Fast Scalar Multiplication Algorithm Based on Co_Z Operations on Elliptic Curves over GF(3^m)

Shuang-Gen Liu, Shi-Mei Lu, and Rui-Wen Gong
(Corresponding author: Shuang-Gen Liu)
School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
(Email: liusgxupt@163.com)
(Received Jan. 11, 2019; Revised and Accepted June 19, 2019; First Online July 30, 2019)

Abstract

In this paper, the Co_Z idea is adopted to propose the Co_Z point addition formulae in Jacobian projective coordinate and Lopez & Dahab projective coordinate on elliptic curve over GF(3^m) in order to improve the efficiency and safety of scalar multiplication algorithm. Considering the optimized symmetric ternary algorithm for scalar multiplication against Simple Power Attack (SPA), the computational efficiency is increased by 16%, 26% and 10% compared with the balanced ternary scalar multiplication algorithm, the symbolic ternary form(STF) algorithm and the optimized symmetric ternary scalar multiplication algorithm. In addition, the new formula for $3P + Q$ operation in Jacobian projective coordinate is also proposed. The efficiency of improved balanced ternary scalar multiplication algorithm is increased by 7% compared with the previous algorithm.

Keywords: Characteristic Three; Co_Z *Operation; Projective Coordinate; Scalar Multiplication*

1 Introduction

Elliptic curve, as an important issue of algebraic geometry, has been studied for more than 100 years. Until in the year of 1985, Kolitz and Miller introduced it into the field of cryptography and constructed ECC (Elliptic Curve Cryptography). Compared with other cryptosystems, ECC has advantages such as short key, less storage space and low bandwidth requirements, which makes it has a superior development prospect. In 2017, Du [5] constructed an ID-based dynamic group communication sign-cryption scheme by using hyperelliptic curve cryptosystem and ID-based sign-cryption model. Elliptic curve cryptosystem mainly includes elliptic curve key generation, key exchange, encryption, decryption, signature and other algorithms. In these elliptic curve cryptography algorithms, scalar multiplication is the most time-consuming operation. How to improve the efficiency of scalar multiplication on the elliptic curve has been a hot topic of public key cryptography. Scalar multiplication, ie, the computation of the point $kP = P + ... + P$, where k is an integer and P is a point on the elliptic curve. There are two main ways to improve the efficiency of scalar multiplication. On the one hand, it is the effective representation of scalar k, such as Binary Scalar Multiplication (BSM), non-adjacent form (NAF) [18] and Balanced Ternary form (BTF) [4]. On the other hand, it is the improvement of point addition and doubling formulas. Point doubling and addition involve operations over prime or ternary fields, In these operations, it is inversion that is the most time-consuming. We can change the coordinates to reduce the number of field inversions, the commonly used coordinate systems are projective coordinate, Jacobian coordinate [3], and Lopez & Dahab coordinate [12].

In recent years, there have been many studies on elliptic curves over GF(2^m) and GF(p). References [1,9,13] provided the fast scalar multiplication algorithms over GF(p), which accelerate the computing speed of elliptic curve cryptosystem. Hisil *et al.* [8] provided a faster mixed addition on modified Jacobiquartic coordinates, and introduced tripling formulae for different forms. In 2007, Meloni [15] proposed the earliest Co_Z point addition operation on Weierstraβ elliptic curves, which used a small amount of computation to calculate the addition operation of two different points with identical Z coordinate. Goundar *et al.* [7] presented the further Co_Z addition formula for various point additions on Weierstraβ elliptic curves. In 2017, Yu *et al.* [19] proposed the Co_Z montgomery algorithm over the finite field of characteristic three by using the same Z-coordinate. In this paper, we develop the fast Co_Z point addition formulas in different projective coordinate systems over GF(3^m), these new operational formulas are used to optimize the existing scalar multiplication algorithms [4,13,14]. As a result, we get efficient scalar multiplication algorithms based on Co_Z point addition using signed ternary representation.

