Fast Point Multiplication on Elliptic Curves of Even Order

Rongquan Feng, Hongfeng Wu
School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China
fengrq@math.pku.edu.cn, wuhf@math.pku.edu.cn

Abstract

Every elliptic curve of even order over a finite field of characteristic > 3 is birationally equivalent to a curve in the Jacobi quartic form. This paper presents the new explicit formulae for group operations on a Jacobi quartic curve. The algorithm for doubling uses only $1M + 6S$, for the mixed-addition uses only $8M + 2S$ and the unified addition formula only $9M + 2S$ to be the best case. For elliptic curve of even order, these algorithm are more efficient than the other algorithms in the literature.

Keywords: Elliptic curve, Point Multiplication, Jacobi quartic, Side-Channel Analysis

1 Introduction

The main operation for elliptic curve cryptosystems is the point multiplication: $Q = kP$, where the multiplier k is generally a secret (or private) parameter. Efficient elliptic curve arithmetic is crucial for cryptosystems based on elliptic curves. Such cryptosystems often require computing kP for a given integer k and a curve point P. For example, if k is a secret key and P is another user's public key then kP is a Diffie-Hellman secret shared between the two users. Many methods to speed up this operation have been actively studied. See [2] and [6] for more details.

Every elliptic curve of even order over a finite field of characteristic > 3 is birationally equivalent to a curve in the Jacobi quartic form. Elliptic curve standards [9] recommend the use of elliptic curves with group order $\#E = h \cdot q$, where q is a prime and the cofactor h is ≤ 4. In this paper, we propose new scalar multiplication algorithm based on the new Jacobi quartic coordinates. The improved formulae and

algorithms for the basic arithmetic on the elliptic curves, such as point addition, point doubling and mixed additions are presented. These algorithms are more efficient than the best known algorithms in these elliptic curves.

This paper is organized as follows. In the next section some essential concepts to the Jacobi quartic is reviewed. Then, in Section 3, we show the improved point multiplications on Jacobi quartic curves. In section 4 the comparison of the new algorithms with other algorithms in literatures. Finally, we conclude in Section 5.

2 Background

In this section, the basic arithmetic of elliptic curves are briefly described. The reader is referred to [11] for more details. Throughout this paper, we assume that K represents a field of characteristic Char$K > 3$.

We start by recalling the explicit group law on elliptic curves. Let E denote an elliptic curve over K given by the Weierstrass form

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

and let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on the curve. The inverse of P_1 is $-P_1 = (x_1, -y_1)$. If $P_1 \neq -P_2$, then the sum $P_3 = P_1 + P_2 = (x_3, y_3)$ is defined as

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

where

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{when } x_1 \neq x_2, \\ \dfrac{3x_1^2 + a}{2y_1} & \text{when } x_1 = x_2. \end{cases}$$

The computational cost (timing) of scalar multiplications on elliptic curve operations depends on the cost of the arithmetic operations that have to be performed in the underlying field. In general, among these arithmetics, a field squaring, a field multiplication and a field inversion are more expensive than other field arithmetics, such as a field addition and a field subtraction. So we only take into account the cost of inversion, multiplication, and squaring in the field $\mathbb{F}_q$, which we denote by I, M and S, respectively. The symbol C stands for the cost of multiplication by a constant. The addition of two points in above formula requires $1I + 2M + 1S$ (doubling need$1I + 2M + 2S$). As inversion in a field of large characteristic is a relatively costly operation, projective

representations of points are preferred. A projective point $P = (X_P, Y_P, Z_P)$ on the curve satisfies the projective Weierstrass equation

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3$$

corresponds to the affine point $P = (X_P/Z_P, Y_P/Z_P)$ when $Z_P \neq 0$. The inverse of P is $-P = (X_P, -Y_P, Z_P)$.

