montgomery multiplication with original RSA and Multi-prime are included in table 8.

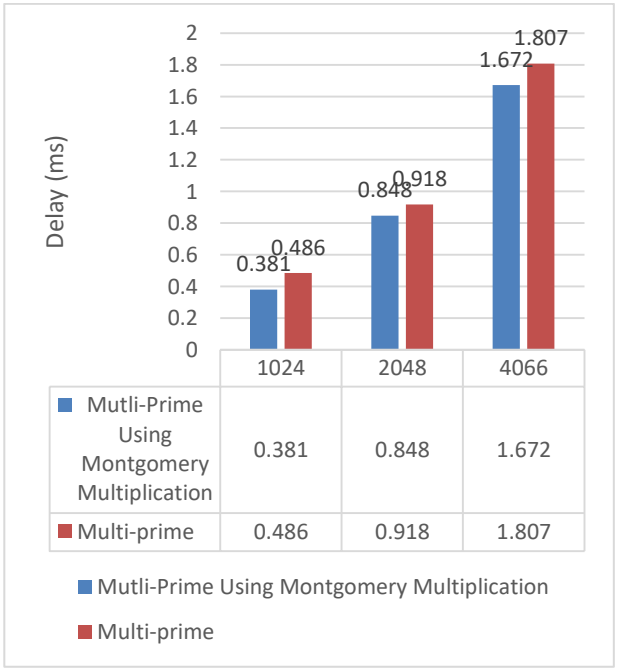

Figure 1. Encryption delay comparison

Table 8. decryption time comparison

Key size / ***Delay (ms)***	1024	2048	4096
Mutli-Prime Using Montgomery Multiplication	0.273	0.635	1.382
Multi-prime	0.335	0.687	1.487
RSA	0.643	0.895	1.768

Figure 2 shows the comparison delay of the decryption time. Overall delay of encryption and decryption time are included in figure 3. It can be seen that noticeable improvement has achieved in overall delay by using Montgomery multiplication in the implementation of Multi-prime RSA cryptosystem.

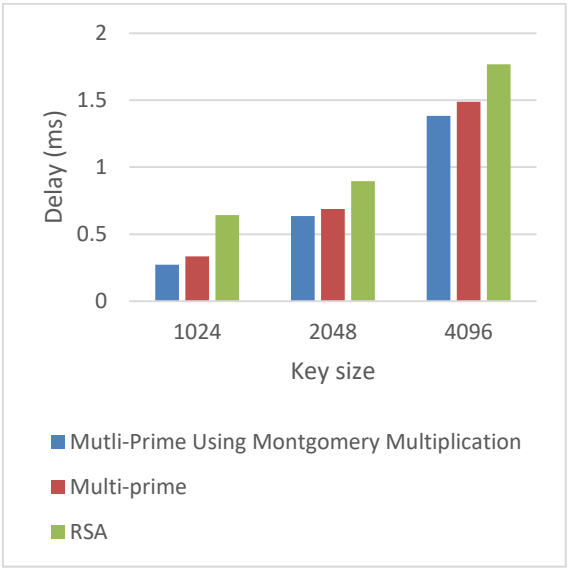

Figure 2. Comparison delay of the decryption

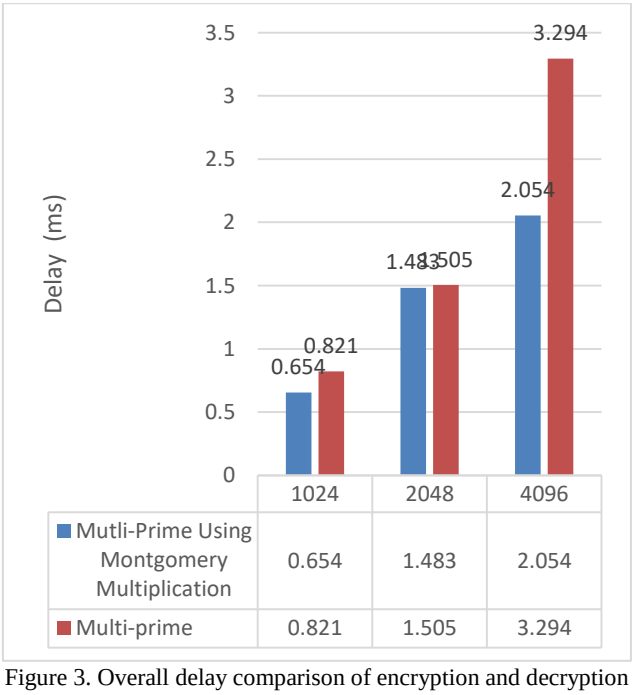

Figure 3. Overall delay comparison of encryption and decryption

V. CONCLUSION

Multi-prime RSA is one of the improvement over RSA cryptography which increased the speed of the decryption process by employing multi prime numbers instead of two in original RSA. Although the improvement has achieved, the need for more efficient implementation still exist in order to achieve higher speed in cryptosystem. In this paper, in order to achieve higher performance, encryption and decryption process of Multi-prime RSA is implemented by using Montgomery multiplication. The implementation results show the noticeable improvement in the speed of the Multi-prime RSA.

