International Journal of Computational Intelligence and Informatics, Vol. 1 : No. 3, October - December 2011

Area Compactness Architecture for Elliptic Curve Cryptography

M. Janagan
Arunai College of Engineering,
Tiruvannamalai
janaganraja89@gmail.com

M. Devanathan
Arunai College of Engineering,
Tiruvannamalai
devanathanvlsi@gmail.com

Abstract

Elliptic curve cryptography (ECC) is an alternative to traditional public key cryptographic systems. Even though, RSA (Rivest-Shamir-Adleman) was the most prominent cryptographic scheme, it is being replaced by ECC in many systems. This is due to the fact that ECC gives higher security with shorter bit length than RSA. In Elliptic curve based algorithms elliptic curve point multiplication is the most computationally intensive operation. Therefore implementing point multiplication using hardware makes ECC more attractive for high performance servers and small devices. This paper gives the scope of Montgomery ladder computationally. Montgomery ladder algorithm is effective in computation of Elliptic Curve Point Multiplication (ECPM) when compared to Elliptic Curve Digital Signature Algorithm (ECDSA). Compactness is achieved by reducing data paths by using multipliers and carry-chain logic. Multiplier performs effectively in terms of area/time if the word size of multiplier is large. A solution for Simple Power Analysis (SPA) attack is also provided. In Montgomery modular inversion 33% of saving in Montgomery multiplication is achieved and a saving of 50% on the number of gates required in implementation can be achieved.

Keywords: Elliptic Curve Cryptography, Elliptic curve point multiplication, Montgomery modular inversion, Signal Power Analysis, Digital signature.

1. Introduction

Recent advances in telecommunications and computer networking have brought us into the era of electronic mail. The widespread implementation and use of such systems will require secure means for validating and authenticating the electronic messages they exchange. Validation and authentication refer to the methods of certifying the contents of a message and its originator, respectively. Attacks are based on the information when faults are induced in cryptographic devices. Hence these risks can be mitigated by employing strong cryptography to ensure authentication, authorization, data confidentiality, and data integrity. Symmetric cryptography, which is computationally inexpensive, can be used to achieve some of these goals. However, it is inflexible with respect to key management as it requires pre-distribution of keys. On the other hand, public key cryptography allows for flexible key management, but requires a significant amount of computation. However, Elliptic Curve Cryptosystems [2], [9] are becoming an alternative to the traditional RSA systems [8], as this system offers similar security but for a smaller size. For instance, a 256-bit ECC key size provides the same level of security as an equivalent 3072-bit RSA key [10]. The various primitives of Elliptic Curve Cryptography are signature generation, signature verification, EC point multiplication, modular inversion, modular addition, modular subtraction and these primitives are evaluated using level of security, functionality, methods of operation, performance, and ease of implementation.

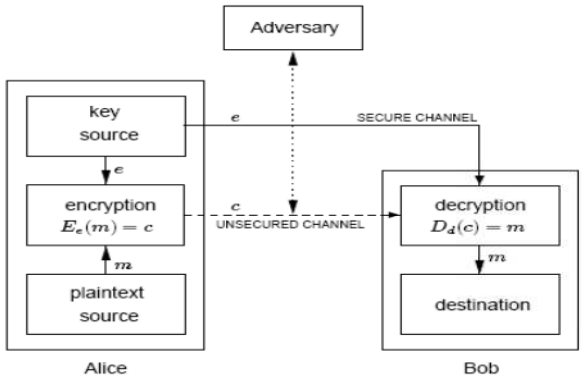

Fig.1 Encryption using public-key technique.

2. Objective

The main objective of using elliptic curve cryptography is to overcome the critical computations of modular multiplication, modular addition, modular subtraction, which reduces the number of operations required and which in turn reduces the total number of gates and data paths required. If the data paths and gates required is reduced which in turn reduces the area of architecture. Montgomery algorithm is used for performing modular operations such as modular multiplication, addition, subtraction and modular inversion. Montgomery ladder algorithm uses constant power consumption in performing

modular operations, which is capable of reducing Simple power analysis attack. Using common data variables reduces the data-paths and power consumptions approximately to 33% [7]. Cryptographic strength is as follows

Table 1
Equivalent Cryptographic Strength.

