International Journal of Electrical Engineering.
ISSN 0974-2158 Volume 4, Number 3 (2011), pp. 353-361

http://www.irphouse.com

File Management Architecture for Montgomery Algorithm in Elliptic Curve

M. Prabu[1] and R. Shanmugalakshmi[2]

[1]Research Scholar, Anna University Coimbatore, Tamil Nadu, India
E-mail: prabu_pdas@yahoo.co.in
[2]Assistant Professor/CSE, Government College of Technology,
Tamil Nadu, India
E-mail: shanmuga_lakshmi@yahoo.co.in

Abstract

In this article, we briefly discussed about different algorithms on Elliptic Curve Cryptography, which is based on Montgomery operation such as register file management allocation. A new technique of register file allocation system is developed and compared with the existing system. Furthermore, a new approach is introduced to allocate the register with least level of steps. This level reduction is used to increase the performance of the algorithm as well as to improve the execution level. These least utilization of registers also provides the same level of output. This paper presents an improved File Register Management for Montgomery Algorithm. The first important contribution lies in the reduction of steps to increase the performance. The second contribution is to increase the level of File Register Management that will help to reallocate the registers. As a result this architecture level achieves a great performance to register allocation and their feat reflected in the File Management Architecture.

Keywords: Montgomery Algorithm, Elliptic Curve Cryptography, File Register Management, File Allocation.

Introduction

In this article, a reconfigurable cryptographic implementation register file has been

efficiently compared with two algorithms namely Montgomery and Modified Montgomery (MM) for elliptic curve operations, such as multiplication, addition and doubling. Elliptic curve is one of the strongest public-key cryptosystem, which is generally used for authentication protocols. The performance of such cryptosystems is primarily determined by the implementation efficiency of the modular multiplication and exponentiation. For performance as well as for physical security reasons, it is often advantageous to recognize them by using hardware. Hardware implementations of Elliptic curve cryptosystems are widely studied as in [2]–[4]. Most proposed implementations of Elliptic curve systems are based on Montgomery modular algorithm. There have been various proposals for systolic array architectures for modular multiplication [2]–[4] and an enhancement during execution time was also found when implementing dedicated units instead of configurations in ALU.

Organization of this paper

The rest of this paper is structured as follows. In Section 2, we recall the basic of the Elliptic curve cryptosystems and Montgomery Algorithm. In Section 3, the model approach is presented to performance of operation. In Section 4, we propose a modify Montgomery algorithm, and then we discuss its security and implementation properties. In Section 5, we turn our attention on power analysis problem challenges. In Section 6, we make concluding remarks.

Background

Montgomery algorithm developed a very efficient technique to compute in the group associated to an elliptic curve over a non-binary finite field G(Fq), in which prime field is F(p).The Montgomery algorithm proves useful for point compression in ECC. More precisely, instead of sending a point as part of some cryptographic protocol. There are variety of ways to perform the Elliptic Curve based Montgomery operations. Our aim, in this article is to give a crystal clear view of its operations and functions. The operations may be classified as Montgomery addition, Montgomery multiplication and Montgomery doubling. A number of Montgomery multiplier, have been suggested [12, 13, 14] and their hardware architectures are designed to deal with a specific maximum number of registers. However, when the computer processing power increases, it provides the same effective security level. Then the cryptosystem will verify the level of register management to be equalized.

Diagram of operations

In this article, we present a single, modified version of file management algorithm, which is suitable for both hardware and software implementations.