ACKNOWLEDGEMENTS

We would like to thank Khorramshahr University of Marine Science and Technology for supporting this work under research grant contract No. 139.

REFERENCES

[1] Diffie, W. and M.E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, Vol. 22: pp. 644-654, 1976.
[2] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comp.*, vol. 48, no. 177, pp. 203-209, 1987.
[3] V. S. Miller, "Use of elliptic curves in cryptography," *Proc. Adv. Cryptology LNCS*, pp. 47-426, 1986.
[4] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, K. Navi, Efficient RNS implementation of elliptic curve point multiplication over GF(p). *IEEE Trans. VLSI Syst*. Vol. 21, pp. 1545–1549, 2013.
[5] C. J. McIvor, M. McLoone, J. V. McCanny, " Hardware elliptic curve cryptographic processor over \${rm GF}_{rm (P)}\$ *", IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 53, no. 9, pp. 1946-1957, 2006.
[6] R.Rivest, A.shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, Vol.21, No.2, pp.120-126, 1978.
[7] N. Priya and M. Kannan, "Comparative Study of RSA and Probabilistic Encryption," *International Journal Of Engineering And Computer Science*, vol. 6, no. 1, pp. 19867 - 19871, January 2017.
[8] H. B. Pethe and S. R. Pande, "Comparative Study and Analysis of Cryptographic Algorithms," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 5, no. 1, pp. 48-56, 1 January 2017.
[9] T.Collins, D.Hopkins, S.Langford, and M.Sabin, "public key cryptographic Apparatus and method ",*US Patent #5*, 848,159, jan.1997.
[10] H.M Sun, M.E Wu, W.CTing and M.J Hinek, "Dual RSA and its security analysis", *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2922-2933, 2007.
[11] G. R. Blakey, "A Computer Algorithm for Calculating the Product AB Modulo M," *IEEE Transaction on Computers*, vol. 32, no. 5, pp. 497-500, 1983.
[12] L. Harn, "Public-Key Cryptosystem Design Based on Factoring and Discrete Logarithms," *IEE Proceedings: Computers and Digital Techniques*, vol. 144, no. 3, pp. 193-195, 1994.
[13] A. Fait, "Batch RSA," *Advance in Cryptology CRYPTO '89*, vol. 435, pp. 175-185, 1989.
[14] D. Boneh and H. Shacham, "Fast Variants of RSA," CryptoBytes, vol. 5, no. 1, pp. 1-10, 2002.
[15] T. Collins, D. Hopkins, S. Langford and M. Sabin, "Public Key Cryptographic Apparatus and Method". US Patent #5848, 1997.
[16] T. Takagi, "Fast RSA-type Cryptosystem Modulo pkq," Advances in Cryptology - CRYPTO '98, vol. 1462, pp. 318-326, 1998.
[17] C. A. M. Paixon, "An efficient variant of the RSA cryptosystem," Cryptology ePrint Archive, 2002.
[18] D. Garg and S. Verma, "Improvement over Public Key Cryptographic Algorithm," in IEEE International Advance Computing Conference, Patiala, 2009.
[19] P.L. Montgomery, "Modular Multiplication without Trial Division," *Math. Computation*, vol. 44, no. 170, pp. 519-521, 1985.
[20] K. Navi, A. S. Molahosseini and M. Esmaeildoust, "How to Teach Residue Number System to Computer Scientists and Engineers," in *IEEE Transactions on Education*, vol. 54, no. 1, pp. 156-163, 2011.
[21] Rupal Yadav, Kaptan Singh, Amit saxena, "*Hybrid Cryptographic Solution to Overcome Drawbacks of RSA in Cloud Environment*", International Journal of Computer Sciences and Engineering, Vol.8, Issue.8, pp.56-60, 2020.
[22] V. Kapoor, "Data Encryption and Decryption Using Modified RSA Cryptography Based on Multiple Public Keys and 'n'prime Number," *International Journal of Scientific Research in Network Security and Communication*, Vol.1, Issue.2, pp.35-38, 2013.
[23] Harsh Sahay, "*Modified RSA Cryptosystem with Data Hiding Technique in the Terms of DNA Sequences*", International Journal of Computer Sciences and Engineering, Vol.7, Issue.9, pp.91-94, 2019.

Authors Profile

Mohammad Esmaeildoust received the M.Sc. and the Ph.D. degree in computer architecture from Shahid Beheshti University, Tehran, Iran, in 2008 and 2012, respectively. He is currently an Assistant Profesor with the Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Khorramshahr, Iran. His Research interests include the network security, cryptography and computer arithmetic.

Vahid Zarei received the M.Sc degree in Electronic engineering from Shahrood University of Tecnology, Shahrood, Iran, in 2012. He is currently a Lecturer with the Faculty of University of Marine Science and Technology , Khorramshahr, Iran. His Research interests include the signal and image processing, pattern recognition and cryptography.

Amer Kaabi received the Ph.D. degree in Applied Mathematics from Ferdowsi University of Mashhad, Mashhad, Iran, in 2006. He is currently an Associate Professor with the Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Khorramshahr, Iran. His research interests include the numerical analysis, operational research, numerical algebra and computer arithmetic.