Symmetric	56	80	112	128	192	256
RSA n	512	1024	2048	3072	7680	15360
ECC p	112	161	224	256	384	512
Key size ratio	5:1	6:1	9:1	12:1	20:1	30:1

3. Montgomery Modular Multiplication

3.1 Modular Multiplication

The public key cryptosystem's performance is primarily determined by the efficiency of implementation of modular exponentiation which is performed by modular multiplication. Modular multiplication is performed using two methods namely

1. Multiply-then-reduce
2. Interleaved multiplication and reduction.

3.1.1 Multiply-then-reduce

Method uses add-and-shift algorithm, booth's algorithm, Barrett's algorithm and so on. In the add-and-shift algorithm the reduction step is implemented with successive subtractions until the result is less than the modulus. In booth's algorithm the size of number of partial products generated exceeds the size of the multiplier operand. So we move to interleaved multiplication and reduction method.

3.1.2 Interleaved Multiplication and Reduction method

In this method Montgomery's algorithm simplest method for calculations the modular multiplication is used. Montgomery algorithm replaces the division by adding a shift and modulate, which make the computer faster. Montgomery algorithm is well suited for hardware implementation such as PPGA or ASIC.

Booth's method is slower in response because of its less GPGA usage, but it has no variations in response time, hence Montgomery multiplication is used due to its variations on response time.

3.2 Montgomery Multiplication

The Montgomery multiplication (MM) of two integers X and Y with a parameter of n-bits precision produces an output variable $Z=MM(X, Y) = XYr^{-1} \mod M$. Where $r=2^n$ and M is an integer given as $2^{n-1} < M < 2^n$ such that the gcd(r, m) = 1. M should be a prime number of a product of prime number hence the condition gcd (r, M)= 1 condition is satisfied. Complexity of the Montgomery multiplication (MM) is much lower when compared to the regular modular multiplication. Montgomery multiplication is done by a series of point addition and point doubling, the simulation graph of point addition and point doubling is as shown in the graph taken from reference [3].

4. Algorithm

ECDSA is a standardized algorithm for generating and verifying digital signatures [4]. Figure 2 shows the operation hierarchy for ECDSA signature operations. The critical computations in terms of area and speed in an elliptic curve digital signature are an Elliptic Curve Point Multiplication (ECPM). For ease computation of Elliptic Curve Point Multiplication Montgomery ladder algorithm is used, another benefit of this algorithm is the fact that the branches for key-bit 0 and 1 are balanced.

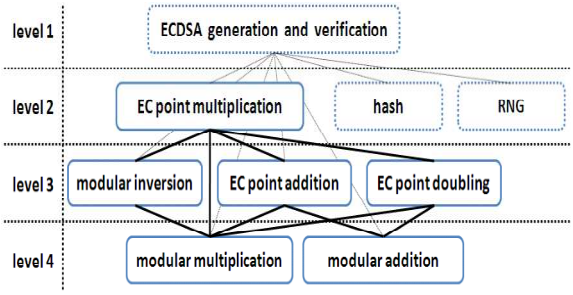

Fig. 2. Operation hierarchy of ECDSA signature generation and verification.

Algorithm.1 shows that an ECPM can be performed using two lower-level point operations, namely Elliptic Curve Point Addition (ECPA) and Elliptic Curve Point Doubling (ECPD). These two operations are implemented as reported in [1]. An ECPA and an ECPD can be formulated in one set of formulas (ECPAPD).