2.1 Jacobi Models

Remarkably, an elliptic curves with even order can be expressed by the extended Jacobi form ([1])

$$J : Y^2 = \epsilon X^4 - 2\delta X^2Z^2 + Z^4$$

Let $(\theta, 0) \in E(K)$ be a point of order 2. Then, the above Weierstrass elliptic curve E is birationnally equivalent to the extended Jacobi quartic curve J with $\epsilon = -(3\theta^2+4a)/16$ and $\delta = 3\theta/4$, under the transformations

$$\begin{cases} (\theta, 0) & \longmapsto & (0, -1, 1), \\ (x, y) & \longmapsto & (2(x-\theta), (2x+\theta)(x-\theta)^2) - y^2, y), \\ \mathcal{O} & \longmapsto & (0, 1, 1); \end{cases}$$

and

$$\begin{cases} (0, 1, 1) & \longmapsto & \mathcal{O}, \\ (0, -1, 1) & \longmapsto & (\theta, 0), \\ (X, Y, Z) & \longmapsto & (2\dfrac{(Y+Z^2)}{X^2} - \dfrac{\theta}{2}, Z\dfrac{4(Y+Z^2) - 3\theta X^2}{X^3}); \end{cases}$$

Note that two triplets (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) represent the same point if and only if there exists $t \in K \setminus \{0\}$ such that $X_1 = tX_2$, $Y_1 = t^2Y_2$ and $Z_1 = tZ_2$.

Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points on the Jacobi curve J. The sum $P_3 = P_1 + P_2$ is defined as (X_3, Y_3, Z_3) with

$$\begin{aligned} X_3 &= X_1Z_1Y_2 + X_2Z_2Y_1 \\ Y_3 &= [(Z_1Z_2)^2 + \epsilon(X_1X_2)^2](Y_1Y_2 - 2\delta X_1X_2Z_1Z_2) \\ &\quad + 2\epsilon X_1X_2Z_1Z_2(X_1^2Z_2^2 + Z_1^2X_2^2) \\ Z_3 &= (Z_1Z_2)^2 - \epsilon(X_1X_2)^2 \end{aligned}$$

The negative of a point (X, Y, Z) on the extended Jacobi quartic is given by $(-X, Y, Z)$. It is worth noting the same formula applies for adding distinct points or for doubling a point. The cost of the algorithm in [1] for the unified addition formula is $10M+3S+3C$.

Finally, when the curve contains a copy of $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$, the Jacobi curve equation J can be rescaled, in most cases, to the value $\epsilon = 1$. In [1], Billet and Joye pointed out that when constant δ (resp. ϵ) is small, the cost of a multiplication by δ (resp. ϵ) can be neglected. In particular, most elliptic curves over the prime field $K = \mathbb{F}_q$, with three points of order 2, can be rescaled to the case $\epsilon = 1$. So we can have the simple Jaconi form

$$SJ : Y^2 = X^4 - 2\rho X^2 Z^2 + Z^4.$$

3 Fast Point Multiplications

3.1 The addition Formulae in JQ^{N1}

Now we modify the Jacobi quartic coordinates in order to obtain the fastest possible addition and doubling. We represent the Jacobi quartic coordinates as a quadruple (X, Y, Z, XZ). We call this the modified Jacobi quartic coordinate, and denote it by JQ^{N1}. The addition formulae in the modified Jacobi quartic coordinates are the following. Let $P = (X_1, Y_1, Z_1, X_1Z_1)$, $Q = (X_2, Y_2, Z_2, X_2Z_2)$ and $P + Q = R = (X_3, Y_3, Z_3, X_3Z_3)$

• **Addition formula in JQ^{N1} ($P \neq \pm Q$)**

$$X_3 = U_1 - U_2 - V, Y_3 = (S_2^2 + \epsilon S_1^2)(U_2 - 2\delta V) + 2\epsilon V(H^2 - 2V), Z_3 = (S_2^2 - \epsilon S_1^2), X_3Z_3,$$

where $U_1 = (X_1Z_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V = X_1Z_1X_2Z_2$, $S_1 = X_1X_2$, $S_2 = Z_1Z_2$, $H = (X_1 + X_2)(Z_1 + Z_2) - X_1Z_1 - X_2Z_2$;

• **Doubling formula in JQ^{N1} ($P = Q$)**

$$X_3 = U_1 - U_2 - S_1, Y_3 = (T + \epsilon V_1^2)(U_2 - 2\delta S_1) + 4\epsilon S_1^2, Z_3 = T - \epsilon V_1^2, X_3Z_3,$$

where $U_1 = (X_1Z_1 + Y_1)^2$, $U_2 = Y_1^2$, $V_1 = X_1^2$, $S_1 = (X_1Z_1)^2$, $T = U_2 - \epsilon V_1^2 + 2\delta S_1$;