The organizational structure of the paper is as fol-

lows. The next section introduces the basics of elliptic curves over GF(3^m), the point operations under different projective coordinate systems, and the scalar multiplication algorithm based on symmetric ternary representation. The third section describes the specific steps to improve the calculation formula of point addition in different coordinates by using Co_Z idea. We also optimize $3P+Q$ formula for Jacobian projective coordinate, then the improved scalar multiplication algorithm is proposed. The fourth section gives the performance analysis of the scalar multiplication algorithm combined with new formulas. Finally, the fifth section draws a conclusion.

2 Preliminaries

2.1 Basic Knowledge of Elliptic Curve over GF(3^m)

Definition 1. *An elliptic curve over Fp is given by using the generalized Weierstrass equation:*

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

where $a_1, a_2, a_3, a_4, a_6 \in Fp$, and satisfy the discriminant $\triangle \neq 0$ on elliptic curve E. The condition of $a = 0$ ensures the smoothness of the elliptic curve, that is to say, no point on the curve has two or more tangent lines. If the characteristic of Fp is equal to 3 and $a_1^2 \neq -a_2$, elliptic curve E can be given by:

$$y^2 = x^3 + ax^2 + b,$$

where $a, b \in Fp, a, b \neq 0$, and the curve is nonsuper singular. The basic operations of points on the elliptic curve are defined as follows:

Let $P = (x_1, y_1) \in E(GF(3^m))$, $Q = (x_2, y_2) \in E(GF(3^m))$, $P \neq \pm Q$, then $R = P + Q = (x_3, y_3)$

$$\begin{cases} x_3 = (\frac{y_2-y_1}{x_2-x_1})^2 - x_1 - x_2 - a \\ y_3 = \frac{y_2-y_1}{x_2-x_1}(x_1 - x_3) - y_1. \end{cases}$$

Given the point $P = (x_1, y_1) \in E(GF(3^m))$, its double $R = 2P = (x_3, y_3)$ is obtained by:

$$\begin{cases} x_3 = (\frac{ax_1}{y_1})^2 + x_1 - a \\ y_3 = \frac{ax_1}{y_1}(x_1 - x_3) - y_1 \end{cases} \quad (1)$$

It can be seen from Equation (1) that $1I + 2M + 1S$ is required for point addition. If $a = 1$, point doubling needs $1I + 1S + 2M$. We denote field inversion as I, field squaring as S, field multiplication as M, and field cubing as C.

2.2 Projective Coordinate System

In affine coordinate, point doubling and addition on the elliptic curve over GF(3^m) involve the number of field inversions, which is a quite time-consuming operation. In order to avoid inversion operation, projective coordinate systems (X, Y, Z) are introduced. There are mainly two types of projective coordinate systems discussed in this paper, that is Jacobian projective coordinate and Lopez & Dahab projective coordinate.

2.2.1 Jacobian Projective Coordinate

Let $P = (x, y) \in E(GF(3^m))$, the correspondence between the Jacobian projective coordinate and the affine coordinate is: $(x, y) \longmapsto (\frac{X}{Z^2}, \frac{Y}{Z^3})$.

The curve equation in Jacobian projective coordinate is represented by:

$$Y^2 = X^3 + aX^2Z^2 + bZ^6.$$

Given two points $P = (X_1, Y_1, Z_1)$, $Q = (X_2, Y_2, Z_2) \in E(GF(3^m))$, and $P \neq \pm Q$, the point $R = P + Q = (X_3, Y_3, Z_3)$ is obtained by:

$$\begin{cases} X_3 &= U - (U_1 + U_2)H^2 \\ Y_3 &= -RU + (S_1 + S_2)H^3 \\ Z_3 &= HZ_1Z_2 \end{cases} \quad (2)$$

where $U_1 = X_1Z_2^2; U_2 = X_2Z_1^2; S_1 = Y_1Z_2^3; S_2 = Y_2Z_1^3; H = U_2 - U_1; R = S_2 - S_1; U = R^2 - aZ_3^2$.

Let $P = (X_1, Y_1, Z_1) \in E(GF(3^m))$, its double $R = 2P = (X_3, Y_3, Z_3)$ is obtained by

$$\begin{cases} X_3 &= U + X_1Y_1^2 \\ Y_3 &= -(aZ_1^2X_1)U - Y_1^4 \\ Z_3 &= Y_1Z_1 \end{cases}$$

where $U = (aZ_1^2X_1)^2 - aZ_1^2Y_1^2$.