Require: point P, non-negative integer $k = (1k_{l-2}...k_1k_0)_2$
Ensure: $Q = x(kP)$
  $Q \leftarrow P$, $S \leftarrow 2P$
  for $i = l - 2$ downto 0 **do**
    if $k_i = 1$ **then**
      $x(Q) \leftarrow x(Q + S)$, $x(S) \leftarrow x(2S)$
    else
      $x(S) \leftarrow x(Q + S)$, $x(Q) \leftarrow x(2Q)$
    end if
  end for
  Return $x(Q)$

Algorithm 1: Montgomery Ladder Algorithm

4.1 Backgrounds in ECC

An elliptic curve E over Zp is defined by equation $y^2=x^3+ab+b$, where p is prime greater than 3, and a, b ∈

Z_p and $4a^3+2+b^3$ not equal to 0 (mod p).

1. P+o = o+p = p for all p ϵ(Z_p).
2. Let P=(x_1, y_1), Q=(x_2, y_2) then x_3 and y_3 are obtained from the equation

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda (x_1 - x_3) - y_1$$

$$\lambda = \{ \frac{y_2 - y_1}{x_2 - x_2} \text{ if P not equal to Q ,}$$

$$\frac{3x^2 + a}{2y_1} \text{ if Q equal to Q}\}$$

4.2 Hash function

Cryptographic hash functions play a fundamental role in modern cryptography. Hash functions take a message as input and produce an output referred to as a *hash code*, *hash-result*, *hash-value*, or simply *hash*. More precisely, a hash function h maps bit strings of arbitrary finite length to strings of fixed length, say n bits. For a domain D and range R with h: D$\rightarrow$R and mod D > mod R, the function is many-to-one. The basic idea of cryptographic hash functions is that a hash-value serves as a compact representative image (sometimes called an *imprint*, *digital fingerprint*, or *message digest*) of an input string, and can be used as if it were uniquely identifiable with that string.

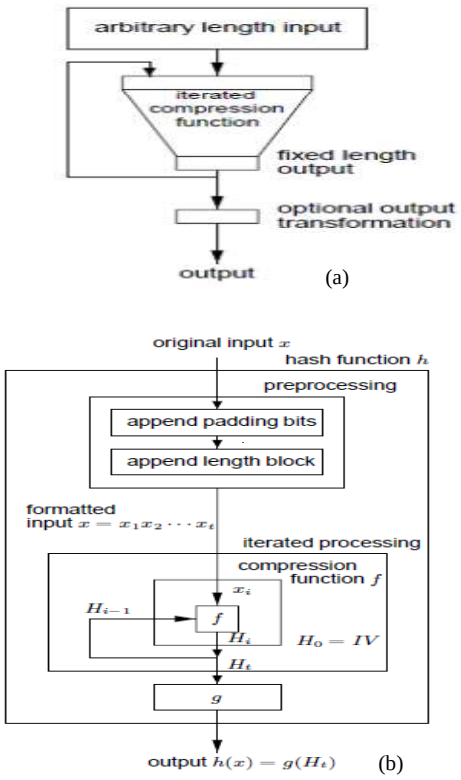

Fig 3. Schematic of iterated hash function; (a) High-level view; (b) Detailed view

5. Processor Architecture

Elliptic Curve Digital Signature Algorithm is based on the results of Modular Multiplication (MM), and Modular Addition/Subtraction (MAS). Modular Multiplication is effectively implemented using Montgomery Ladder Algorithm and Modular Addition Subtraction is calculated using consecutive addition and/or subtraction. The proposed processor architecture is designed in such a way that it can handle MMs and MASs, it is as shown in the Figure 4.

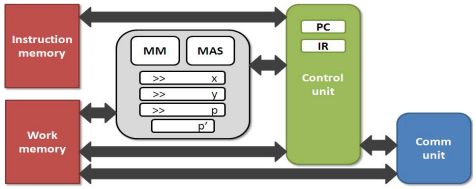

Fig. 4 General Processor Architecture

In the proposed processor, the usual sequence for instruction handling is followed. To provide the user with a set of programs rather than a single program, the instruction memory is divided into multiple parts. With an offset, given through the communication unit, the user can select which program shall be executed. The processor is designed to aid secure communication. The architecture of Montgomery modular multiplication is as shown in the figure 5. The multiplier architecture operation is mainly based on the input given to the control unit. If the input of control unit is 00, 01, 10, 11 multiplier operation is modular multiplication, modular inversion, modular addition and modular subtraction respectively [13].

