Fast Multi-Scalar Multiplication Methods on Elliptic Curves with Precomputation Strategy using Montgomery Trick

Katsuyuki Okeya Hitachi Ltd.

Kouichi Sakurai Kyushu Univ.

Abstract

Motivation	The use of multi-scalar multiplication in the verification of ECDSA	[ANSI]
	The transformation from scalar multiplication to multi-scalar multiplication	[GLV01]
Problem	Speeding up the multi-scalar multiplication	
Result	Efficient Precomputation provides speedup for multi-scalar multiplication	

3 times faster

HSDL
Harmonize the Society with Digital Life

Contents

What is Multi-Scalar Multiplication?

Scalar multiplication

k an integer
P an elliptic point

Scalar multiplication

$$kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$$

Multi-scalar multiplication

k, l integers
P, Q elliptic points

Multi-scalar multiplication

$$kP + lQ = \underbrace{P + P + \cdots + P}_{k \text{ times}} + \underbrace{Q + Q + \cdots + Q}_{l \text{ times}}$$

Two Computation Methods for Multi-Scalar Multiplication

Separate Method

Compute separately two scalar multiplications

Comb method **[LL94]**

Window method **[Knu81, CMO98]**

k, P —Scalar multiplication→ kP

l, Q —Scalar multiplication→ lQ

kP, lQ —addition→ $kP + lQ$

Simultaneous Method

Compute simultaneously two scalar multiplications

Shamir's trick **[Elg85, HHM00]**

Improvement **[Aki01, Moe01]**

k, P / l, Q —Multi-scalar multiplication→ $kP + lQ$

$P, Q, P + Q$ —addition→ $sP + tQ$

Computation Process

Input → **Precomputa-tion stage** → **Evaluation stage** → Output

Precomputation stage: Preparation of a table

Evaluation stage: Actual computation

k, P
l, Q →

O	P	$2P$	$3P$
Q	$P+Q$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

Precomputation table

→ $\boxed{sP+tQ}$ + addition → $kP+lQ$

$\boxed{2P+3Q}$

Table

Target of Speedup

	Precomputation Stage	Evaluation Stage
Separate Method	Fast [CMO98]	Slow [CC87, MO90, LL94, CMO98, ...]
Simultaneous Method	Slow	Fast [Aki01, Moe01, Sol01]

Many researches exist

Not so much studies as evaluation stage

Controversial !

What are Obstacles to Speed up the Precomputation Stage?

Obstacles

Inversions are required (1 per point)

Many precomputation points

Some points are not used in evaluation stage

Multi-scalar multiplication only

What are Obstacles to Speed up the Precomputation Stage?

Obstacles	Reason
Inversions are required (1 per point)	Points are computed in affine coordinates
Many precomputation points	Table should be saved points in affine coordinates for speeding up evaluation stage
Some points are not used in evaluation stage	The operation in affine coordinates requires inversion

Multi-scalar multiplication only

HSDL

Harmonize the Society with Digital Life

What are Obstacles to Speed up the Precomputation Stage?

Obstacles	Reason
Inversions are required (1 per point)	
Many precomputation points	**2 dimensions**
Some points are not used in evaluation stage	
Multi-scalar multiplication only	

$$uP + vQ \qquad u, v = 0, 1, \cdots$$

O	P	$2P$	$3P$
Q	$P+Q$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

What are Obstacles to Speed up the Precomputation Stage?

Obstacles	Reason
Inversions are required (1 per point)	Precomputation stage Points to compute: 64 points
Many precomputation points	Evaluation stage Points to use: 54 points
Some points are not used in evaluation stage	160 bits, window width 3
Multi-scalar multiplication only	

