Elliptic Curve Scalar Multiplication Algorithm Based on Side Channel Atomic Block over GF(2^m)

Shuang-Gen Liu, Yan-Yan Hu, and Lan Wei
(Corresponding author: Shuang-Gen Liu)

School of Cyberspace Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
Email: liusgxupt@163.com

(Received Feb. 28, 2020; Revised and Accepted Dec. 10, 2020; First Online May 30, 2021)

Abstract

Elliptic Curve Cryptography (ECC) has become one of the research hotspots in cryptography in recent years. Scalar multiplication is the most crucial operation in ECC, and it largely determines the efficiency of ECC. To improve ECC's speed, we propose a new secure and efficient scalar multiplication algorithm on elliptic curves over GF(2^m). In addition, we present the new composite operation formulas $3P_1$ and $2P_1 + P_2$ using only x-coordinate, where P_1 and P_2 are points on an elliptic curve. To ensure the safety and efficiency of the proposed algorithm, we constitute an atomic block by adding dummy operations and using the Montgomery trick.

Keywords: Elliptic Curve Cryptography; Scalar Multiplication; Side Channel Atomic Block; Simple Power Analysis

1 Introduction

As the hacking techniques become more and more powerful, safe and efficient encryption technology is needed. Since Miller [17] and Koblitz [9] independently proposed Elliptic Curve Cryptography (ECC) in 1985, it has become one of the research hotspots in the field of cryptography due to its short key and high security. ECC can provide the same functions as the RSA cryptosystem and it requires a shorter key length than RSA under the same security. It is generally used for digital signature, authentication, encryption, decryption [8, 19, 25]. Because of its advantages in security, encryption and decryption performance, and space consumption, ECC has a wide range of applications, such as transport layer security (TLS), cryptocurrency, SM2 public key cryptography and government agencies, etc. Besides, it is especially suitable for environments with limited storage resources, such as smart cards and secure storage chips.

In ECC, it is easy to obtain the point Q when $Q = kP$, and the number k and point P are given. But it is difficult to find k when point P and point Q are given. This is the classical discrete logarithm problem (DLP). ECC uses this feature to encrypt where point Q is the public key, big number k is the private key and point P is the base point on an elliptic curve. The most crucial operation in ECC is scalar multiplication kP that largely determines the speed of ECC. There are two main methods to improve the efficiency of scalar multiplication. The first method is to reduce computation by optimizing the bottom arithmetic formulas, such as reducing the number of field inversion operations by transforming coordinates. The second method is to decrease the number of point addition and doubling in the scalar multiplication algorithm by studying the expanded form of k, such as double-base chain [27] and symmetric ternary form (STF) [13].

Side channel analysis (SCA) is a method to attack the cryptographic devices by analyzing the leaked side channel information such as time consumption, power consumption or electromagnetic radiation during the operation of cryptographic devices [24]. Power analysis is a form of SCA. It is an attack by collecting power consumption information generated by cryptographic devices or cryptographic chips during encryption, decryption or signature operations, and analyzing the key by using statistics, cryptography and other relevant knowledge. Power analysis can be divided into simple power analysis (SPA) and differential power analysis (DPA). SPA has a direct threat to cryptographic devices. It can directly analyze the power information collected during the execution of cryptographic algorithm. When the device performs encryption or decryption, the key can be derived from the difference in power consumption trajectories. The key in this paper refers to the private key k.

In 1987, the Montgomery algorithm was proposed by Montgomery [18]. The basic idea is that each loop has a point addition and doubling so that the energy consumed by each loop is basically the same. In 1999, Lopez and Dahab [16] optimized the Montgomery ladder algorithm on elliptic curves over GF(2^m). The new point addition and doubling formulas eliminated the calculation of y-coordinate, which improved the calculation speed of the algorithm. In 2008, the new point addition and doubling formulas proposed by Yu *et al.* [**?**] not only omitted the y-coordinate but also dislodged the field inversion operation. In 2013, Sung *et al.* [5] posed the new composite formulas $4P_1$, $3P_1 + P_2$ and $2P_1 + 2P_2$ with only x-coordinate, and presented the extended quaternary Montgomery ladder algorithm over GF(2^m). In 2016, Lai and Zhang [10] proposed Co_Z point addition algorithm, conjugate point addition algorithm and point doubling-point addition algorithm with omitting Z-coordinate on Hessian elliptic curves and applied them to the traditional Montgomery ladder algorithm. In 2017, Yu *et al.* [26] optimized the Montgomery algorithm using the Co_Z technique in projective coordinates over GF(3^m). In 2019, Liu *et al.* [14] proposed the ternary Montgomery ladder algorithm, which combines the original Montgomery ladder algorithm with the ternary representation of the scalar k.