From the above equations, if $a = 1$, the point addition costs $10M + 3C + 3S$, and the point doubling costs $5M + 3S$. The mixed addition $P + Q$ costs $7M + 2C + 2S$ when $Z_2 = 1$.

2.2.2 Lopez &Dahab Projective Coordinate

In the Lopez &Dahab coordinate, the point addition formula is as follows:

Suppose $P = (X_1, Y_1, Z_1)$, $Q = (X_2, Y_2, Z_2) \in E(GF(3^m))$. If $P \neq \pm Q$, the point $R = P + Q = (X_3, Y_3, Z_3)$ is obtained by:

$$\begin{cases} X_3 &= (A - B)^2 - E(C - D)^2(C + D - aE) \\ Y_3 &= E(C - D)(A - B)[DE(C - D)^2 - X_3] \\ & \quad -Z_3B(C - D)^2 \\ Z_3 &= [E(C - D)]^2 \end{cases} \quad (3)$$

where $A = Y_2Z_1^2; B = Y_1Z_2^2; C = X_2Z_1; D = X_1Z_2; E = Z_1Z_2; F = Z_1^2; G = Z_2^2$.

From Equation (3), the point addition costs $13M + 5S$. The mixed addition $P + Q$ costs $10M + 4S$ when $Z_2 = 1$.

2.3 Scalar Multiplication Algorithm Based on Symmetric Ternary

Definition 2. *An arbitrary positive integer* K *is expressed as* $K = a_n a_{n-1}...a_1 a_0$, *where* $a_n a_{n-1}...a_1 a_0$ *are numbers from -1, 0, 1. It is called symmetric ternary representation.*

In 2017, Liu *et al.* [13] optimized the bottom operation using the Jacobian coordinate on elliptic curves over GF(p). And they also optimized the symmetric ternary representation. In Jacobian coordinate system, point doubling is faster than point addition, so the non-zero Hamming weight is reduced by changing operations, and the efficiency of scalar multiplication is improved by replacing point addition with point doubling. Algorithm 1 describes the optimized symmetric ternary scalar multiplication method.

Algorithm 1 Optimized symmetric ternary scalar multiplication algorithm

Input: $K = (k_{n-1}k_{n-2}...k_1k_0)_3, P(X, Y, Z)$
Output: KP
$i = 0, P_1 = \propto, Q = \propto$
while $i \leq n-1$ **do**
  if $k_i = 1$ **then**
    $Q = Q + P, P = 3P$
  else if $k_i = -1$ **then**
    $Q = Q - P, P = 3P$
  else if $k_i = 2$ **then**
    $P_1 = 2P, Q = Q + P_1, P = 3P$
  else if $k_i = -2$ **then**
    $P_1 = 2P, Q = Q - P_1, P = 3P$
  else
    $P = 3P$
  end if
end while
$i = i + 1$
Return $Q = (X_q, Y_q, Z_q)$
End

According to the analysis, the calculation of Algorithm 1 requires $n(9M+7S)+\frac{n}{6}(4M+6S)+\frac{n}{2}(12M+4S)$.

3 The Improved Scalar Multiplication Algorithm

3.1 Scalar Multiplication Algorithm Based on Co_Z Operation

3.1.1 Co_Z Point Addition in Jacobian Projective Coordinate

Let $P = (X, Y, Z), Q = (X, Y, Z)$ be two points with the same Z-coordinate in Jacobian projective coordinate on elliptic curve over GF(3^m). We perform the Co_Z operation on Equation (2) and obtain the Co_Z point addition formula:

$$\begin{cases} X_3 &= Z^6(Y_2 - Y_1)^2 - aZ_3^2 - Z^6(X_1 + X_2) \cdot (X_2 - X_1)^2 \\ Y_3 &= -Z^3(Y_2 - Y_1)[Z^6(Y_2 - Y_1)^2 - aZ_3^2] \\ & \quad + Z^9(Y_1 + Y_2)(X_2 - X_1)^3 \\ Z_3 &= Z^4(X_2 - X_1). \end{cases}$$