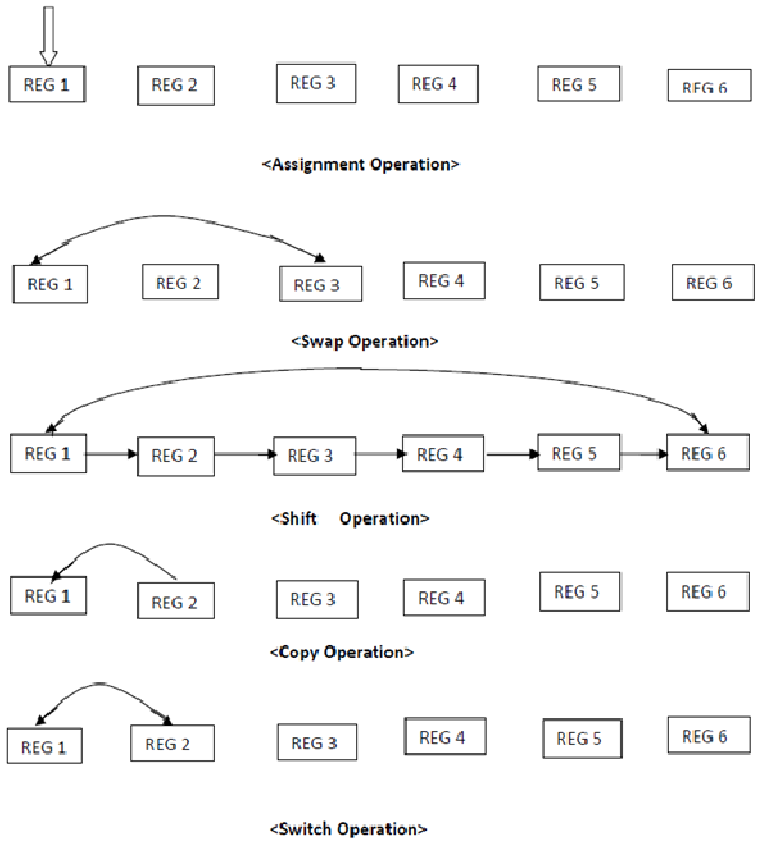

Figure 1: Performance of operations.

Side Channel Attacks

In recent years, cryptosystems have come under harass from various forms of side channel attacks. Kocher et al[4].discovered that cryptosystem implementation leak information which can help an attacker to access secret data. One such technique for retrieving secret information is SPA. SPA involves monitoring the power consumption of cryptographic algorithms in single execution.

When a cryptographic algorithm is actually implemented, its weakness can be induced with some unanticipated ways. The attackers can track these weaknesses to circumvent the security of the underlying algorithms. These kind of attacks are referred and reviewed by Kelsey[9], as so called side channel attacks.

- Active attack
- Passive attack

How to analyze the power consumption

Normally, the applications are viewed as tamper- resistant hardware. All side channel attacks power analysis, exploits the power consumption of a cryptographic system and

it can be carried out easily. These attacks are very effective in breaking the cryptographic algorithms. In the literature review, many designs consider the face of the ECC over prime fields [14].While the power consumption will be higher, the calculation time and the total concern consumption per ECC operations will be lower. Each and every instruction has a different power consumption value therefore it is possible to return the sequence of instruction during the algorithm execution. It is used to describe the leakage of intended information from a supported device, when implementing the cryptographic algorithm.

Nature of Montgomery Algorithm

ECC calculations in prime fields GF(p) are based on Galois Field. It consists of addition, multiplication and doubling, these operations must be preformed modulo a prime number. Montgomery multiplication is a common operation in many public key cryptographic algorithms including elliptic curve cryptosystems GF(p).Lopez-dahab's Montgomery scalar multiplication algorithm, which uses a projective co-ordinates system in which projective system [1] is used.

Montgomery Adding algorithm
Algorithm 1.

Step1. T2 ← X1 + X2

Step 2. T2 ←$(T2)^2$

Step 3. T1 X1 × X2

Step 4. X1← x

Step 5. X1 ← T2 × X1

Step 6. X1 ← X1 + T1

Design for the Montgomery Algorithm

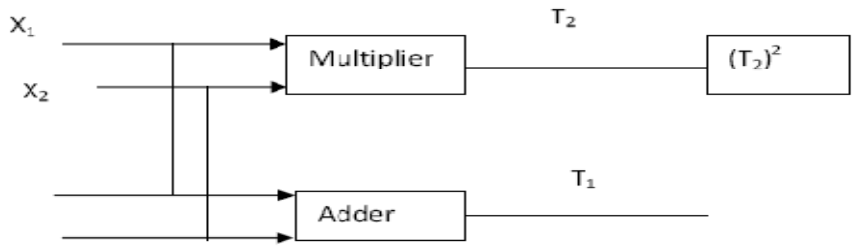

Figure 2: Initial level of Design.

The following two step can be explained through the hardware design. The inputs

are X1, X2 for the both multiplier and adder. After the Multiplier the output (T2) can be extracted by the Square concept.