• **Mixed-addition formula in JQ^{N1} ($P \neq \pm Q, Z_1 = 1$)**

$$X_3 = U_1 - U_2 - V_1, Y_3 = (H + \epsilon V_2^2)(U_2 - 2\delta V_1) + 2\epsilon V_1(S^2 - 2V_1), Z_3 = H - \epsilon V_2^2, X_3Z_3,$$

where $U_1 = (X_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V_1 = X_1X_2Z_2$, $V_2 = X_1X_2$, $S = X_1Z_2 + X_2$, $H = Z_2^2$;

The computation costs are $t(\text{Addition}) = 9M + 3S + 3C$, $t(\text{Doubling}) = 2M + 6S + 3C$, $t(\text{Mixed-addition}) = 8M + 3S + 3C$.

3.2 The addition Formulae in JQ^{N2}

In the above algorithms, if we represent the Jacobi quartic coordinates as a quintuple (X, Y, Z, X^2, XZ), and denote it by JQ^{N2}. Let $P = (X_1, Y_1, Z_1, X_1^2, X_1Z_1)$, $Q = (X_2, Y_2, Z_2, X_2^2, X_2Z_2)$ and $P + Q = R = (X_3, Y_3, Z_3, X_3^2, X_3Z_3)$, Then we have:

• **Addition formula in JQ^{N2} $(P \neq \pm Q)$**

$$X_3 = U_1 - U_2 - V, Y_3 = (S_2 + \epsilon S_1)(U_2 - 2\delta V) + 2\epsilon VH, Z_3 = (S_2 - \epsilon S_1), X_3^2, X_3Z_3,$$

where $U_1 = (X_1Z_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V = X_1Z_1X_2Z_2$, $T_1 = Z_1^2$, $T_2 = Z_2^2$, $S_1 = X_1^2X_2^2$, $S_2 = T_1T_2$, $H = (X_1^2 + T_1)(X_2^2 + T_2) - S_1 - S_2$;

• **Doubling formula in JQ^{N2} $(P = Q)$**

$$X_3 = U_2 - U_2 - S_1, Y_3 = (T + \epsilon V_1)(U_2 - 2\delta S_1) + 4\epsilon S_1^2, Z_3 = T - \epsilon V_1, X_3^2, X_3Z_3,$$

where $U_1 = (X_1Z_1 + Y_1)^2$, $U_1 = Y_1^2$, $V_1 = (X_1^2)^2$, $S_1 = (X_1Z_1)^2$, $T = U_2 - \epsilon V_1 + 2\delta S_1$;

If we can take α as a square-root of the corresponding square $\epsilon - \delta^2$, then we have the following algorithm:

$$X_3 = U, Y_3 = S_1 - \alpha V, Z_3 = T_2 + 2\delta T_1 - 2\epsilon X_1^4, X_3^2 = V, X_3Z_3,$$

where $T_1 = (X_1Z_1)^2$, $T_2 = Y_1^2$, $S_1 = (T_2 + 2\alpha T_1)^2$, $U = (X_1Z_1 + Y_1)^2 - T_1 - T_2$; $V = U^2$;

• **Mixed-addition formula in JQ^{N2} $(P \neq \pm Q, Z_1 = 1)$**

$$X_3 = U_1 - U_2 - V_1, Y_3 = (H + \epsilon V_2)(U_2 - 2\delta V_1) + 2\epsilon V_1(S_2 + X_2^2), Z_3 = H - \epsilon V_2, X_3^2, X_3Z_3,$$

where $U_1 = (X_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V_1 = X_1X_2Z_2$, $V_2 = X_1^2X_2^2$, $S_1 = Z_2^2$, $S_2 = X_1^2S_1$;

The computation costs are $t(\text{Addition}) = 9M + 3S + 3C$, $t(\text{Doubling}) = 2M + 6S + 3C$ or $t(\text{Doubling}) = 1M + 6S + 4C$, $t(\text{Mixed-addition}) = 8M + 2S + 3C$. Noting that, for the sake of saving memory we don't need to store X in the intermediate stage of computing. That is we can use substitution of quadruple (Y, Z, X^2, XZ) for quintuple (X, Y, Z, X^2, XZ).