To obtain a safe and efficient scalar multiplication algorithm, we first propose the new composite operation formulas $3P_1$ and $2P_1 + P_2$ using only x-coordinate in affine coordinate system to reduce the bottom field operations and we apply them to the ternary Montgomery ladder algorithm. Then we constitute an atomic block by adding dummy operations to the proposed composite operation formulas to prevent SPA. Last, we use Montgomery trick in the atomic block to optimize the computational cost, which can decrease the number of field inversion operations.

The rest of this paper is presented as follows. In section II, we briefly introduce Elliptic Curve Cryptography and the Montgomery ladder algorithm. In section III, we give a detailed presentation on new composite operation formulas and the anti-SPA scalar multiplication algorithm based on side channel atomic block. In section IV, we compare the performance of the proposed algorithm with existing algorithms.

2 Elliptic Curve Cryptography and Montgomery Ladder Algorithm

2.1 Elliptic Curve Cryptography

Definition 1. *The equation of a non-super singular elliptic curve E over GF(2^m) is given as follows:*

$$E/GF(2^m) : y^2 + xy = x^3 + ax^2 + b. \tag{1}$$

with $a, b \in GF(2^m), b \neq 0$. All points on E and the infinity point $\mathcal{O}$ form an abelian group. Assume $P_1 = (x_1, y_1) \in E(GF(2^m))$, $P_2 = (x_2, y_2) \in E(GF(2^m))$, $-P_1 = (x_1, x_1 + y_1)$ and $P_2 \neq -P_1$.

If $P_1 \neq P_2$, $P_3 = P_1+P_2 = (x_3, y_3)$, then point addition operation:

$$\begin{cases} x_3 = (\dfrac{y_1 + y_2}{x_1 + x_2})^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a \\ y_3 = \dfrac{y_1 + y_2}{x_1 + x_2}(x_1 + x_3) + x_3 + y_1 \end{cases} \tag{2}$$

If $P_1 = P_2$, $P_3 = 2P_1 = (x_3, y_3)$, then point doubling operation:

$$\begin{cases} x_3 = (x_1 + \dfrac{y_1}{x_1})^2 + \dfrac{y_1}{x_1} + x_2 + a \\ y_3 = (x_1 + \dfrac{y_1}{x_1})(x_1 + x_3) + x_3 + y_1 \end{cases} \tag{3}$$

It can be seen that the computational costs of point addition and doubling are both $1I + 2M + 1S$, where I, M, S are the representations of field inversion, field multiplication and field squaring, respectively.

2.2 Montgomery Ladder Algorithm

The Montgomery algorithm was initially proposed to improve the speed of scalar multiplication. The left-to-right Montgomery ladder algorithm [20] is described by Algorithm 1, which is a classical way to compute the scalar multiplication.

Algorithm 1 Left-To-Right Montgomery Ladder Algorithm

```
 1: Input: $P = (x, y) \in E(GF(2^m))$, and $k = (k_{n-1}k_{n-2} \cdots k_1k_0)_2$
 2: Output: $Q = kP \in E(GF(2^m))$
 3: $R_0 = P, R_1 = 2P$
 4: for $i \leq n - 2, \cdots, 0$ do
 5:     if $k_i = 1$ then
 6:         $R_0 = R_0 + R_1, R_1 = 2R_1$
 7:     else if $k_i = 0$ then
 8:         $R_1 = R_0 + R_1, R_0 = 2R_0$
 9:     end if
10: end for
11: Return $Q = R_0$
12: End
```

Based on the original Montgomery ladder algorithm, Liu *et al.* [14] proposed the ternary Montgomery ladder algorithm, which is described by Algorithm 2.

3 New Algorithm Based on the Ternary Montgomery Ladder Algorithm

3.1 Composite Operation Formulas

Improving the performance of the Montgomery ladder algorithm by using only x-coordinate method was first