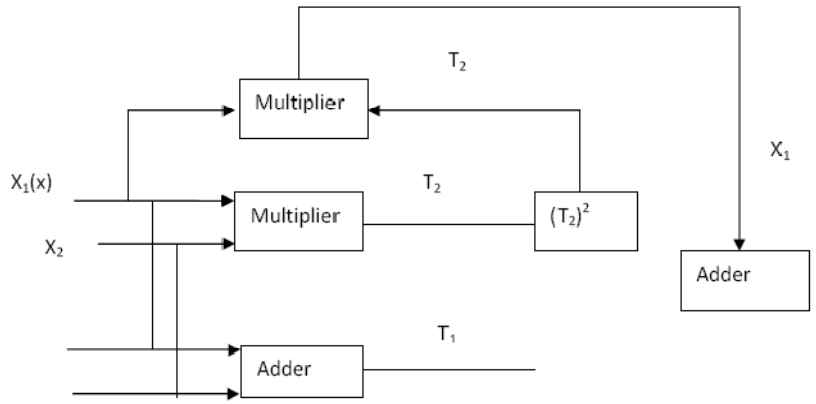

Figure 3: Second Level of Design.

After completing the level 1 design, the second level extracted by the level 1 design's output that is $(T_2)^2$ .After getting the result from the level one, it interacted to the input again X1(x), which is given by the user . After that process that the resulting value will be stored in X1.

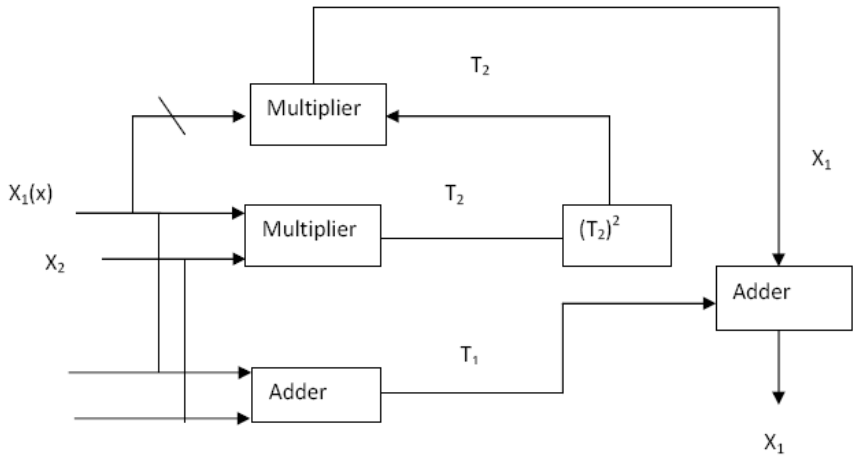

Figure 4: Final Designing Section for Montgomery Algorithm.

The final level. The latest X1 value is interacted with the adder with T1.

A Register File Management
Circular Shift Register File Architecture

Table 1: Register File Management for Adding Algorithm.

Steps	Field	Operation	Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg6
1	Initial		X_1	X_2	Z			
2	Shift			X_1	X_2	Z		
3	Copy		X_1	X_1	X_2	Z		
4	Add	$T_2 \leftarrow x_1+x_2$	T_2	X_1	X_2	Z		
5	Shift			T_2	X_1	X_2	Z	
6	Copy		T_2	T_2	X_1	X_2	Z	
7	Shift			T_2	T_2	X_1	X_2	Z
8	Multiply	$T_2 \leftarrow T_2 2$	T_2			X_1	X_2	Z
9	Shift*4			X_1	X_2	Z	T_2	
10	Multiply	$T_1 \leftarrow X_1 x X_2$	T1		X_2	Z	T_2	
11	Switch			T_1	X_2	Z	T_2	
12	Load x	$X_1 \leftarrow x$	X_1	T_1	X_2	Z	T_2	
1	Shift* 2		T_2		X_1	T_1	X_2	Z
14	Switch			T_2	X_1	T_1	X_2	Z
15	Multiply	$X_1 \leftarrow T_2 x X_1$	X_1	T_2		T_1	X_2	Z
16	Switch		T_2	X_1		T_1	X2	Z
17	Shift*5		X_1		T_1	X_2	Z	T_2
18	Add	$X_1 \leftarrow X_1+T_1$	X_1		T_1	X_2	Z	T_2