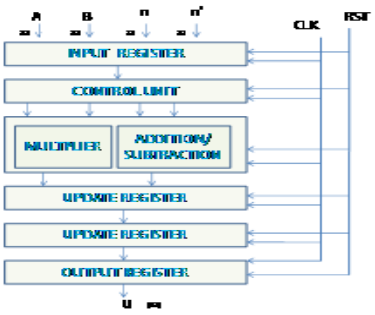

Fig 5. Montgomery modular multiplier

In the proposed processor, the usual sequence for instruction handling is followed. To provide the user with a set of programs rather than a single program, the instruction memory is divided into multiple parts. With an offset, given through the communication unit, the user can select which program shall be executed. The processor is designed to aid secure communication.

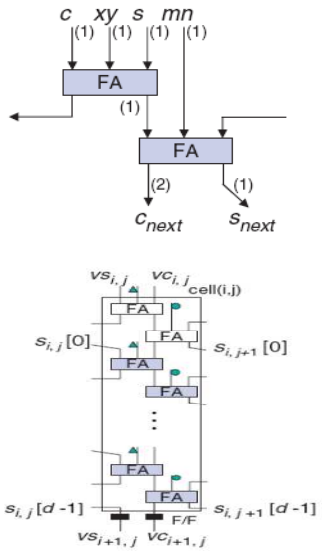

Fig 6. Configuration of Modular Addition with 4-2 Carry save adder

A modular multiplication algorithm invented by Montgomery, computes the answer to a completely different sum to the one we asked for. The good thing about this technique is that the decisions about what multiple of the modulus to add depend only on the last digit of the result, which is not affected by any carries. So Ripple carry adder is replaced by carry-save architecture which works very well indeed here.

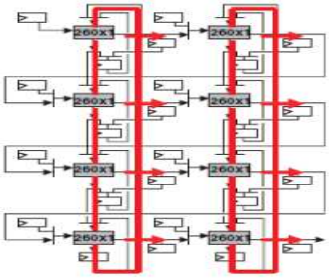

Fig.7: Reconfigurable Data path

The architecture of Montgomery modular multiplication is as shown in the figure 7 . The multiplier architecture operation is mainly based on the input given to the control unit. If the input of control unit is 00, 01, 10, 11 multplier operation is modular multiplication, modular inversion, modular addition and modular subtraction respectively [12]. The operation of modular inversion is as shown in the flowchart 1. There are two types of modular inversion they are classical modular inversion and Montgomery modular inversion, since Montgomery modular inversion is more effective operation it is used in the proposed architecture.

6. Memory Organization

The data's are received word serially hence the memory must be scanned for operand's word, so RAM modules are used to stored data's. RAMs are loaded from the host system with the operand values. In order to make the system flexible enough for several values of precision the memory elements are designed as a queue. The maximum length of the queue (Qmax) depends on the maximum number of words that may be stored in the memory (emax). Since the size of memory components is the major limiting factor for operand precision, the memory space should be allocated previously with a provision of maximum number of words to be used in system. The memory spaced required by this approach is not more than the conventional Montgomery modular (MM) algorithm need. Timing interface between nodes are theoretically not needed but practically it is needed and it is as shown in the table 2, which is referred from [10].

7. Signal Power Analysis

For analyzing SPA [7] fault analysis technique is used where the attacker checks the threshold condition to induce a fault by changing the fault intensity. For example, we consider how the output signal is determined at the logic gates of AND, OR and XOR. In Fig. 8, to describe this we suppose a case where input is determined first, and is determined later. For the AND and OR gates, the output value could be determined regardless of the value of b. For instance, if a *is* 0, the output of the AND gate becomes 0. If a is 1, the output of the OR gate is determined to be 1. In contrast, the output value of the XOR gate is determined only when both a and b are determined. In order to avoid the trace of power consumed by the processor AND and OR gates in the architecture is replaced by XOR gates.