Algorithm 2 The Ternary Montgomery Ladder Algorithm

```
1: Input: $P = (x, y) \in E(GF(2^m))$, and $k = (k_{n-1}k_{n-2}\cdots k_1k_0)_3$, where $k_{n-1} = 1$ or $2$
2: Output: $Q = kP \in E(GF(2^m))$
3: $R_0 = k_{n-1}P, R_1 = (k_{n-1}+1)P$
4: for $i \leq n-2, \cdots, 0$ do
5:     if $k_i = 0$ then
6:         $R_2 = 3R_0, R_1 = 2R_0 + R_1$
7:     else if $k_i = 1$ then
8:         $R_2 = 2R_0 + R_1, R_1 = 2R_1 + R_0$
9:     else if $k_i = 2$ then
10:        $R_2 = 2R_1 + R_0, R_1 = 3R_1$
11:    end if
12:    $R_0 = R_2$
13: end for
14: Return $Q = R_0$
15: End
```

introduced by Lopez & Dahab [16]. Then several x-coordinate-only methods were presented [5, 22, 28]. Assume P_i is a point on an elliptic curve E. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $-P_1 = (x_1, x_1 + y_1)$, $P_2 - P_1 = P = (x, y)$, and $P_2 \neq -P_1$, then we can obtain

$$x_3 = \begin{cases} x + \frac{x_1}{x_1+x_2} + \left(\frac{x_1}{x_1+x_2}\right)^2 & P_1 \neq P_2 \\ x_1^2 + \frac{b}{x_1^2} & P_1 = P_2 \end{cases} \quad (4)$$

The formula for restoring the y coordinate at the last step is

$$y_1 = (x_1 + x)\{(x_1 + x)(x_2 + x) + x^2 + y\}/(x + y) \quad (5)$$

It can be seen from Equation (4) that the costs of both two operations are $1I + 1M + 1S$. Based on the idea of Lopez & Dahab, this paper proposes two composite operation formulas $3P_1$ and $2P_1 + P_2$.

Theorem 1. *Let $P_1 = (x_1, y_1)$ be a point on an elliptic curve E over $GF(2^m)$. Then, x_{3P_1} can be gained:*

$$x_{3P_1} = x_1 + \frac{{x_1}^3}{{x_1}^4 + {x_1}^3 + b} + \left(\frac{{x_1}^3}{{x_1}^4 + {x_1}^3 + b}\right)^2 \quad (6)$$

with cost $1I + 1M + 3S + 1C$, where C is the representation of field cubing.

Proof. Let $3P_1$ be computed as $2P_1 + P_1$. Equation (4) gives

$$x_{3P_1} = x_1 + \frac{x_{P_1}}{x_{P_1} + x_{2P_1}} + \left(\frac{x_{P_1}}{x_{P_1} + x_{2P_1}}\right)^2 \quad (7)$$

Then, we obtain Equation (6). □

Theorem 2. *Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ be points on an elliptic curve E over $GF(2^m)$. Then, $x_{2P_1+P_2}$ can be gained:*

$$x_{2P_1+P_2} = x_2 + \frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2 + x_1x_2} + \left(\frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2 + x_1x_2}\right)^2 \quad (8)$$

with cost $1I + 2S + 4M$.

Proof. Let $2P_1 + P_2$ be computed as $(P_1 + P_2) + P_1$ and $P_2 - P_1 = P(x, y)$ which is an input. Equation (4) gives

$$x_{2P_1+P_2} = x_2 + \frac{x_{P_1}}{x_{P_1} + x_{P_1+P_2}} + \left(\frac{x_{P_1}}{x_{P_1} + x_{P_1+P_2}}\right)^2 \quad (9)$$

Then, we obtain Equation (8). □

The probability that k_i is equal to 0, 1, and 2 is 1/3 [11]. When Algorithm 2 is computed by Equation (6) and Equation (8), the average calculation costs are $2I + 6M + 14/3S + 2/3C$ per loop.

3.2 New Algorithm Based on Side Channel Atomic Block

In view of SCA attack, in 2004, Mames, ciet and joye proposed a method that almost does not increase the amount of computation: Side channel atomic block method [4]. Its main idea is to decompose the operations on elliptic curves into a series of indistinguishable atomic blocks with multiple side channels. The general method is to add dummy operations so that there is no difference in the side channel analysis of different execution processes.

In this paper, as can be seen from Algorithm 2 that the discrimination of each loop is $3P_1$ and $2P_1 + P_2$. To make it anti-SPA, we can add some dummy operations to $3P_1$ and $2P_1 + P_2$ to make the costs of $3P_1$ and $2P_1 + P_2$ indistinguishable so that the amount of calculations in each loop is exactly the same. In this way, the scalar multiplication can be represented as a series of indistinguishable atomic blocks of code, so the attacker cannot obtain the side channel information by SPA attack.

