Ultra High-Performance ASIC Implementation of SM2 with Power-Analysis Resistance

Dan Zhang
Department of Microelectronics
Tsinghua University
Beijing, China
Email: zhangdan10560363@163.com

Guoqiang Bai
Department of Microelectronics
Tsinghua University
Beijing, China
Email: baigq@tsinghua.edu.cn

***Abstract*—In this paper, we propose a high-performance implementation of elliptic curve cryptography over SCA-256 prime field by introducing an all-new isochronous architecture, which can also resist power-analysis attack. By modifying Montgomery ladder-based scalar multiplication, point addition (PA) and point double (PD) can operate synchronously, resisting simple power analysis (SPA) and double attack with minimum time-cost. Then PA and PD are designed to be strictly isochronous units by matching our configurable modular multiplication unit of pipelined stage. Both algorithm and hardware schedule are optimized from bottom to up, random cycles are also inserted to resist differential power analysis (DPA). In the hardware evaluation using CMOS standard cell library of 0.13μm, our ECC processor achieves 211μs and 8.5μJ for one scalar multiplication with 208k gate counts. Compared to other related designs, our architecture offers not only 2$\sim$6 times better area-time product but also great power-analysis resistance.**

***Index Terms*—elliptic curve cryptography, SM2, power analysis resistance, isochronous architecture, fast reduction scheme**

I. Introduction

In order to support the growing demand of sensitive information exchange, Elliptic Curve Cryptography (ECC) has become widely accepted in practice due to its higher security compared to other public-key cryptosystems, such as RSA and Diffie-Hellman. After some other international standard organizations including ANSI, NIST and IEEE, in December 2010, Chinese State Cryptography Administration (SCA) published the national public key cryptographic algorithm based on ECC in [1], known as SM2. It is enforced to replace RSA in the commercial encryption system by the government, bringing this algorithm a spacious application foreground. SM2 is defined over a 256-bit pseudo-Mersenne prime field, called SCA-256. Combining with other recommended parameters, this special prime brings SM2 enough latent possibility of optimization in hardware design.

Since proposed, there have been a lot of researches [2]-[12] aiming at hardware implementation of ECC, ranging from algorithm improvement to circuit optimization. Among these study results, most ones have been conducted targeting versatility, focusing on the prime field or even dual-field. At the same time, various side-channel attacks are confirmed to be effective in ECC decryption, especially SPA and DPA. Performance, security and cost are indispensable but restricted mutually dimensions of ECC implementation.To the authors knowledge, there are very few researches concentrating on one specific prime field and then achieving excellent balance between comprehensive performance and overall security. But in fact, this type of design is meaningful for some recommended elliptic curves, such as SM2. To fill this gap, in this paper, we propose an all-new isochronous architecture aimed at high-performance VLSI implementation of SM2 with multiple power-analysis resistance

The rest of this paper is organized as follows. Section II presents the background information of ECC and SM2. Section III introduces our all-new isochronous architecture for ECC over SCA-256. Both bottom-up optimization and security consideration are presented in detail. Section IV gives the implementation performance as well as comparison with previous works. Finally, Section V concluded this paper .

II. Preliminaries

A. Elliptic curves over GF(p)

A non-supersingular elliptic curve over GF(p) is usually expressed as the Weierstrass equation in Eq. 1:

$$E : y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in GF(p)$, $4a^3 + 27b^2 \neq 0 \pmod p$. All the solutions $(x, y) \in GF(p) * GF(p)$ of this equation make up the curve, together with the point P_∞ at infinity. To form an abelian group, ECC arithmetic defines the unique addition operation of two points over this curve in Eq. 2. Let $P = (x_1, y_1), Q = (x_2, y_2) \in E$, then $R(x_3, y_3) = P + Q \in E$. If $P \neq Q$, we have the point addition (PA) formulas, otherwise we have point double (PD) formulas.

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = -y_1 - (x_3 - x_1)\lambda$$

where

$$\lambda = \begin{cases} \frac{3x_1^2 + a}{2y_1} & (x_1, y_1) = (x_2, y_2) \\ \frac{y_1 - y_2}{x_1 - x_2} & otherwise. \end{cases} \tag{2}$$

In practical applications, elliptic curve cryptosystems need to perform various functions, such as digital signature, public key encryption and so on. The main building block of all these functions is scalar multiplication, defined as $kP = \sum_1^k P =$

978-1-4799-8364-3/15/$31.00 ©2015 IEEE

$P + P + \cdots + P$, where P is a point on elliptic curve and k is a random integer. It is computed by a series of PA and PD, further decomposed into a certain number of finite field operations, as shown in Fig. 1.