3.3 The addition Formulae in JQ^{N3}

Now we represent the Jacobi quartic coordinates as a sextuplet (X, Y, Z, X^2, Z^2, XZ), and denote it by JQ^{N3}. Let $P = (X_1, Y_1, Z_1, X_1^2, Z_1^2, X_1Z_1)$, $Q = (X_2, Y_2, Z_2, X_2^2, Z_2^2, X_2Z_2)$ and $P + Q = R = (X_3, Y_3, Z_3, X_3^2, Z_3^2, X_3Z_3)$, Then we have:

• **Addition formula in JQ^{N3} $(P \neq \pm Q)$**

$X_3 = U_1 - U_2 - V, Y_3 = (S_2 + \epsilon S_1)(U_2 - 2\delta V) + 2\epsilon VH, Z_3 = (S_2 - \epsilon S_1), X_3^2, Z_3^2, X_3Z_3,$

where $U_1 = (X_1Z_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V = X_1Z_1X_2Z_2$, $S_1 = X_1^2X_2^2$, $S_2 = Z_1^2Z_2^2$, $H = (X_1^2 + Z_1^2)(X_2^2 + Z_2^2) - S_1 - S_2$;

• **Doubling formula in JQ^{N3} $(P = Q)$**

$$X_3 = U_1 - U_2 - S_1, Y_3 = (T + \epsilon V_1)(U_2 - 2\delta S_1) + 4\epsilon S_1^2, Z_3 = T - \epsilon V_1, X_3^2, Z_3^2, X_3Z_3,$$

where $U_1 = (X_1Z_1 + Y_1)^2$, $U_2 = Y_1^2$, $V_1 = (X_1^2)^2$, $S_1 = (X_1Z_1)^2$, $T = U_2 - \epsilon V_1 + 2\delta S_1$;

• **Mixed-addition formula in JQ^{N3} $(P \neq \pm Q, Z_1 = 1)$**

$X_3 = U_1 - U_2 - V, Y_3 = (Z_2^2 + \epsilon S_1)(U_2 - 2\delta V) + 2\epsilon V(S_2 + X_2^2), Z_3 = Z_2^2 - \epsilon S_1, X_3^2, Z_3^2, X_3Z_3,$

where $U_1 = (X_1 + Y_1)(X_2Z_2 + Y_2)$, $U_2 = Y_1Y_2$, $V = X_1X_2Z_2$, $S_1 = X_1^2X_2^2$, $S_2 = X_1^2Z_2^2$;

The computation costs are $t(\text{Addition}) = 9M + 2S + 3C$, $t(\text{Doubling}) = 2M + 7S + 3C$ and $t(\text{Mixed-addition}) = 8M + 2S + 3C$. Noting that, For the sake of saving memory we don't need to store X and Z in the intermediate stage of computing. That is we can use substitution of quadruple (Y, X^2, Z^2, XZ) for sextuplet (X, Y, Z, X^2, Z^2, XZ). The costs of these algorithms presented in **Table 1**.

4 Comparison

The reader is referred to [6] for a survey of the algorithms for scalar multiplications. We can make the standard assumption that the input point P has $Z = 1$, and all additions of P can thus be computed as mixed additions. A scalar multiplication kP consists of a sequence of doubling and additions, where the doubling and mixed-addition are the main operations. For example, for an average t-bit scalar multiplication kP, the NAF algorithm uses approximately t doubling and approximately $\frac{t}{3}$ mixed-additions. The

Table 1: Costs of Algorithm

Coordinates	JQ^{N1}	JQ^{N2}	JQ^{N3}
Addition	$9M + 3S + 3C$	$9M + 3S + 3C$	$9M + 2S + 3C$
$\epsilon = 1$	$9M + 3S + C$	$9M + 3S + C$	$9M + 2S + C$
$1C = 0.5M$	$11.9M$	11.9	$11.1M$
Neglect C	$11.4M$	$11.4M$	$10.6M$
Doubling	$2M + 6S + 3C$	$2M + 6S + 3C$ or $1M + 6S + 4C$	$2M + 7S + 3C$
$\epsilon = 1$	$2M + 6S + C$	$2M + 6S + C$ or $1M + 6S + 3C$	$2M + 7S + C$
$1C = 0.5M$	$7.3M$	$7.3M$ or $7.3M$	$8.1M$
Neglect C	$6.8M$	$6.8M$ or $5.8M$	$7.6M$
Mixed-addition	$8M + 3S + 3C$	$8M + 2S + 3C$	$8M + 2S + 3C$
$\epsilon = 1$	$8M + 3S + C$	$8M + 2S + C$	$8M + 2S + C$
$1C = 0.5M$	$10.9M$	$10.1M$	$10.1M$
Neglect C	$10.4M$	$9.6M$	$9.6M$