The Montgomery trick is an efficient way to improve performance by computing field inversions simultaneously. For instance, a^{-1} and b^{-1} can be computed as $a^{-1} = (ab)^{-1} \cdot b$, $b^{-1} = (ab)^{-1} \cdot a$. It converts two field inversion operations into one field inversion operation and three field multiplication operations, which can save $1I - 3M$ calculation costs per loop. Therefore, we apply the Montgomery trick to Equation (6) and Equation (8) in the atomic block to reduce the amount of field inversion operations to optimize the algorithm.

As stated above, we constitute the atomic block by adding dummy operations in each loop to make it anti-SPA and using Montgomery trick to reduce the amount of field inversion operations. Table 1, called the atomic block elliptic curve triple and double-and-add, *i.e.* AETD, describes the atomic block in detail.

In the upper section, Algorithm 2 can be computed efficiently by using the proposed composite operation formulas Equation (6) and Equation (8), with cost $2I+14/3S+6M+2/3C$ per loop. However, the computation costs of AETD just require $1I+4S+10M$. Applying AETD to Algorithm 2, Algorithm 3 is obtained. Algorithm 3 saves $I+2/3S-4M+2/3C$ compared with Algorithm 2 computed by Equation (6) and Equation (8), and saves $3I-2M$ compared with Algorithm 2 computed by Equation (2) and Equation (3) in each loop. From Algorithm 3, we can conclude that only one atomic block is used in each loop, so each loop requires the same amount of calculations regardless of the value of k_i.

Algorithm 3 Anti-SPA Scalar Multiplication Algorithm Based On Side Channel Atomic Block

```
1: Input: $P=(x,y)\in E(GF(2^m))$, and $k=(k_{n-1}k_{n-2}\cdots k_1k_0)_3$, where $k_{n-1}=1$ or 2
2: Output: $Q=kP\in E(GF(2^m))$
3: $R_0=k_{n-1}P, R_1=(k_{n-1}+1)P$
4: for $i\leq n-2,\cdots,0$ do
5:     $(R_0,R_1)=AETD[k_i](R_0,R_1)$
6: end for
7: Return $Q=R_0$
8: End
```

4 Performance Analysis

4.1 Security Analysis

In ECC, if a scalar multiplication algorithm has different power consumption according to k_i, it is vulnerable to SPA. In other words, if the algorithm has the same power consumption regardless of the value of k_i, it is resistant to SPA. Therefore, all countermeasures against SPA have to modify the algorithm to obtain a uniform power consumption trace. In general, there are three main ways to anti-SPA. The first way is uniform algorithm behavior, such as Montgomery ladder algorithm. The second way is uniform point addition and doubling formulas, such as Edwards curve [2]. The third way is to add dummy field operations [6].

To improve the efficiency of the ternary Montgomery ladder algorithm, the composite operation formulas $3P_1$ and $2P_1+P_2$ are proposed. However, the power consumption of $3P_1$ and $2P_1+P_2$ is different. Algorithm 2 computes $3P_1$ and $2P_1+P_2$ when k_i is equal to 0 or 2, while it computes $2P_1+P_2$ twice when k_i is equal to 1. SPA gains the key according to the peak shape of the energy graph [12], so it is easy to obtain the value of k_i by observing the power consumption curve leaked during execution of the algorithm. Therefore, we adopt the third way to add dummy field operations to constitute an atomic block. It can be seen from Table 1 and Algorithm 3 that the field operation of each step of every atomic block is the same and only one atomic block is used in each loop, so the power consumed by each loop is the same whatever $k_i=0$, 1, or 2, which is secure to resist SPA. In addition, Algorithm 3 can also resist DPA so long as randomize the scalar k.

4.2 Efficiency Analysis

Because the extra calculations of algorithms are negligible, in this paper, we mainly compare the calculations of main iteration of algorithms. In this section, the efficiency of the proposed composite operation formulas and Algorithm 3 is analyzed.

Table 2 shows the computation costs of Algorithm 2 under different calculation formulas. From it, we can draw the conclusion that Algorithm 2 can be computed efficiently by using Equation (6) and Equation (8) proposed in this paper. It requires $2I+14/3S+6M+2/3C$ on average, with saving $2I-2M-2/3S-2/3C$ than Equation (4) and saving $2I+2M-2/3S-2/3C$ than Equation (2) and Equation (3) in each loop.

Given an integer k, assume that $m=\lceil\log_3 k\rceil$ is the length of the ternary representation and $n=\lceil\log_2 k\rceil$ is the length of the binary representation, $m=n\log_3 2$, *i.e.* 160-binary is equivalent to 101-ternary and 192, 256, 600-binary [21] is equivalent to 122, 162, 379-ternary, respectively. We suppose $n=160$ bits, $m=101$ bits. According to the experiment of Bernstein [3], we assume $I/M=8$, $S/M=0.8$.