Our proposed design of Montgomery Addition algorithm's implementation table shows six level register used for making two operations. This 6 level register concept is already done in[4]. But in our approach same level of registers are used in calculation part and there is a reduction in implementation steps shown in Fig

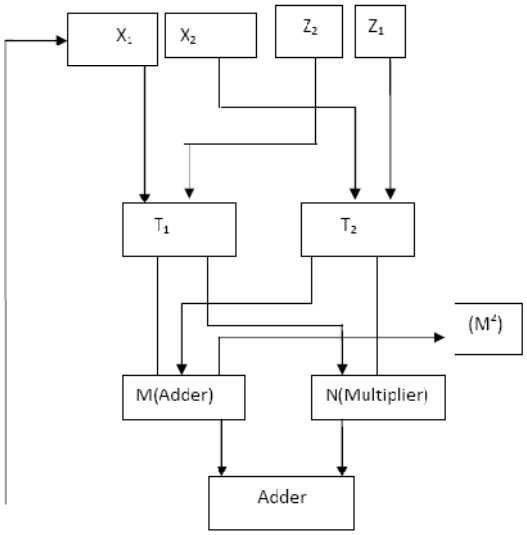

Figure 5: The Modified Design for Montgomery Algorithm.

Overall System Architecture

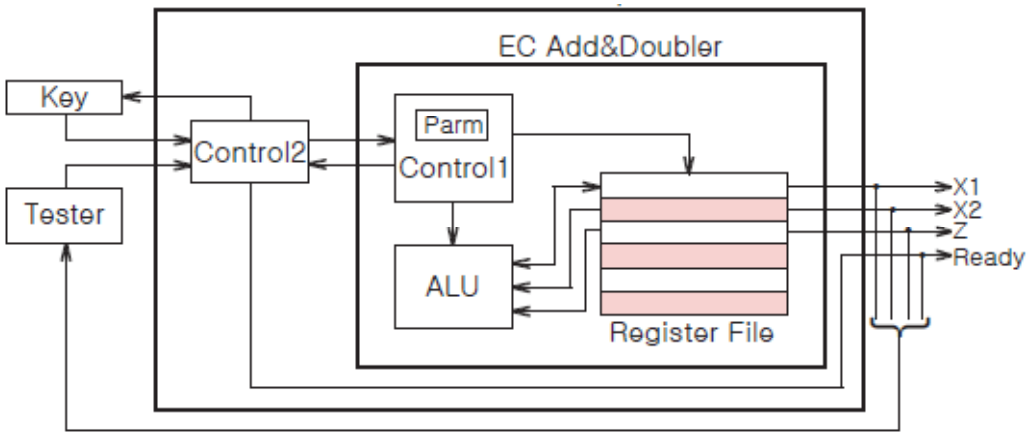

Figure 6: Overall System Architecture [1].

In this design, all the operations are performed by using the numbers in the Montgomery domain. Montgomery multiplication, which is faster that n, the ordinary modular multiplication and leads to much smaller hardware utilization. The design is based on the affine representation of the curve points. This simplifies the design and reduces the necessary storage elements to the minimal of two register for a curve point and leads to lower power consumption.

Allocation and Implementation of Register File Management for Montgomery Algorithm

Modified algorithm

Algorithm 2

Initially declare $T_1 \leftarrow X_1.Z_2, T_2 \leftarrow Z_1.X_2$

Registers X1,Z2,X2,Z1,T1,T2,M,N

Step 1: $M \leftarrow T_1+T_2$

Step 2: $Z_1 \leftarrow M^2$

Step 3: $N \leftarrow T_1.T_2$

Step 4: $M \leftarrow x.Z_2$

Step 5: $X_1 \leftarrow M+N$

Table 2: Register File Management for Modified Adding Algorithm.