signed width-4 sliding windows algorithm involves, on average, approximately t doublings, $\frac{17t}{100}$ directly $2T + P$ additions, $\frac{3t}{100}$ mixed additions. Thus, for these algorithms of scalar multiplications, we use the JQ^{N2} coordinates.

4.1 Comparison with other algorithms

In this subsection we compare the costs of the algorithms in Section 3 to the costs of previous algorithms for elliptic-curve doubling, elliptic-curve mixed-addition and unified addition. In general, no special value for the parameters of the curve is considered when this has a limited impact anyway on the complexity of the operations. In this work, some savings can be made if $\epsilon = 1$ or the parameters ϵ, δ is specially chosen such that we can neglect of multiplying by these parameters, like a small ϵ, δ or more generally ϵ, δ with low Hamming weight expansion. The different costs for doubling and mixed-addition formula are summarized in Table 2.

Note that we have a very efficient doubling and mixed-addition algorithms, it is more efficient be applied in the NAF scalar multiplication, JSF(joint sparse form) algorithm and the other windows algorithm.

Table 2: Comparison with other algorithms

System	This work	[8]	[7]
Equation	$Y^2 = \epsilon X^4 - 2\delta X^2 Z^2 + Z^4$	$y^2 = x^3 + 3u(x+1)^2$	$y^2 = x^3 + ax + b$
Coordinates	JQ^{N2}	Jacobian J^n	Jacobian J
Doubling	$1M + 6S + 4C$	$4M + 4S + 2C$	$4M + 5S + 1C$
special $\epsilon = 1$	$1M + 6S + 3C$	–	–
$1C = 0.5M$	$7.3M$	$8.2M$	$8.5M$
neglect C	$1M + 6S$	$4M + 5S$	$4M + 4S$
Mixed Addition	$8M + 2S + 3C$	$8M + 3S + 1C$	$8M + 3S$
special $\epsilon = 1$	$8M + 3S + 1C$	–	–
$1C = 0.5M$	$10.9M$	$10.9M$	10.4
Neglect C	$8M + 2S$	$8M + 3S$	$8M + 3S$

Table 3: Comparison for 256-bit scalar multiplication algorithms

System	NAF Algorithm($C = M$)	$C = 0.5M$	Neglect C
[7]	$3200M$	$3063.5M$	$2944M$
[8]	$3584M$	$3029.3M$	$2935.5M$
This work	$2969.6M$	$2798.9M$	$2304M$

Table 4: Unified Addition Comparison

System	Costs	Cofactor
Weierstrass form[3]	$17M + 1C$ (general case)	–
with unified formula	$16M + 1C(a_4 = -1)$	–
Hessian form	$12M$	$h \propto 3$
Extended Jacobi form	$13M + 3C$ (general case)	$h \propto 2$
In [1]	$13M + 1C$ $(\epsilon = 1)$	$h \propto 4$
$\cap$ of 2 quadrics [**?**]	$16M + 1C$	$h \propto 4$
Extended Jacobi form	$11M + 2C$ (general case)	$h \propto 2$
This work JQ^{N3}	$11M + 1C$ $(\epsilon = 1)$	$h \propto 4$

4.2 Side-Channel Attacks

In traditional cryptographic model, the security of the system relies solely on the secrecy of the key and the mathematical properties of the cryptographic algorithm used. Side-channel analysis (SCA) or information leakage analysis (ILA), refers to a new and emerging type of cryptanalysis that uses leaked side-channel information from a cryptographic device to determine the secret key. In general scalar multiplications of elliptic curves, many secret bits of the integers k_i are leaked, through the pattern of doubling and mixed addition and non-mixed addition, to side-channel attacks such as simple power analysis. For example, from the distinction between the two operations, doubling or for adding two, a simple power analysis might produce different power traces, revealing the value of k in the point multiplication algorithm. Using the unified addition formula is one of the basically approaches to circumvent the leakage. Note that the addition algorithm in Section 3 is the unified addition formula for adding or doubling points. The different costs for unified addition formula are summarized in Table 4. The unified algorithm in atomic blocks prevent SCA is given in Table 5 on the JQ^{N2} coordinates system.