Table 3 shows the comparison of Algorithm 3 and existing algorithms over GF(2^m). It can be seen that Algorithm 3 has a good improved efficiency compared with the algorithms of [12, 23] and [15]. In comparison, the improved efficiency of Algorithm 3 is 13.6%, 33.9%, 8.7%, 13.4%, 1.6%, and 15.4%, respectively.

In order to analyze the dynamic changes of the improved efficiency of Algorithm 3 than existing algorithms, we suppose

$$I/M=\beta \tag{10}$$

$S/M=0.8$. The improved efficiency of Algorithm 3, *i.e.* ε can be given as follows:

$$\varepsilon=1-\frac{(m-1)(\#I_1+\#M_1)}{\ell(\#I_2+\#M_2)} \tag{11}$$

$(\#I_1+\#M_1)$ represents the amount of calculations of Algorithm 3 per loop, and $(\#I_2+\#M_2)$ represents the amount of calculations of existing algorithms in each loop. $(m-1)$ and ℓ indicate the number of iterations of Algorithm 3 and existing algorithms.

Field inversion operations can be efficiently computed by the Extended Euclidean Algorithm (EEA) over GF(2^m), which uses $gcd(a,b)=gcd(b+ca,a)$ for all binary polynomials. According to [1], when the field size is 163 bits, performance of a field inversion operation using the EEA is equal to about 6.67-10.33 field multiplication operations in binary field, which means β is about 6.67-10.33.

Figure 1 shows the comparison of Algorithm 3 and existing algorithms. I/M is the x-axis and the improved

Table 1: The atomic block elliptic curve triple and double-and-add (AETD)

