A Survey on Montgomery Modular Multiplication

Greshma Eldhose[1] Betcy K Joy[2]
[1]PG Scholar [2]Assistant Professor
[1,2]Department of Electronics and Telecommunication Engineering
[1,2]CKC, Kerala, India

Abstract— This paper present a survey on Montgomery modular multiplication algorithm used in public key cryptography [1] system. It is a very important algorithm in which the efficiency of cryptosystem depends on the speed of modular multiplication. This survey provides the comparison between different modifications done in Montgomery modular multiplication.

Key words: Public Key Cryptosystem, Montgomery Modular Multiplication

I. INTRODUCTION

In 1976, diffie and hellman introduced public key cryptography [1] to eliminate the need for a secure channel to exchange important information. Public key cryptosystem are importance as they provide confidentiality authentication and data integrity. Hence the public key cryptosystem has to do the complex operations, for example modular exponentiation [2] [1] etc. Modular exponentiation can be accomplished by repeating modular multiplication.

It is crucial to optimize the modular multiplication to improve the performance of the overall public key cryptosystem .one of the method to optimize the modular multiplication is the Montgomery modular multiplication[2] [4]. In this, the need of division operation is eliminated at the cost of additional multiplication. Since the cost of multiplication is much lower than the division operation.

II. MONTGOMERY MODULAR MULTIPLICATION

Modular multiplication of two integers can be done by performing,

$$Z = A * B * \bmod N$$

Here A and B and N are k bit number and the value of N should be greater than A and B. In Montgomery modular multiplication this can be computed by ,

$$Z' = \text{MONT}(A, B, N) = A*B*R^{-1} \bmod N$$

In which the value of R^{-1} can be computed by,

$$R * R^{-1} \bmod N = 1$$

Where $R = 2^k$, k is the word size of modulus N. MONT represent the Montgomery product.

The important factor is the value of A and B should be converted to Montgomery domain. The conversion can be done by,

$$A' = A*R \bmod N$$
$$B' = B*R \bmod N$$

Here the B' and A' are the Montgomery form of A and B. Also Z can be converted back by,

$$Z = Z'*1*R^{-1} \bmod N$$

III. RELATED WORKS

Several modifications are done in the Montgomery modular multiplication to get more efficient, low cost and high performance design. The modification done in different papers are discussed below. First we go through the need of Montgomery modular multiplication in the cryptography system.

Encryption/decryption [1] is one of the important step in the cryptography system. This process required modular exponentiation algorithm. The modular exponentiation is done by the repeated modular multiplication. Hence the performance of the cryptosystem is depends on the through put rate of modular multiplication and the number of required modular multiplication. Montgomery modular multiplication is one of the algorithm which can carry out fast modular multiplication. It eliminate the need for division operation by using additional multiplication operation. The cost of multiplication operation is much lower than the division operation. It is addition multiplication and shift operation to perform the modular multiplication.

R.L Rivest , Ashamir , L Adleman ,[1] they proposed a method for implementation a public key cryptosystem. In this, it provide a secure communication without the use of key exchange. Here they break the long message into a series of blocks and represent each block as such an integer. It is used to convert message into numeric form. Then the encryption and decryption process take place. It required modular multiplication. The challenging issue in the Montgomery modular multiplication is the time consuming carry propagation for very large modulus. Several approaches are proposed to avoid the time consuming carry propagation.

To solve the carry propagation again several modifications are done. One of the method is to use the carry save adder. In this the carry is saved without propagating it to the next bit. S R Kuang, J P Wage, K C Chang, H W Hsu [6] proposed energy efficient high through put Montgomery modular multiplication for RSA cryptosystem. In this each input as well as the intermediate result are also stored in the carry save format in expense of multiple cycle overhead and the number of operands for the carry save adder is increased.

Again efficient carry save adder architecture for Montgomery modular multiplication is proposed by Y Y Zhang, Z Li, L Yang and S W Zhang [4]. In this each input and output operands are represented in binary. The intermediate result in carry save format to avoid carry propagation. But the format conversion from carry save format [4] [5] of the final modular product into its binary representation is needed at the end of each Montgomery modular multiplication. This conversion can be accomplished by extra clock carry propagation adder. Hence it required extra clock cycle for the format conversion and it also increases the hardware complexity.

To enhance the efficiency of Montgomery modular multiplication, combination of carry save adder with other technique such as high radix [8] is used. The high radix means the number of multiplier bits are inspected per clock cycle. It enhances the efficiency of carry save adder in the expense of

critical path delay and increased number of clock cycle for performing high radix multiplication.

IV. CONCLUSION

In this paper, we have given an introduction of Montgomery modular multiplication which is used in the cryptography system. And also compared different modification done in Montgomery modular multiplication. From the above discussion it is clear that, we can further improve the performance of the Montgomery modular multiplication.

REFERENCES

[1] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.
[2] P. L. Montgomery, "Modular multiplication without trial division," *Math. Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.
[3] H. Zhengbing, R. M. Al Shboul, and V. P. Shirochin, "An efficient architecture of 1024-bits cryptoprocessor for RSA cryptosystem based on modified Montgomery's algorithm," in *Proc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst.*, Sep. 2007, pp. 643–646.
[4] Y.-Y. Zhang, Z. Li, L. Yang, and S.-W. Zhang, "An efficient CSA architecture for Montgomery modular multiplication," *Microprocessors Microsyst.*, vol. 31, no. 7, pp. 456–459, Nov. 2007.
[5] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," *IEE Proc.- Comput. Digit. Techn.*, vol. 151, no. 6, pp. 402–408, Nov. 2004.
[6] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput Montgomery modular multipliers for RSA cryptosystems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 11, pp. 1999–2009, Nov. 2013.
[7] P. Amberg, N. Pinckney, and D. M. Harris, "Parallel high-radix Montgomery multipliers," in *Proc. 42nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2008, pp. 772–776.
[8] G. Sassaw, C. J. Jimenez, and M. Valencia, "High radix implementation of Montgomery multipliers with CSA," in *Proc. Int. Conf. Microelectron.*, Dec. 2010, pp. 315–318.
[9] F. Gang, "Design of modular multiplier based on improved Montgomery algorithm and systolic array," in *Proc. 1st Int. Multi-Symp. Comput. Comput. Sci.*, vol. 2. Jun. 2006, pp. 356–359.
[10] Y. S. Kim, W. S. Kang, and J. R. Choi, "Asynchronous implementation of 1024-bit modular processor for RSA cryptosystem," in *Proc. 2nd IEEE Asia-Pacific Conf. ASIC*, Aug. 2000, pp. 187–190.