After simplification, we get:

$$\begin{cases} X_3 &= (Y_2 - Y_1)^2 - a[Z(X_1 - X_2)]^2 \\ & \quad -(X_1 + X_2) \cdot (X_2 - X_1)^2 \\ Y_3 &= -(Y_2 - Y_1)[(Y_2 - Y_1)^2 - a[Z(X_2 \\ & \quad -X_1)]^2] + (Y_1 + Y_2)(X_2 - X_1)^3 \\ Z_3 &= Z(X_2 - X_1). \end{cases}$$

Algorithm 2 describes the Co_Z point addition operation in Jacobian projective coordinate, where the coordinate representation of output point satisfies $(X_1(X_2 - X_1)^2, Y_1(X_2 - X_1)^3, Z(X_2 - X_1) \sim (X_1, Y_1, Z))$. In Algorithm 2, it requires $4M + 3S + 1C$ for per bit. In this paper, we take $S = 0.8M, C = 1.37M$. Compared with mixed addition, the reduced cost is about $2M + 1C$.

Algorithm 2 Co_Z point addition algorithm in Jacobian projective coordinate(J-ZADD)

Input: $P(X_1, Y_1, Z), Q(X_2, Y_2, Z)$
Output: $P + Q = (X_3, Y_3, Z_3)$
$A \leftarrow (X_2 - X_1), B \leftarrow Z \cdot A$
$C \leftarrow (Y_2 - Y_1)^2, D \leftarrow (X_2 - X_1)^3$
$X_3 = C - aB^2 - (X_1 + X_2) \cdot (X_2 - X_1)^2$
$Y_3 = -(Y_2 - Y_1) \cdot (C - aB^2) + (Y_1 + Y_2) \cdot (X_2 - X_1)^3$
$Z_3 = Z \cdot A$
Return $Q = (X_3, Y_3, Z_3)$
End

3.1.2 Co_Z Point Addition in Lopez &Dahab Projective Coordinate

Let $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$ be two points with the same Z coordinate in Lopez & Dahab projective coordinate, then the operation formula of $P + Q = (X_3, Y_3, Z_3)$ can be simplified as:

$$\begin{cases} X_3 &= (Y_2 - Y_1)^2 - Z(X_2 - X_1)^2(X_1 + X_2 + aZ) \\ Y_3 &= Z(X_2 - X_1)(Y_2 - Y_1)[X_1Z(X_2 - X_1)^2 - X_3] \\ & \quad - Y_1[Z(X_2 - X_1)^2]^2 \\ Z_3 &= [Z(X_2 - X_1)]^2. \end{cases}$$

In Algorithm 3, the Co_Z point addition operation in Lopez &Dahab projective coordinate is given, which satisfies $(X_1U, Y_1U^2, ZU) \sim (X_1, Y_1, Z)$ in the process of algorithm execution, so $P + Q = (X_3, Y_3, Z_3)$ has the same Z coordinate as input points P and Q.

The computation of Algorithm 3 requires $8M + 3S$ per bit, and the reduced cost is about $5M + 2S$ compared with previous point addition. Compared with mixed addition, the reduced cost is about $3M$.

Although the Algorithm 1 mentioned in section 2.3 improves the efficiency of scalar multiplication algorithm,

Algorithm 3 Co_Z point addition algorithm in Lopez &Dahab projective coordinate(LD-ZADD)

Input: $P(X_1, Y_1, Z), Q(X_2, Y_2, Z)$
Output: $P + Q = (X_3, Y_3, Z_3)$
$S \leftarrow (X_2 - X_1)^2, U \leftarrow Z \cdot S$
$V \leftarrow Z \cdot (X_2 - X_1), W \leftarrow (Y_2 - Y_1)^2$
$X_3 = W - U \cdot (X_1 + X_2 + aZ)$
$Y_3 = V \cdot (Y_2 - Y_1) \cdot (X_1 \cdot U - X_3) - Y_1 \cdot U^2$
$Z_3 = Z \cdot U$
Return $Q = (X_3, Y_3, Z_3)$
End

it cannot resist SPA attack. Therefore, combining with Co_Z point addition operation in Jacobian projective coordinate, we propose an optimized symmetric ternary scalar multiplication algorithm to resist SPA attacks.