Contents

- Multi-Scalar Multiplication
- Target of Speedup
- Proposed Method
- Comparison

Simple Improvements

$$P = (x, y) \rightarrow -P = (x, -y)$$

Simultaneous inversion

$\pm Q$ has same x-coordinates

$$P \pm Q$$

Omit computation

Negate the y-coordinate

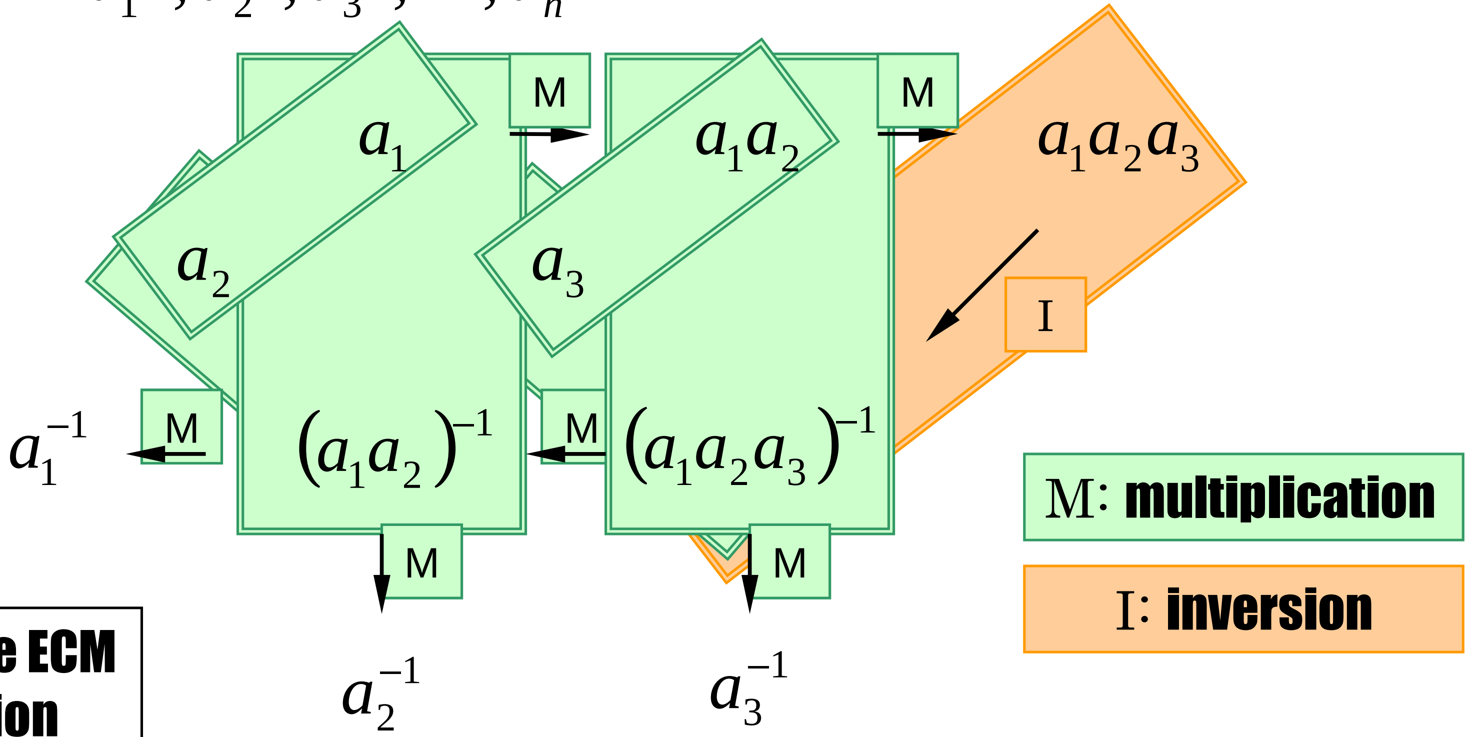
$$P + Q \rightarrow -P - Q$$

Montgomery Trick of Simultaneous Inversions [Coh93]

Input $a_1, a_2, a_3, \cdots, a_n$

Cost $3(n-1)M + I$

Output $a_1^{-1}, a_2^{-1}, a_3^{-1}, \cdots, a_n^{-1}$



[Coh93]

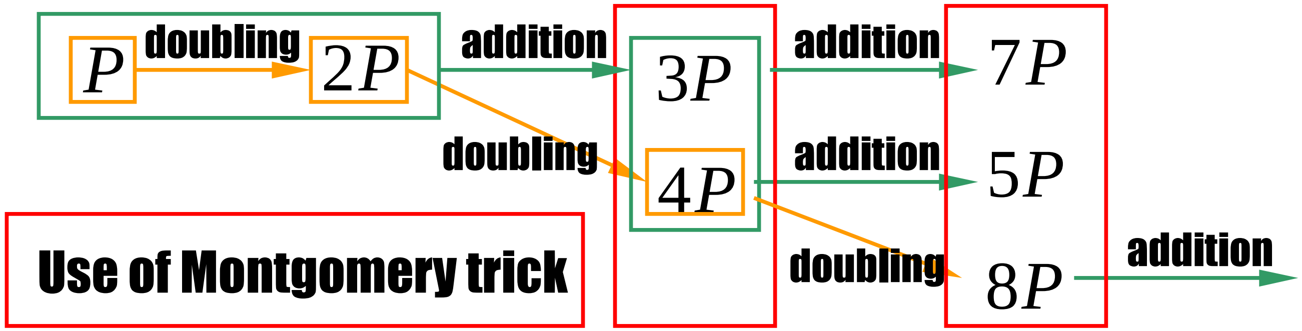
It speeds up the ECM of factorization

Use of Montgomery Trick (Scalar Multiplication)