5 Conclusions

Every elliptic curve of even order over a finite field of characteristic > 3 is birationally equivalent to a curve in the Jacobi quartic form. This paper describes the fast formulas for group operations on a Jacobi quartic curve in three new coordinates system. The algorithm for doubling uses only $1M + 6S$, for the mixed-addition uses only $8M + 2S$ and the unified addition formula only $9M + 2S$ to the best case. These formulae will

Table 5: **Unified algorithm in atomic blocks** JQ^{N2}

Input:$P = (Y_1, Z_1, X_1^2, X_1Z_1)$, $Q = (Y_2, Z_2, X_2^2, X_2Z_2)$

Output: $P + Q = (X_3, Y_3, Z_3, X_3^2, X_3Z_3)$

Init: $R_1 = Y_1, R_2 = Z_1, R_3 = X_1^2, R_4 = X_1Z_1, R_5 = Y_2, R_6 = Z_2, R_7 = X_2^2, R_8 = X_2Z_2$

$R_9 = R_4R_8$	$(= X_1Z_1X_2Z_2)$
$R_4 = R_1 + R_4$	$(= X_1Z_1 + Y_1)$
$R_8 = R_5 + R_8$	$(= X_2Z_2 + Y_2)$
$R_1 = R_1R_5$	$(= Y_1Y_2)$
$R_5 = R_4R_8$	$(= (X_1Z_1 + Y_1)(X_2Z_2 + Y_2))$
$R_5 = R_5 - R_1$	$(= (X_1Z_1 + Y_1)(X_2Z_2 + Y_2) - Y_1Y_2)$
$R_5 = R_5 - R_9$	$(= X_3)$
$R_4 = 2\delta R_9$	$(= 2\delta X_1Z_1X_2Z_2)$
$R_1 = R_1 - R_4$	$(= Y_1Y_2 - 2\delta X_1Z_1X_2Z_2)$
$R_2 = R_2^2$	$(= Z_1^2)$
$R_6 = R_6^2$	$(= Z_2^2)$
$R_4 = R_3R_7$	$(= X_1^2X_2^2)$
$R_8 = R_2R_6$	$(= Z_1^2Z_2^2)$
$R_3 = R_2 + R_3$	$(= X_1^2 + Z_1^2)$
$R_7 = R_6 + R_7$	$(= X_2^2 + Z_2^2)$
$R_3 = R_3R_7$	$(= (X_1^2 + Z_1^2)(X_2^2 + Z_2^2))$
$R_6 = R_4 + R_8$	$(= X_1^2X_2^2 + Z_1^2Z_2^2)$
$R_3 = R_3 - R_6$	$(= X_1^2Z_2^2 + X_2^2Z_1^2)$
$R_1 = R_1R_6$	$(= (Y_1Y_2 - 2\delta X_1Z_1X_2Z_2)(X_1^2X_2^2 + Z_1^2Z_2^2))$
$R_6 = 2R_3R_9$	$(= 2X_1X_2Z_1Z_2(X_1^2Z_2^2 + X_2^2Z_1^2))$
$R_1 = R_1 + R_6$	$(= Y_3)$
$R_6 = R_8 - R_4$	$(= Z_3)$
$R_7 = R_5^2$	$(= X_3^2)$
$R_8 = R_5R_6$	$(= X_3Z_3)$