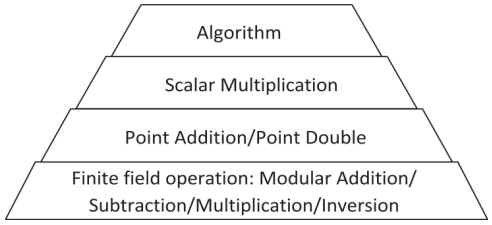

Fig. 1. Hierarchy of operations in ECC algorithm

The form of equation, points and operation formulas change with different coordinates. Since modular division is the most complex and costly finite field operations, designer usually select projective, Jacobian, mix or other coordinates to avoid it. Standard projective points (X, Y, Z) correspond to affine points (XZ^{-1}, YZ^{-1}), while Jacobian projective points (X, Y, Z) correspond to affine points (XZ^{-2}, YZ^{-3}), where $Z \neq 0$. Computation complexity is shown in TABLE I.

TABLE I. CALCULATION COST OF PA AND PD

Coordinate	Point Double	Point Addition
Affine	1I,4M	1I,3M
Projective	10M	14M
Jacobi	8M	16M
Mixed Jacobi-Affine	$\sim$	11M

[1] M for modular multiplication, I for modular inversion.

B. SM2 and SCA-256

Compared with international standard ECC algorithm, SM2 adopts the unique prime field, called SCA-256. This algorithm has also been improved in some procedures, such as the steps of encryption, which enhances its applicability and safety in commercial environment. The parameters of SM2 are clearly specified in [1], given as follows:
(1) The pseudo-Mersenne prime field:

$$p_{SCA-256} = 2^{256} - 2^{224} - 2^{96} + 2^{32} - 1$$

(2) Weierstrass equation and Base point $G = (G_x, G_y)$:

$$E : y^2 = x^3 + ax + b$$

$a = p_{SCA-256} - 3$
$G_x = 32c4ae2c\ 1f198119\ 5f990446\ 6a39c994\ 8fe30bbf\ f2660be1\ 715a4589\ 334c74c7$
$G_y = bc3736a2\ f4f6779c\ 59bdcee3\ 6b692153\ d0a9877c\ c62a4740\ 02df32e5\ 2139f0a0$

III. PROPOSED ECC PROCESSOR

In this section we'll present our high-performance ASIC implementation of SM2 in detail. The main algorithm of scalar multiplication is firstly decided based on security consideration. Then the succeeding units are achieved and optimized from the bottom up.

A. Main Algorithm Selection based on Security Consideration

Side-channel attacks such as SPA and DPA have proved to be threatening for ECC device. The resistant strategy is to dilute or even cut off the relevance of power consumption and secret key. For ECC, the LR-DAA scalar multiplication in Algorithm 1 is usually used for SPA resistance. But in step4, the constant addictive factor of base point gives opportunity for double attack. This step also must be executed after step3, resulting in lower speed and hardware efficiency.

Algorithm 1 LR-DAA Scalar Multiplication

Input: integer k and point P, m = bit length of k
1: **Initial:**$Q_1 = P, Q_0 = 0, i = m - 1$
2: **While** $i \geq 0$, **do**:
3: $Q_0 = 2Q_0$
4: $Q_1 = Q_0 + P$
5: $Q_0 = Q_{k_i}$, $i = i - 1$
6: **end While**
Output: $kP = \sum_1^k P = Q_0$

An improved algorithm of this is Montgomery ladder-based scalar multiplication (MLSM) describe in [7]. Although doubling the necessary operation amount, its updated double-and-add-always property can fundamentally resist SPA and double attack. AS shown in Algorithm 2, we modify the MLSM so that PA and PD can be computed simultaneously. Then by making them isochronous unit, the time cost is cut in half. Based on so many improvements, DPA attack may still be conducted because of the key-dependent step of point switch. Aligning all the power traces to one is the precondition of successful crack. So we insert random cycles to MLSM, blocking the alignment and then the DPA attack.

Algorithm 2 Modified Montgomery Ladder-based Scalar Multiplication

Input: integer k and point P, m = bit length of k
1: **Initial:**$Q_1 = Q_0 = 0, Q_T = P, i = 0$
2: **While** $i < m$, **do**:
3: $Q_1 = Q_0 + Q_T$, $Q_2 = 2Q_T$
4: **If**$(k_i = 1)$ $Switch(Q_0, Q_1)$
5: $Q_T = Q_2$, $i = i + 1$
6: **end While**
Output: $kP = \sum_1^k P = Q_0$

B. Modular Addition/Subtraction and Inversion

As the basic building block of ECC algorithm, optimization for finite field operations will significantly improve the global performance. Modular operations defined over prime field ask all the results to sustain in the range of [0:p-1], which brings subsequent data processing steps of adding or subtracting p to traditional results. For higher hardware efficiency, we design a combination module which can execute both modular addition and subtraction, costing only one cycle. We also adopt a radix-4 binary inversion in [9] to speed the modular inversion. This fast radix-4 unified division algorithm can scan two bits at one cycle of logical judgment, effectively reducing execution time.