Algorithm 4 Optimized symmetric ternary scalar multiplication algorithm against SPA

Input: $P = (X, Y, Z) \in E(GF(3^m)), and \quad k = \sum_{i=0}^{n-1} k_i 3^i, k_i \in (-2, -1, 0, 1, 2)$
Output: kP
$Q = O, Q_1 = P, Q_{-1} = -P, Q_2 = 2P, Q_{-2} = -2P$
for $i = n - 1, ..., 0$ **do**
  $Q = 3Q$
  $Q = J - ZADD(Q, Q_{ki})$
end for
Return Q
End

In Algorithm 4, each loop performs point addition and tripling(T) operation, and two doublings are required in precomputation. Through the analysis, we can get computation complexity of Algorithm 4 about $n(ZA+T)+2D$, where ZA represents Co_Z addition and D represents doubling operation. The implementation of the process in Algorithm 4 is independent of specific location of scalar k, so attacker can't get the information about scalar k through the side channel information. Algorithm 4 can resist the SPA attack. Compared with Algorithm 1, the proposed Algorithm 4 improves security and efficiency of ECC system.

3.2 Tripling-Add Operation in Jacobian Coordinate

Computing $3P + Q$. The Literature [20] proposed $3P + Q$ formula in affine coordinate, in order to avoid inverse operation, let $3P+Q = (X_3, Y_3, Z_3)$, according to $3P+Q$ formula in affine coordinate, we get:

$$\begin{cases} X_3' &= \frac{C^2 - B(DZ_2)^2 - (Z_1^7 AD)^2 (aZ_2^2 - X_2)}{(Z_1^7 ADZ_2)} \\ Y_3' &= \frac{C}{Z_1^7 ADZ_2} (\frac{X_3}{Z_3^2} - \frac{X_2}{Z_2^2} - \frac{Y_2}{Z_2^3}) \end{cases}$$

where $A = a(X_1 + bZ_1^2); B = (X_1^3 + bZ_1^6)^3 - Z_1^{12} a^3 bX_1^3; C = Z_2^3[Y_1^9 - a^3 Z_1^6 Y_1^3 (X_1^3 + bZ_1^6)^2] - Y_2(Z_1^7 A)^3; D = BZ_2^2 - X_2(Z_1^7 A)^2$, Then we take $Z_3 = Z_1^7 ADZ_2$, The formula of $3P + Q$ is derived as followed:

$$\begin{cases} X_3 &= C^2 - B(DZ_2)^2 - (Z_1^7 AD)(aZ_2^2 - X_2) \\ Y_3 &= C[X_3 - X_2(Z_1^7 AD)^2] + Y_2(Z_1^7 AD)^3 \\ Z_3 &= Z_1^7 ADZ_2. \end{cases}$$

By storing the intermediate results and setting $a = 1$, it can be seen that the computation of $3P + Q$ is $23M + 4C + 8S$ instead of $1I + 13M + 5S + 6C$. We use combined tripling-add operation in Jacobian coordinate to replace point addition and tripling operation, and get an improved balanced ternary scalar multiplication algorithm.

Algorithm 5 Improved balanced ternary scalar multiplication algorithm

Input: $P = (X, Y, Z) \in E(GF(3^m)), and \quad k = \sum_{i=0}^{n-1} k_i 3^i, k_i \in (-1, 0, 1)$
Output: kP
$Q \leftarrow O$
for $i = n - 1, ..., 0$ **do**
  if $k_i = 1$ **then**
    $Q = 3Q + P$
  end if
  if $k_i = -1$ **then**
    $Q = 3Q - P$
  end if
  if $k_i = 0$ **then**
    $Q = 3Q$
  end if
end for
Return Q
End

The computation amount of Algorithm 5 is $\frac{1}{3}nT + \frac{2}{3}n \cdot TA$, where TA represents tripling-add operation.

4 Performance Analysis

In this section, we analyze the computational efficiency of improved scalar multiplication algorithm on the elliptic curve over GF(3^m). First, Table 1 shows the comparisons of calculation costs in different coordinate systems. It can be seen from Table 1, the newly proposed Co_Z point addition formula is more efficient than traditional point addition.