Input:$T_1 = P_1 = x_1, T_2 = P_2 = x_2, T_3 = P = x$ Output:$(3P_1, 2P_1 + P_2)$ or $(2P_1 + P_2, 2P_2 + P_1)$ or $(2P_2 + P_1, 3P_2)$		
$k_i = 0$	$k_i = 1$	$k_i = 2$
$(T_1, T_2) = (3P_1, 2P_1 + P_2)$	(T_1, T_2)=$(2P_1 + P_2, 2P_2 + P_1)$	$(T_1, T_2) = (2P_2 + P_1, 3P_2)$
$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$	$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$	$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$
$T_5 \leftarrow {T_1}^2({x_1}^2)$	$T_5 \leftarrow {T_4}^2((x_1 + x_2)^2)$	$T_5 \leftarrow {T_4}^2((x_1 + x_2)^2)$
$T_6 \leftarrow T_3 + T_2(dummy)$	$T_6 \leftarrow T_3 + T_2(x + x_2)$	$T_6 \leftarrow T_3 + T_2(x + x_2)$
$T_6 \leftarrow T_5 \cdot T_5({x_1}^4)$	$T_6 \leftarrow T_6 \cdot T_5((x + x_2)(x_1 + x_2)^2)$	$T_6 \leftarrow T_6 \cdot T_5((x + x_2)(x_1 + x_2)^2)$
$T_7 \leftarrow T_1 \cdot T_5({x_1}^3)$	$T_7 \leftarrow T_2 \cdot T_5(x_2(x_1 + x_2)^2)$	$T_7 \leftarrow T_2 \cdot T_5(x_2(x_1 + x_2)^2)$
$T_5 \leftarrow b$	$T_4 \leftarrow b$	$T_5 \leftarrow b$
$T_8 \leftarrow T_1 \cdot T_2(x_1 x_2)$	$T_8 \leftarrow T_1 \cdot T_2(x_1 x_2)$	$T_8 \leftarrow T_1 \cdot T_2(x_1 x_2)$
$T_5 \leftarrow T_5 + T_7(b + {x_1}^3)$	$T_4 \leftarrow T_5 + T_7(dummy)$	$T_4 \leftarrow T_5 + T_7(dummy)$
$T_5 \leftarrow T_6 + T_5(A)$	$T_4 \leftarrow T_6 + T_8(A)$	$T_4 \leftarrow T_6 + T_8(A)$
$T_6 \leftarrow T_3 + T_1(x + x_1)$	$T_6 \leftarrow T_3 + T_1(x + x_1)$	$T_6 \leftarrow T_3 + T_1(dummy)$
$T_4 \leftarrow {T_4}^2((x_1 + x_2)^2)$	$T_3 \leftarrow {T_3}^2(dummy)$	$T_6 \leftarrow {T_2}^2({x_2}^2)$
$T_9 \leftarrow T_1 \cdot T_4(x_1(x_1 + x_2)^2)$	$T_9 \leftarrow T_1 \cdot T_5(x_1(x_1 + x_2)^2)$	$T_9 \leftarrow T_2 \cdot T_6({x_2}^3)$
$T_4 \leftarrow T_6 \cdot T_4((x + x_1)(x_1 + x_2)^2)$	$T_5 \leftarrow T_6 \cdot T_5((x + x_1)(x_1 + x_2)^2)$	$T_6 \leftarrow T_6 \cdot T_6({x_2}^4)$
$T_6 \leftarrow T_6 + T_9(dummy)$	$T_6 \leftarrow T_6 + T_9(dummy)$	$T_6 \leftarrow T_6 + T_9$
$T_4 \leftarrow T_4 + T_8(B)$	$T_5 \leftarrow T_5 + T_8(B)$	$T_5 \leftarrow T_6 + T_5(B)$
$T_6 \leftarrow T_5 \cdot T_4(AB)$	$T_6 \leftarrow T_4 \cdot T_5(AB)$	$T_6 \leftarrow T_5 \cdot T_4(AB)$
$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$	$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$	$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$
$T_5 \leftarrow T_6 \cdot T_5(B^{-1})$	$T_4 \leftarrow T_6 \cdot T_4(B^{-1})$	$T_5 \leftarrow T_6 \cdot T_5(A^{-1})$
$T_4 \leftarrow T_6 \cdot T_4(A^{-1})$	$T_5 \leftarrow T_6 \cdot T_5(A^{-1})$	$T_4 \leftarrow T_6 \cdot T_4(B^{-1})$
$T_4 \leftarrow T_4 \cdot T_7(A^{-1}{x_1}^3)$	$T_5 \leftarrow T_5 \cdot T_7(A^{-1}x_2(x_1 + x_2)^2)$	$T_4 \leftarrow T_4 \cdot T_7(B^{-1}x_2(x_1 + x_2)^2)$
$T_6 \leftarrow {T_4}^2$	$T_6 \leftarrow {T_5}^2$	$T_6 \leftarrow {T_4}^2$
$T_4 \leftarrow T_4 + T_6$	$T_5 \leftarrow T_5 + T_6$	$T_4 \leftarrow T_4 + T_6$
$T_4 \leftarrow T_1 + T_4(3P_1)$	$T_5 \leftarrow T_1 + T_5(2P_2 + P_1)$	$T_4 \leftarrow T_1 + T_4(2P_2 + P_1)$
$T_5 \leftarrow T_5 \cdot T_9(B^{-1}x_1(x_1 + x_2)^2)$	$T_4 \leftarrow T_4 \cdot T_9(B^{-1}x_1(x_1 + x_2)^2)$	$T_5 \leftarrow T_5 \cdot T_9(A^{-1}{x_2}^3)$
$T_9 \leftarrow {T_5}^2$	$T_9 \leftarrow {T_4}^2$	$T_9 \leftarrow {T_5}^2$
$T_5 \leftarrow T_5 + T_9$	$T_4 \leftarrow T_4 + T_9$	$T_5 \leftarrow T_5 + T_9$
$T_2 \leftarrow T_2 + T_5(2P_1 + P_2)$	$T_1 \leftarrow T_2 + T_4(2P_1 + P_2)$	$T_2 \leftarrow T_2 + T_5(3P_2)$
$T_1 \leftarrow T_4(3P_1)$	$T_2 \leftarrow T_5(2P_2 + P_1)$	$T_1 \leftarrow T_4(2P_2 + P_1)$
$(A = {x_1}^4 + {x_1}^3 + b;$	$(A = (x + x_2)(x_1 + x_2)^2 + x_1 x_2;$	$(A = (x + x_2)(x_1 + x_2)^2 + x_1 x_2;$
$B = (x + x_1)(x_1 + x_2)^2 + x_1 x_2)$	$B = (x + x_1)(x_1 + x_2)^2 + x_1 x_2)$	$B = {x_2}^4 + {x_2}^3 + b)$

Table 2: The computation costs of Algorithm 2 under different calculation formulas

Formulas	$3P_1$	$2P_1 + P_2$	Average costs of main iteration	Anti-SPA
(2)(3)	$2I + 4M + 2S$	$2I + 4M + 2S$	$4I + 8M + 4S$	yes
(4)	$2I + 2M + 2S$	$2I + 2M + 2S$	$4I + 4M + 4S$	yes
(6)(8)	$1I + 1M + 3S + 1C$	$1I + 4M + 2S$	$2I + 6M + 14/3S + 2/3C$	no