C. Modular Multiplication

In this subsection, we designs a configurable modular multiplication unit with pipelined architecture, which can be flexibility invoked by upper operation. Modular multiplication is made up of regular multiplication and modular division. For Mersenne primes which can be written as the difference of several exponent powers of 2, only shift and additions are needed to achieve the modular division result, and it is faster than any other algorithm aiming at general prime. Since our SCA-256 has the similar form with Mersenne ones, we deduce the Fast Reduction Scheme for it, as shown in Algorithm 3. Let 512-bit input m be expressed on the base of 2^{32}. In SCA-256, $(2^{480}, \cdots, 2^{256})$ can be equivalently transformed into expressions of $(2^{224}, \cdots, 2^{0})$. Then 512-bit modular division is eliminated by a fixed number of shift and 256-bit addition.

Algorithm 3 Fast Reduction Scheme for SCA-256

Input: $m = m_15 * 2^{480} + m_14 * 2^{448} + \cdots\cdots + m_0$
$= (m_{15}, m_{14}, \cdots\cdots, m_0), m_i \in [0, 2^{32})$

1: define $S_0, S_1, \cdots\cdots, S_9$ $asfollows$:
$S_0 = (m_7, m_6, m_5, m_4, m_3, m_2, m_1, m_0)$
$S_1 = (m_{15}, 0, 0, 0, 0, 0, m_{15}, m_{14})$
$S_2 = (m_{14}, 0, 0, 0, 0, 0, m_{14}, m_{13})$
$S_3 = (m_{13}, 0, 0, 0, 0, 0, 0, m_{15})$
$S_4 = (m_{12}, 0, m_{15}, m_{14}, m_{13}, 0, 0, m_{15})$
$S_5 = (m_{15}, m_{15}, m_{14}, m_{13}, m_{12}, 0, m_{11}, m_{10})$
$S_6 = (m_{11}, m_{14}, m_{13}, m_{12}, m_{11}, 0, m_{10}, m_9)$
$S_7 = (m_{10}, m_{11}, m_{10}, m_9, m_8, 0, m_{13}, m_{12})$
$S_8 = (m_9, 0, 0, m_{15}, m_{14}, 0, m_9, m_8)$
$S_9 = (m_8, 0, 0, 0, m_{15}, 0, m_{12}, m_{11})$

2: $Sum = 2 * \sum_{i=1}^{4} S^i + \sum_{i=5}^{10} S^i - 2^{64} * (m_8 + m_9 + m_{13} + m_{14})$

Output: $m \mod p_{SCA-256} = Sum \mod p_{SCA-256}$

For regular multiplication, we adopt the multiplier-based architecture. If we use M's N-bit multiplier for 256-bit multiplication, then the execution cycle number will be as Eq. 3. Adding up of the partial multiplication results brings one more cycle. Execution cycles of fast reduction scheme can also be precisely controlled according to hardware resources. We can get isochronous multiplication and modular division by making a reasonable match of multipliers and adders. A two-pipelined stage is also introduced so that they can execute at the same time, greatly raising hardware utilization.

$$(\frac{256}{N})^2 * \frac{1}{M} + 1 \tag{3}$$

D. Point Addition and Point Double

To avoid time-consuming modular inversion, we adopt Jacobian coordinate in our design, and it yields the fastest point double and point addition appropriate for our modified MLSM architecture. In this coordinate, point addition formulas are defined as Eq. 4, point double as Eq. 5.

$$\begin{cases} X_3 = (Y_2 Z_1^3 - Y_1)^2 - (X_2 Z_1^2 - X_1)^2 (X_1 + X_2 Z_1^2) \\ Y_3 = (Y_2 Z_1^3 - Y_1)[X_1 (X_2 Z_1^2 - X_1)^2 - X_3)] - Y_1 (X_2 Z_1^2 - X_1)^3 \\ Z_3 = (X_2 Z_1^2 - X_1) Z_1 \end{cases} \tag{4}$$

$$\begin{cases} X_3 = [3(X_1 + Z_1^2)(X_1 - Z_1^2)]^2 - 8 X_1 Y_1^2 \\ Y_3 = 3(X_1 + Z_1^2)(X_1 - Z_1^2)[12 X_1 Y_1^2 - 9(X_1^2 - Z_1^4)^2] - 8 Y_1^4 \\ Z_3 = 2 Y_1 Z_1 \end{cases} \tag{5}$$