In order to analyze the efficiency better, we choose the ternary length of scalar k to be 101 bits, and consider the typical ratio of inversion and multiplication: $I = 8M$. The calculation comparisons of different scalar multiplication algorithms are given in Table 2, it can be seen from Table 2, that compared with other scalar multiplication algorithms, the computational efficiency of our Algorithm 4 is increased by 16%, 26%, 10%, 3%, respectively.

Second, before we analyze the efficiency of $3P + Q$ in Jacobian coordinate, We define α is the ratio of field in-

Table 1: Comparisons of computation costs in different coordinate systems

Operation	**Jacobian projective coordinate**	**Lopez &Dahab projective coordinate**
Point addition	$10M + 3C + 3S$	$13M + 5S$
Mixed addition($Z_2 = 1$)	$7M + 2C + 2S$	$10M + 4S$
Co_Z point addition	$4M + 1C + 3S$	$8M + 3S$
Tripling	$5M + 2S + 5C$	$7M + 1S + 5C$

Table 2: Computation costs of different scalar multiplication algorithms

Algorithm	**Total cost**	**Anti-SPA**	**n=101bits**
BTF Algorithm [4]	$n(I + 4S + 7M) + \frac{2}{3}n(I + S + 2M)$	no	$2565M$
STF Algorithm [14]	$n(I + 4S + 7M) + n(I + S + 2M)$	yes	$2929M$
Algorithm 3 in Ref. [13]	$n(9M + 7S) + \frac{n}{6}(4M + 6S) + \frac{n}{2}(12M + 4S)$	no	$2390M$
Co_Z Montgomery Algorithm(binary) [19]	$n(10M + C + 3S) + I + 10M + S + C$	yes	$2223M$
Proposed Algorithm 4	$n(5M + 2S + 5C) + n(4M + 1C + 3S) + 2(5M + 3S)$	yes	$2158M$

Table 3: Calculation comparisons of different algorithms

Bit legth	**Algorithms**		
	BTSM Algorithm [20]	Ternary scalar multiplication Algorithm [20]	Proposed Algorithm 5
101	$3005M$	$3407M$	$2801M$
122	$3630M$	$4113M$	$3384M$
142	$4225M$	$4786M$	$3939M$
162	$4820M$	$5458M$	$4493M$

version and multiplication. The efficiency can be derived from the following equation:

$$efficiency = 1 - \frac{34.88}{25.22+\alpha}$$

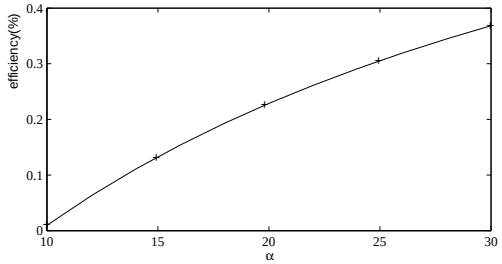

Figure 1: Efficiency of $3P + Q$ formula in Jacobian coordinate

Figure 1 shows the efficiency analysis of $3P + Q$ formula in Jacobian coordinate and affine coordinate. In Table 3, we choose the scalar symmetric ternary length as 101,122,142 and 162. When the scalar length of the ternary expansion is 162, it can be seen from Table 3 that our proposed Algorithm 5 improves the computational efficiency by 7% and 18% compared with the BTSM algorithm and the ternary scalar multiplication algorithm.

5 Conclusions

In this paper, the Co_Z point addition formula in different projective coordinates on elliptic curve over GF(3^m) is proposed. Compared with previously mixed point addition formulas, the computational time is reduced. Combined the new formulaes and optimized symmetric ternary scalar multiplication algorithm against SPA, the computational efficiency is improved. Compared with other algorithms, the efficiency of Algorithm 4 is increased by 16%, 26%, and 10%, respectively. We also proposed $3P+Q$ formula in Jacobian coordinate. The average running time of the improved balanced ternary scalar multiplication algorithm is about 7% faster than that of the BTSM algorithm.

Acknowledgments

The support of NSFC (National Natural Science Foundation of China, No.61872058), Shaanxi Natural Science Foundation (No.2017JQ6010) is gratefully acknowledged.

References

[1] G. Chen, G. Bai, and H. Chen, "A high-performance elliptic curve cryptographic processor for general curves over GF(p) based on a systolic arithmetic

unit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, pp. 412–416, May. 2007.