Table 6: **Unified algorithm in atomic blocks** JQ^{N3}

Input:$P = (Y_1, X_1^2, Z_1^2, X_1Z_1)$, $Q = (Y_2, X_2^2, Z_2^2, X_2Z_2)$

Output: $P + Q = (X_3, Y_3, Z_3, X_3^2, X_3Z_3)$

Init: $R_1 = Y_1, R_2 = X_1^2, R_3 = Z_1^2, R_4 = X_1Z_1, R_5 = Y_2, R_6 = X_2^2, R_7 = Z_2^2, R_8 = X_2Z_2$

$R_9 = R_4R_8$	$(= X_1Z_1X_2Z_2)$
$R_4 = R_1 + R_4$	$(= X_1Z_1 + Y_1)$
$R_8 = R_5 + R_8$	$(= X_2Z_2 + Y_2)$
$R_1 = R_1R_5$	$(= Y_1Y_2)$
$R_5 = R_4R_8$	$(= (X_1Z_1 + Y_1)(X_2Z_2 + Y_2))$
$R_5 = R_5 - R_1$	$(= (X_1Z_1 + Y_1)(X_2Z_2 + Y_2) - Y_1Y_2)$
$R_5 = R_5 - R_9$	$(= X_3)$
$R_4 = 2\delta R_9$	$(= 2\delta X_1Z_1X_2Z_2)$
$R_1 = R_1 - R_4$	$(= Y_1Y_2 - 2\delta X_1Z_1X_2Z_2)$
$R_4 = R_2R_6$	$(= X_1^2X_2^2)$
$R_8 = R_3R_7$	$(= Z_1^2Z_2^2)$
$R_2 = R_2 + R_3$	$(= X_1^2 + Z_1^2)$
$R_6 = R_6 + R_7$	$(= X_2^2 + Z_2^2)$
$R_2 = R_2R_6$	$(= (X_1^2 + Z_1^2)(X_2^2 + Z_2^2))$
$R_6 = R_4 + R_8$	$(= X_1^2X_2^2 + Z_1^2Z_2^2)$
$R_2 = R_2 - R_6$	$(= X_1^2Z_2^2 + X_2^2Z_1^2)$
$R_1 = R_1R_6$	$(= (Y_1Y_2 - 2\delta X_1Z_1X_2Z_2)(X_1^2X_2^2 + Z_1^2Z_2^2))$
$R_6 = 2R_2R_9$	$(= 2X_1X_2Z_1Z_2(X_1^2Z_2^2 + X_2^2Z_1^2))$
$R_1 = R_1 + R_6$	$(= Y_3)$
$R_6 = R_8 - R_4$	$(= Z_3)$
$R_2 = R_6^2$	$(= Z_3^2)$
$R_7 = R_5^2$	$(= X_3^2)$
$R_8 = R_5R_6$	$(= X_3Z_3)$

allow more efficient point multiplications on elliptic curves whose order is even. For these curves, the proposed algorithms are more efficient than the other algorithms in the literature.

References

[1] Olivier Billet and Marc Joye, The Jacobi Model of an Elliptic Curve and Side-Channel Analysis, AAECC 2003, LNCS 2643, 34C42, Spriger-Verlag, 2003.

[2] I.F. Blake, G. Seroussi and N.P. Smart, Advances in Elliptic Curve Cryptography. London Math. Soc. Lecture Note Ser. 317, Cambridge Univ. Press, 2005.

[3] É. Brier and M. Joye. Weierstrass elliptic curves and side-channel attacks, PKC 2002, LNCS 2274, 335-345, Springer-Verlag, 2002.

[4] M. Ciet, M. Joye, K. Lauter, and P.L. Montgomery, Trading inversions for multiplications in elliptic curve cryptography, Design, Codes and Cryptography 39(2), 189-206, 2006.

[5] H. Cohen, Atsuko Miyaji, Takatoshi Ono, Efficient elliptic curve exponentiation using mixed coordinates, ASIACRYPT'98, LNCS 1514, 51C65, Spriger-Verlag, 1998.

[6] H. Cohen, Gerhard Frey (editors), Handbook of elliptic and hyperelliptic curve cryptography, CRC Press, 2006.

[7] V.S. Dimitrov, L. Imbert, and P.K. Mishra, Efficient and secure elliptic curve point multiplication using double-base chains. ASIACRYPT 2005, LNCS 3788, 59C78, Springer-Verlag, 2005.

[8] C. Doche,T. Icart and D.R. Kohel, Efficient Scalar Miltiplication by Isogeny Decompositions, PKC 2006, LNCS 3958, 191-206, Spriger-Verlag, 2006.

[9] IEEE 1363. Standard Specifications for Public Key Cryptography. IEEE, 2000.

[10] M. Joye and J. Quisquater, Hessian Elliptic Curves and Side-Channel Attacks, CHES 2001, Springer-Verlag, LNCS 2162, 402C410, 2001.

[11] J.H. Silverman, The Arithmetic of Elliptic Curves, GTM 106, Springer-Verlag, Berlin, 1986.