Table 3: The computation costs of different scalar multiplication algorithms

Algorithm	Total costs of main iteration	$n = 160$bits,$m = 101$bits	Anti-SPA
Unprotected NAF [23]	$(10M + 20/3S)n$	$2453.3M$	No
Co_Z protected NAF [23]	$(85/6M + 265/36S)n$	$3208.9M$	Yes
Mont2 [12]	$3(1I + 1M + 1S)(n/2 - 1)$	$2322.6M$	No
Mont3 [12]	$37/24(1I + 1M + 1S)(n + 2)$	$2447.6M$	No
Co_Z STF [15]	$(52/3M + 5S)m$	$2154.7M$	No
Co_Z Anti-SPA STF [15]	$(20M + 6S)m$	$2504.8M$	Yes
Algorithm 3	$(1I + 10M + 4S)(m - 1)$	$2120M$	Yes

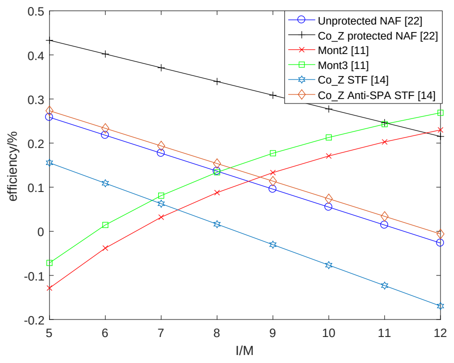

Figure 1: The comparison of Algorithm 3 and existing algorithms

efficiency of Algorithm 3 than existing algorithms is the y-axis. When $I/M = 8$, Table 3 can be obtained. It can be seen from Figure 1, for algorithms of [23] and [15], the improved efficiency of Algorithm 3, *i.e.* ε, decreases linearly as β increases. For algorithms of [12], ε increases as β increases and the larger β, the slower ε increases. When β is 6.67-10.33, Algorithm 3 is more efficient than other algorithms except for Co_Z STF algorithm [15]. However, Algorithm 3 performs better than Co_Z STF algorithm [15] when β is less than 8.3. In summary, Algorithm 3 has a good improvement in efficiency compared with existing algorithms.

5 Conclusions

In this paper, we proposed an anti-SPA scalar multiplication algorithm based on side channel atomic block over GF(2^m). Besides, we have optimized the bottom field operations by presenting new composite operation formulas $3P_1$ and $2P_1 + P_2$. Figure 1 intuitively shows the comparison of the proposed algorithm and existing algorithms. When $I/M = 8$, it can be seen from Table 3 that the proposed algorithm is more efficient than existing algorithms, ranging from 1.6% to 33.9%. Then we can apply it to the specific environments, such as wireless sensor networks with resource-limited. Next, we will try to transform the coordinate to optimize the proposed composite operation formulas and then propose a more efficient scalar multiplication algorithm.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61872058), the Key Research and Development Program of Shaanxi (Program No.2021NY-211). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Avanzi and N. Thériault, "Effects of optimizations for software implementations of small binary field arithmetic," in *International Workshop on the Arithmetic of Finite Fields*, pp. 69–84, 2007.

[2] D. J. Bernstein and T. Lange, "Faster addition and doubling on elliptic curves," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 1–20, 2007.

[3] D. J. Bernstein and T. Lange, "Analysis and optimization of elliptic-curve single-scalar multiplication," *Contemporary Mathematics*, vol. 461, no. 461, pp. 1, 2008.

[4] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity," *IEEE Transactions on computers*, vol. 53, no. 6, pp. 760–768, 2004.

[5] S. M. Cho, S. C. Seo, T. H. Kim, Y. H. Park, and S. Hong, "Extended elliptic curve Montgomery ladder algorithm over binary fields with resistance to simple power analysis," *Information Sciences*, vol. 245, pp. 304–312, 2013.

[6] L. Elmegaard-Fessel, "Efficient scalar multiplication and security against power analysis in cryptosystems based on the NIST elliptic curves over prime fields," *IACR Cryptology ePrint Archive*, vol. 2006, pp. 313, 2006.

[7] R. R. Farashahi, H. F. Wu, and C. A. Zhao, "Efficient arithmetic on elliptic curves over fields of characteristic three," in *International Conference on Selected Areas in Cryptography*, pp. 135–148, 2012.