The former needs 16 times modular multiplication while the later needs 8 times. Since one more multiplication cycle is needed for the pipelined stage, PA needs 17 while PD needs 9. Considering about data dependency, we carefully schedule the multiplication sequence so that extra modular addition/subtraction is fully in parallel with multiplication. To achieve isochronous PA and PD unit, we use Eq. 3 to calculate the hardware need for our configurable modular multiplication unit. One multiplication can be saved by either of PA and PD, since they both need the square of Q_T's x-axis coordinate. Besides, one more cycle is added for the cost of data storage. Plugging all the numbers into Eq. 3, we get Eq. 6 and Eq. 7. The more accurate our chosen solution is, the better matching capability PA and PD get.

$$(\frac{256^2}{M_1 * N^2} + 2) * (9 - 1) = (\frac{256^2}{M_2 * N^2} + 2) * 17 \tag{6}$$

$$(\frac{256^2}{M_1 * N^2} + 2) * 9 = (\frac{256^2}{M_2 * N^2} + 2) * (17 - 1) \tag{7}$$

Based on the closest integer solution of Eq. 7, $N = 64$, $M_1 = 1$, $M_2 = 2$, we designed two multiplication modules, one with one 64-bit multiplier for PD while the other with two for PA, and the later borrows the ${X_t}^2$ form the former, as shown in Fig. 2. Speed and hardware cost of this configuration both meet our expectation. By this new architecture, PA and PD can execute simultaneously with strictly same cycle number, and we can easily insert random cycles to block the alignment of power traces for DPA resistance.

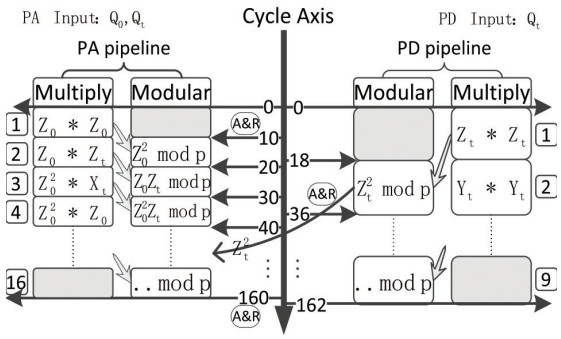

Fig. 2. Block Diagram of PA and PD's Isochronous Architecture

E. SM2 Architecture

The whole architecture is made up of three modules, as presented in Fig. 3. PA and PD are encapsulated into independent function units. But they share the common modular addition/subtraction unit. Register heap and its own store logic make up the storage module. It efficiently stores and exchanges all the data in ECC algorithm. Main Control module serves as the commander. This brain contains our implementation algorithm of SM2 and guides the other two modules to perform it. Since the arithmetic module and storage module are designed to execute some functions independently, main module can control them by instructions transmission, which greatly improves the execution efficiency and flexibility.

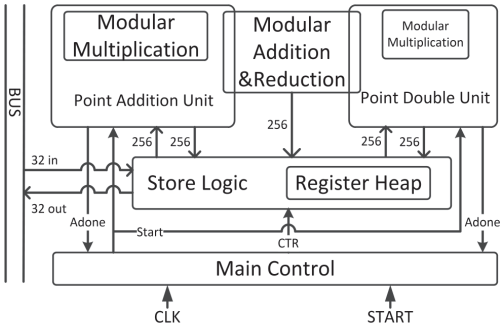

Fig. 3. Block Diagram of SM2 Control Architecture

IV. Comparison and Analysis

Our architecture has been verified in Verilog-HDL and evaluated in 0.13m CMOS standard cell library. Results show that working frequency can be set at 215MHz, with critical path delay of 4.66ns. Based on the execution cycle number of 45.5k, it takes about 211μs and 8.5μJ to complete one scalar multiplication, with the total area of 208K gates. The previously published results targeting 256-bit prime field are compared with our performance in TABLE II. Our architecture offers the best area-time product and the most comprehensive analysis resistance.