9. Experimental Result

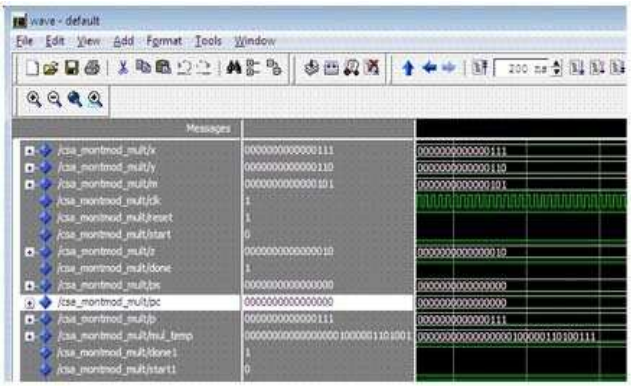

Fig 8. Simulation Result of Montgomery Multiplier

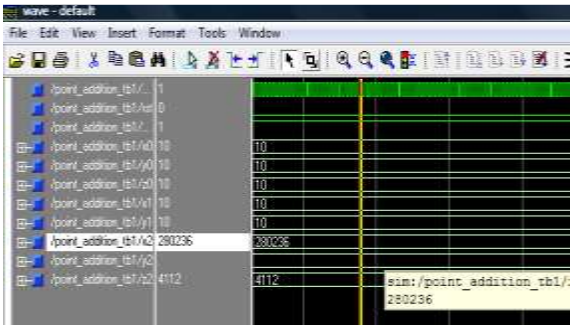

Fig 9. Simulation - Point addition m = 163

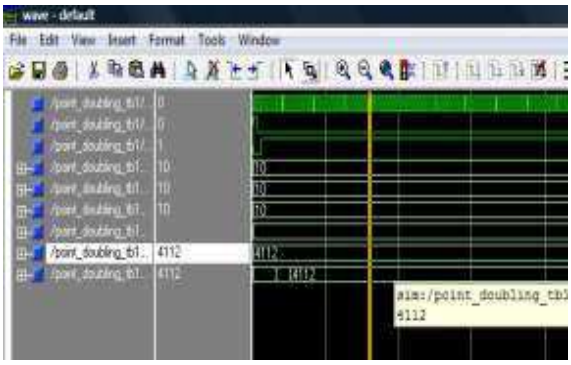

Fig 10. Point doubling for m =163

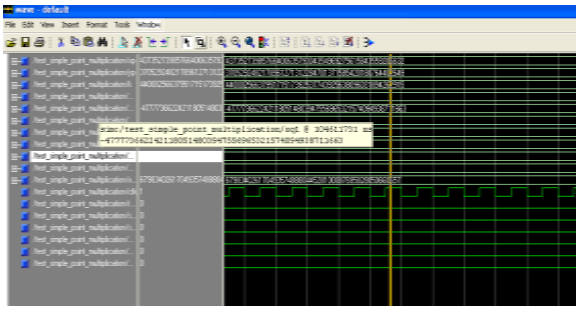

Fig 11. Scalar multiplication m = 163

Table 2
Timing between nodes of communication

Operation	Theoretical	Measured
Key pair generation		
End-to-end	n/a	1143.32
Computation	0.28	**0.60**
ECC only	0.28	0.50
Signature generation		
End-to-end	n/a	2073.84
Computation	0.35	**0.94**
ECC only	0.28	0.54
Signature verification		
End-to-end	n/a	2296.73
Computation	0.67	**1.61**
ECC only	0.55	1.00

10. Conclusion

In this paper, we proposed a compact architecture for Elliptic Curve Cryptography in which Montgomery ladder algorithm is used for Elliptic curve point Multiplication, Addition, and Subtraction. Use of Montgomery algorithm made constant power consumption in the modular operations and hence message transferred using method is not duplicated by any other person. The use of Carry save adder reduced number of required gates to about 50% and 30% of architecture size is reduced.