[2] J. J. Cheng, F. Y. Zheng, and J. Q. Lin, "High-performance implementation of Curve25519 on GPU," *Netinfo Security (in Chinese)*, no. 9, pp. 122–127, 2017.

[3] D. V. Chudnovsky and G. V. Chudnovsky, "Sequences of numbers generated by addition informal groups and new primality and factorization tests," *Advances in Applied Math*, vol. 7, no. 4, pp. 385–434, 1986.

[4] W. Y. Deng and X. H. Miao, "Application of balanced ternary in elliptic curve scalar multiplication," *Computer Engineering (in chinese)*, vol. 38, no. 5, pp. 152–154, 2012.

[5] Q. L. Du, "Identity-based dynamic group communication signcryption scheme," *Netinfo Security(in Chinese)*, no. 9, pp. 42–44, 2017.

[6] Q. F. Duanmu, X. Y. Zhang, Y. B. Wang, and K. Z. Zhang, "Fast arithmetic operations of elliptic curve over GF(3^n)," *Journal of PLA University of Science and Technology (Natural Science Eddition) (in Chinese)*, vol. 12, no. 01, pp. 1–6, 2011.

[7] R. R. Goundar, M. Joye, and A. Miyaji, "Co_Z addition formula and binary ladders on elliptic curves," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 65–79, 2010.

[8] H. Hisil, G. Carter, E. Dawson, "New formulae for efficient elliptic curve arithmetic," in *International Conference on Cryptology in India*, pp. 138–151, 2007.

[9] M. S. Hossain, Y. N. Kong, E. S, and N. V, "High-performance elliptic curve cryptography processor over NIST prime fields," *IET Computers and Digital Techniques*, vol. 11, no. 1, pp. 33–42, 2017.

[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 and T. Zhan, "Co_Z addition operation of hessian curve," in *Seventh International Conference on Computational Intelligence and Security*, pp. 915–919, Dec. 2011.

[12] Q. W. Li, Z. F. Wang, and X . C. Liu, "Fast point operation architecture for elliptic curve cryptography," in *IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 184–188, Nov. 2008.

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

[14] S. Liu, H. Yao, and X. A. Wang, "SPA resistant scalar multiplication based on addition and tripling indistinguishable on elliptic curve cryptosystem," in *The 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pp. 785 – 790, Nov. 2015.

[15] Meloni and Nicolas, "New point addition formulae for ECC applications," in *International Workshop on the Arithmetic of Finite Fieldss*, pp. 189–201, June. 2007.

[16] B. Y. Peng, Y. C. Hsu, Y. J. Chen, and D. C. Chueh, "Multi-core FPGA implementation of ECC with homogeneous Co-Z coordinate representation," in *The 15th International Conference on Cryptology and Network Security*, pp. 637–647, Nov. 2016.

[17] C. S. Sin, "Regular ternary algorithm for scalar multiplication on elliptic curves over finite fields of characteristic three," in *Technical Report Iacr Cryptology Eprint Archive*, July 2012. `https://eprint.iacr.org/2012/390.pdf`

[18] M. Wang and Z. Wu, "Algorithm of NAF scalar multiplication on ECC against SPA," *Journal on Communications (in Chinese)*, vol. 33, no. S1, pp. 228–232, 2012.

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

[20] N. Zhang and X. T. Fu, "Ternary method in elliptic curve scalar multiplication," in *The 5th International Conference on Intelligent Networking and Collaborative Systems*, pp. 490–494, Sep. 2013.

[21] Z. B. Zhou, S. B. Zhang, and E. T. Luo, "A group RFID tag ownership transfer protocol without trusted third party," *Netinfo Security (in Chinese)*, no. 6, pp. 18–27, 2018.

Biography

Shuang-Gen Liu, born in 1979, Ph.D, associate professor. A member of the China Computer Federation, and a member of the Chinese Association for Cryptologic Research. His main research interests focus on information security and cryptography.

Shi-Mei Lu, 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.

Rui-Wen Gong, born in 1998. An undergraduate student in information security in Xi'an University of posts and telecommunications. Her main research interest is cryptography.