Steps	Field	Operation	Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg6
1	Initial		X_1	X_2	Z			
2	Shift			X_1	X_2	Z		
3	copy		X_1	X_1	X_2	Z		
4	Add	$T_2 \leftarrow X_1 + X_2$	T_2	X_1	X_2	Z		
5	shift			T_2	X_1	X_2	Z	
6	copy		T_2	T_2	X_1	X_2	Z	
7	Shift			T_2	T_2	X_1	X_2	Z
8		$T_2 \leftarrow T_2 2$	T_2			X_1	X_2	Z
9	Shift*4			X_1	X_2	Z	T_2	
10		$T_1 \leftarrow X_1 x\ X_2$	T_1		X_2	Z	T_2	
11	Shift		T_2		T_1		X_2	Z
12	swap			T_2	T_1		X_2	Z
13		$X_1 \leftarrow x.\ T_2$	X_2	T_2	T_1		X_2	Z
14	Shift			X_1	T_2	T_1	X_2	Z
15	swap		X_1		T_2	T_1	X_2	z
16		$X_1 \leftarrow X + T_2$	X_1		T_1	T_2	X_2	z

Discussion &Conclusion

The comparison of important properties with the referenced Register file management on Montgomery algorithm is summarized in Table 1. And the proposed Register file management on modified Montgomery algorithm is explained. Compared with the known output, the proposed algorithm was more suitable for the real applications. Optimistically, our comparative study results will encourage the further researchers on this worth full topic.

We have proposed a new Montgomery register file management and a step by step execution of the file management that implements the Montgomery cryptographic algorithms in a efficient manner.It has been shown that the register file is used to complete the encryption process through assigning spaces. The new proposed file register management is faster and level of register is also decreased than the previous Montgomery file management architecture.

References

[1] A. Cilardo, A. Mazzeo and N. Mazzocca, “Representation of elements in F2m enabling unified field arithmetic for elliptic curve cryptography”, ELECTRONICS LETTERS 7th July 2005 Vol. 41 No. 14

[2] P. Montgomery: Speeding the Pollard and elliptic curve methods of factorization.Mathematics of Computation, Vol. 48. (1987) 243–264

[3] J. L´opez and R. Dahab: Fast multiplication on elliptic curves over GF(2m) without precomputation. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Lecture Notes in Computer Science, Vol. 1717. Springer- Verlag (1999) 316–327

[4] N. Meloni: Fast and Secure elliptic Curve Scalar Multiplication Over Prime Fields Using Special Addition Chains. Cryptology ePrint Archive: listing for 2006 (2006/216) (2006)

[5] C. Paar: Light-Weight Cryptography for Ubiquitous Computing. Invited talk at the University of California, Los Angeles (UCLA). Institute for Pure and Applied Mathematics (December 4, 2006)

[6] K. Sakiyama, L. Batina, N. Mentens, B. Preneel and I. Verbauwhede: Smallfootprint ALU for public-key processors for pervasive security. Workshop on RFID Security (2006) 12 pages

[7] L. Batina, N. Mentens, K. Sakiyama, B. Preneel and I. Verbauwhede: Low-costElliptic Curve Cryptography for wireless sensor networks. Third European Workshop on Security and Pri-vacy in Ad hoc and Sensor Networks. Lecture Notes in Computer Science, Vol. 4357. Springer-Verlag (2006) 6–17

[8] E. ¨A Ozt¨Aurk, Berk Sunar and Erkay Savas: Low-power elliptic curve cryptography using scaled modular arithmetic. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Lecture Notes in Computer Science, Vol. 3156. Springer-Verlag (2004) 92–106

[9] A. Satoh and K. Takano: A Scalable Dual-Field Elliptic Curve Cryptographic Processor. IEEE Transactions on Computers, Vol. 52. No. 4 (2003) 449–460

[10] J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997

[11] C. Walter, "Systolic Modular Multiplication," IEEE Trans. Computers, vol. 42, no. 3, pp. 376-378, Mar. 1993

[12] P. Kornerup, "A Systolic, Linear-Array Multiplier for a Class of Right-Shift Algorithms," IEEE Trans. Computers,vol. 43, no. 8, pp. 892-898, Aug. 1994

[13] W.C. Tsai, C.B. Shung, and S.J. Wang, "Two Systolic Architecture for Modular Multiplication,", IEEE Trans. VLSI, vol. 8, no. 1, pp. 103-107, Feb. 2000

[14] RECently, A.K. Tecca, and C.K.Koc published several papers[9][9] about scabal able Montgomery multiplier.1