[8] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[9] N. KOBLITZ, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[10] Z. X. Lai and Z. J. Zhang, "Scalar multiplication on hessian curves based on Co_Z operations," *Bulletin of Science and Technology (in Chinese)*, vol. 32, no. 2, pp. 28–33, 2016.

[11] L. Li, "Research on the ternary algorithm in the elliptic curve operations," *Journal of Network Safety Technology and Application (in Chinese)*, no. 11, pp. 94–96, 2015.

[12] Y. Li, J. L. Wang, X. W. Zeng, and X. Z. Ye, "A segmented Montgomery scalar multiplication algorithm with resistance to simple power analysis," *Computer Engineering and Science (in Chinese)*, vol. 39, no. 1, pp. 92–102, 2017.

[13] H. Z. Liu, Q. H. Dong, and Y. B. Li, "Efficient ECC scalar multiplication algorithm based on symmetric ternary in wireless sensor networks," in *Progress in Electromagnetics Research Symposium-Fall*, pp. 879–885, 2017.

[14] S. G. Liu, R. R. Wang, Y. Q. Li, and C. L. Zhai, "An improved ternary Montgomery ladder algorithm on

elliptic curves over GF(3ˆm)," *International Journal Network Security*, vol. 21, no. 3, pp. 384–391, 2019.

[15] S. G. Liu, Y. Y. Ding, R. Shi, and S. M. Lu, "Co_Z addition on elliptic curves over finite fields GF(2ˆm)," *Journal of Wuhan University (in Chinese)*, vol. 65, no. 2, pp. 207–212, 2019.

[16] J. López and R. Dahab, "Fast multiplication on elliptic curves over GF(2ˆ m) without precomputation," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 316–327, 1999.

[17] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417–426, 1985.

[18] P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Mathematics of Computation*, vol. 48, no. 177, pp. 243–264, 1987.

[19] J. Moon, D. Lee, and J. Jung, "Improvement of efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 1053–1061, 2017.

[20] T. Oliveira, J. López, and F. Rodríguez-Henríquez, "The Montgomery ladder on binary elliptic curves," *Journal of Cryptographic Engineering*, vol. 8, no. 3, pp. 241–258, 2018.

[21] S. F. Tzeng and M. S. Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem," *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 61–71, 2004.

[22] H. Wang, B. Li, and W. Yu, "Montgomery algorithm on elliptic curves over finite fields of character three," *Journal on Communications (in Chinese)*, vol. 29, no. 10, pp. 25–29, 2008.

[23] J. Wei, X. Liu, H. Liu, and W. Guo, "A low-time-complexity and secure dual-field scalar multiplication based on co-z protected NAF," *IEICE Electronics Express*, vol. 11, no. 11, pp. 20140361–20140361, 2014.

[24] X. S. Yan, X. G. Zhang, H. F. Zhang, and L. Liu, "Countermeasures in CPU for timing and power side channel attack," *DEStech Transactions on Engineering and Technology Research*, 2018. DOI: 10.12783/dtetr/pmsms2018/24880.

[25] J. You, Q. Zhang, C. D'Alves, B. O'Farrell, and C. K. Anand, "Using z14 fused-multiply-add instructions to accelerate elliptic curve cryptography," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pp. 284–291, 2019.

[26] W. Yu, B. Li, K. W. Wang, and W. H. Li, "Co_Z Montgomery algorithm on elliptic cureves over finite fields of characteristic three," *Journal of Computer (in Chinese)*, vol. 40, no. 5, pp. 1121–1131, 2017.

[27] W. Yu, S. A. Musa, and B. Li, "Double-base chains for scalar multiplications on elliptic curves," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 538–565, 2020.

[28] N. Zhang and Q. Q. Peiand G. Z. Xiao, "Elliptic curve scalar multiplication with x-coordinate," *Wuhan University Journal of Natural Sciences*, vol. 12, no. 1, pp. 163–166, 2007.

Biography

Shuang-Gen Liu, born in 1979. He received the PH.D. in cryptography form Xidian University in 2008. He is currently an associate professor with the school of cyber security, Xian University of Posts and Telecommunications, Xi?an, China. He is a member of the China Computer Federation, and a member of the Chinese Association for Cryptologic Research. His recent research interests include crptography and information security.

Yan-Yan Hu, born in 1995. A graduate student of Xi'an University of posts and telecommunications. She is mainly engaged in the research of elliptic curve cryptosystem.

Lan Wei, born in 1998. An undergraduate student of Xi'an University of posts and telecommunications.She is mainly engaged in the research of elliptic curve cryptosystem.