References

[1] Jo Vliegen, NeleMentens, Jan Genoe, An Breaken, Serge Kubera, AbdellahTouhafi and Ingrid Verbauwhede, (IEEE 2010), “A Compact FPGA-based architecture for elliptic compact cryptography over prime fields”, pp.313-316

[2] Sahbuddin Abdul Kadir, ArifSasongko, Muhammad Zulkifli, (IEEE 2010) , “Simple Power Analysis Attack Against Elliptic Curve Cryptography Processor on PFGA Implementation”, International conference on electrical Engineering and Informatics

[3] D. Giry. (2009) “Cryptographic key length recommendation”.

[4] National Institute of Standards and Technology(2009).FIPS 186-3: “Digital Signature Standard”.

[5] T. G¨uneysu and C. Paar (2008),“ Ultra high performance ECC over NIST primes on commercial FPGAs”, In E. Oswald and P. Rohatgi, editors, Proceedings of the 10th international workshop on Cryptographic Hardware and Embedded Systems (CHES), number 5154 in Lecture Notes in Computer Science, pages 62–78. Springer-Verlag.

[6] P. Marietti, M. Oliveri, G. Scotti, and A. Trifiletti, (IEEE 2008), “A New Dynamic Differential Logic Style as a Countermeasure to Power Analysis Attacks”, pp.364-367.

[7] Hamid Reza Ahmadi and Ali Afzali-Kusha, (IEEE 2007), “Low-Power Flexible GF (p) Elliptic-Curve Cryptography Processor”.

[8] K. Sakiyama, N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede(2007), “Reconfigurable modular arithmetic logic unit supporting high-performance RSA and ECC over GF(p)”, International Journal of Electronics, 94(5):501–514.

[9] Ciaran J. Mcivor, MaireMcloone (2006), “ Hardware Elliptic Curve Cryptographic Processor Over GF(p)”, IEEE Transactions on circuits and systems.

[10] Ciaran J. McIvor, MaireMcLoone, Member, IEEE and John V. McCanny, Fellow, (IEEE 2006), “Hardware Elliptic Curve Cryptographic Processor over GF (p)”, IEEE Transactions on circuits and systems

[11] Ciaran McIvor, MaireMcLoone, John V McCanny, (IEEE 2004), “FPGA Montgomery Modular Multiplication Architectures Suitable For ECCs over GF (p)”, pp.509-512

[12] C. McIvor, M. McLoone, and J.V. McCanny (2004), “An FPGA Elliptic Curve Cryptographic accelerator over GF(p)”, IEE Conference Publications, 2004(CP506):589–594.

[13] D. Hankerson, A. Menezes, and S. Vanstone (2003) , " Guide to Elliptic Curve Cryptography", Springer-Verlag.

[14] S.B.O rs, L. Batina, B. Preneel, and J. Vandewalle(2003), "Hardware implementation of an Elliptic Curve Processor over GF(p)", In In 14th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), pages 433–433. IEEE Computer Society.

[15] T. Izu, B. M¨oller, and T. Takagi(2002), " Improved Elliptic Curve multiplication methods resistant against Side Channel Attacks", In Progress in Cryptology (IndoCrypt), number 2551 in Lecture Notes in Computer Science, pages 296–313. Springer- Verlag.

[16] M. Brown, D. Hankerson, J. Lopez, A. Menezes(2001), "Software implementation of the Elliptic Curves Over Prime Fields".

[17] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes(2000), "Software Implementation of Elliptic Curve Cryptography over Binary Fields".

[18] I. Semaev, "Evaluation of discrete logarithms on some elliptic curves", to appear in Mathematics of Computation.