[CM098]

Montgomery trick reduces from plural inversions to 1 inversion

Preparation of precomputation table

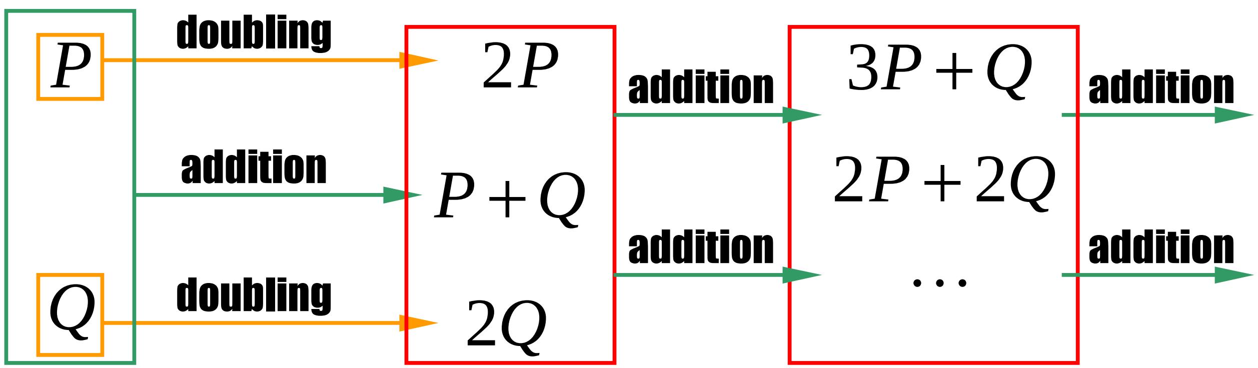


Compute inversion using Montgomery trick

Use of Montgomery Trick (Multi-Scalar Multiplication)

Montgomery trick reduces from plural inversions to 1 inversion

Preparation of precomputation table



Compute inversion using Montgomery trick

Complicated because of 2 dimensions

Preparation of Precomputation Table

Precomputation Table

O	P	$2P$	$3P$
Q	$P+Q$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

- Step 0: O, P, Q
- Step 1: $2P$, $P+Q$, $2Q$
- Step 2: $3P$, $2P+Q$, $3P+Q$, $P+2Q$, $2P+2Q$, $3Q$, $P+3Q$
- Step 3: $3P+2Q$, $2P+3Q$, $3P+3Q$

Each step uses Montgomery trick of simultaneous inversion

Some Points Do Not Need to be Computed

Precomputation Table

O	P	$2P$	$3P$
Q	$\cancel{P+Q}$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

Step 0

Step 1

Step 2

Step 3

They cannot be computed in Step 2

Consider how the points are computed!

HSDL
Harmonize the Society with Digital Life

Proposed Method

Precomputation Table

O	P	$2P$	$3P$
Q	$P+Q$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

Step 0

Step 1

Step 2

Step 3

uP, vQ **are first, the middles are last**

Some Points Do Not Need to be Computed

Precomputation Table

O	P	$2P$	$3P$
Q	$\cancel{P+Q}$	$2P+Q$	$3P+Q$
$2Q$	$P+2Q$	$2P+2Q$	$3P+2Q$
$3Q$	$P+3Q$	$2P+3Q$	$3P+3Q$

Step 0

Step 1

Step 2

Step 3

It does not affect the computation for the other points

Comparison

160 bits	Precomputation stage	Evaluation stage	Total
Separate Method [CM098]	336.8M	2809.8M	3146.6M
Simultaneous Method			
Conventional Method [HHM00] [Moe01]	1011.6M	1655.5M	2667.1M
Proposed method	279.2M	1655.5M	1934.7M

Conclusion

Problem	Speeding up the Multi-scalar multiplication
Points	Montgomery trick of simultaneous inversions
	Simplification of precomputation procedures
Result	Efficient Precomputation provides speedup for multi-scalar multiplication
	3 times faster
Application	Speeding up the verification of ECDSA
	Speeding up the scalar multiplication using multi-scalar multiplication