TABLE II. Performance Comparison with Previous Work

	ours	**[8]**	**[4]**	**[5]**	**[9]**
Curve	GF(p) SCA-256	GF(p) 256-bit	GF(p) 256-bit	GF(p) 256-bit	GF(p) 256-bit
Library	0.13m	90nm	90nm	0.13m	0.13m
Freq(MHz)	214	185	250	137.7	556
Gate counts	208k	540k	122k	120k	122k
Cycle	45.5k	22.3k	193k	340k	562k
Time(μs)	211	120	770	2680	1010
kP/s	4.74k	8.33k	1.30k	0.37k	0.99k
AT$^{\alpha}$	1	2.18	3.16	7.47	2.86
Resistance	SPA DPA Double	SPA	No	No	No

$^{\alpha}$ AT= area-time product= Gate counts * Time *0.13μs/Technology.

Only [8] has higher speed than ours, since it adopts full-word length multipliers without too much optimization, which brings high consumption of area and power. Reference [9] has the shortest critical path delay by a pipelined carry chain of systolic array. But the flexibility comes at a costly price of 1.01 ms for one scalar multiplication. Both of [4] and [5] have area of about 120k gate counts, but [9]'s performance is much better. This is because the former adopts a radix-4 unified division unit for finite field operations, which has been selectively learned in our design. Besides, three of them don't take any security consideration into their design. It should be noted that absolutely fair comparison can't be promised due to different backgrounds, while area-time product provides the most objective assessment standards. Although costing much for comprehensive resistant countermeasures, our architecture outperforms other ones with 2∼6 times better AT. Besides the bottom-up optimization in both algorithm and hardware schedule, our all-new isochronous processor and configurable modular multiplication units also play an important role.

V. Conclusion

This paper presents a high-performance implementation of point multiplication for elliptic curve cryptography over SCA-256. To resist SPA and double attack without reducing speed, for the first time, we modify the Montgomery ladder-based algorithm to a parallel processor of point addition and double. By reducing a fast reduction scheme, modular multiplication unit is designed to be hardware configurable architecture of two-pipelined stage. Based on this, the well-designed PA and PD are strictly isochronous and achieve our all-new isochronous MLSM perfectly. Random cycles are also inserted to resist DPA. Synthesize results show that we achieve a 256-bit point multiplication of only 211μs and 8.5μJ at 214MHz, and it can effective resist SPA, DPA and double attack. Comparing with related works, our architecture offers not only the superior area-time product but also the best comprehensive security.

Acknowledgment

This work is supported by the National Natural Science Foundation (NO.61472208) and the National Science and Technology Major Projects of China (No. 2014ZX01032401). We would also like to thank Zhenwei Zhao for providing suggestions and assistance.

References

[1] State Cryptography Administration of China, "Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves," 2010.
[2] D. Hankerson, A. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography," Springer, New York, 2004.
[3] Zhenwei Zhao, Guoqiang Bai, "Ultra High-Speed SM2 ASIC Implementation," unpublished.
[4] Chen Y L, Lee J W, Liu P C, et al, "A dual-field elliptic curve cryptographic processor with a radix-4 unified division unit[C]," Circuits and Systems (ISCAS), 2011 IEEE International Symposium on. IEEE, 2011: 713-716.
[5] A. Satoh, K. Takano, "A scalable dual-field elliptic curve cryptographic processor," IEEE Transactions on Computers.52(4), 449C460(2003).
[6] Kendall Ananyi, Hamad Alrimeih, and Daler Rakhmatov, "Flexible Hardware Processor for Elliptic Curve Cryptography Over NIST Prime Fields," IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, 17(8), 1099- 1112(2009).
[7] Lee, Jen-Wei, et al, "Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture." Very Large Scale Integration Systems IEEE Transactions on 22.1(2014):49 - 61.
[8] Chung, Szu-Chi, Jen-Wei Lee, Chang, Hsie-Chia, Chen-Yi Lee, "A high-performance elliptic curve cryptographic processor over GF(p) with SPA resistance," Circuits and Systems (ISCAS), 2012 IEEE International Symposium on , vol., no., pp.1456,1459, 20-23 May 2012
[9] Chen G, Bai G, Chen H, "A high-performance elliptic curve cryptographic processor for general curves over GF (p) based on a systolic arithmetic-unit[J]," Circuits and Systems II: Express Briefs, IEEE
[10] J.-Y. Lai and C.-T. Huang, "Energy-adaptive dual-field processor for high-performance elliptic curve cryptographic applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.19, no.8, pp. 1512-1517, Aug.2011.
[11] Fan J, Guo X, De Mulder E, et al, "State of the art of secure ECC implementations: a survey on known side-channel attacks and countermeasures[J]," 2010 IEEE International Symposium on, 2010:76-87.
[12] Fan J, Verbauwhede I, "An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost[J]," Lecture Notes in Computer Science, 2012, 6805:265-282.