社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

楕円曲線スカラー倍算における効率的なアトミックブロック

チュンサティアンサップ　チッチャノック†

† 東京大学大学院 情報理工学系研究科　〒 113-0033 東京都文京区本郷 7-3-1
E-mail: †chitchanok@is.s.u-tokyo.ac.jp

あらまし　本論文では、楕円曲線スカラー倍算における安全性を持つ高速アルゴリズムについて論ずる。Chevallier–Mames, Ciet, Joye らは、単純サイドチャネル解析を防止するため、サイドチャネルアトミシティを提案した。しかし、彼らの原案のアトミックブロックをはじめ、工夫されたものすべてにおいて、一つのパターンしか使用されていない。これに対し、本論文では二つのパターンの使用する新しい方法を導入し、演算の調整およびダミー演算の減少をしたパターンを示す。二つのパターンを利用した結果、ブロックごとのコストを 1.36% 減らすことに成功した。
キーワード　サイドチャネル解析, アトミックブロック, 楕円曲線暗号

Efficient Atomic Block for Faster Elliptic Curve Scalar Multiplication

Chitchanok CHUENGSATIANSUP†

† Graduate School of Information Science and Technology
The University of Tokyo　7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033
E-mail: †chitchanok@is.s.u-tokyo.ac.jp

Abstract　We have developed new atomic patterns for faster elliptic curve scalar multiplication by rearranging field arithmetic and removing some unnecessary dummy operations. Atomicity concept is an alternative method to protect the algorithm from simple side-channel analysis. In contrast to previous results that used only one atomic pattern, we present the advantages of using more than one pattern and also introduce a usage of two atomic patterns for left-to-right binary scalar multiplication. As a result, cost per bit of when computing scalar multiplication is reduced by 1.36% if our new patterns are used.
Key words　Side-Channel Analysis, Atomic Block, Elliptic Curve Cryptography

1. Introduction

Side-channel attack, threat in which adversaries observe information leakage via physical devices, has become serious concerns and increasingly gained much attention in cryptography. Countermeasures against this kind of attack are extensively being studied, and many algorithms have already been proposed.

Chevallier-Mames, Ciet, and Joye [1] introduced a concept of *atomicity* to prevent simple side-channel attack. They divided each process into small blocks of instructions called *side-channel atomic block*. Each block was made identical so that it appeared to be indistinguishable by side-channel analysis.

Side-channel atomic block was improved once by Longa and Miri [2] to lower the cost and was made suitable for binary-ternary base scalar multiplication involving point tripling. Giraud and Verneuil [3] further improved atomic block for right-to-left binary scalar multiplication used in embedded devices.

We studied the atomic patterns proposed in [1]〜[3] and found that some dummy operations could be removed with neither losing its nice property of indistinguishability nor weakening its security. In this paper, we introduce three new atomic patterns for three different contexts.

Our first contribution is new atomic patterns for binary left-to-right scalar multiplication. These patterns are designed for general point doubling and mixed point addition. Unlike in all of the previous works, we no longer use just only one pattern. In other words, we introduce an idea of using two patterns together. These two patterns are executed alternately so that they reveal no differences of which point operation is being computed. With our new patterns, 1.36% of a cost per bit of when computing scalar multiplication can

be decreased.

Our second contribution is new atomic patterns for binary right-to-left mixed coordinates scalar multiplication in embedded devices. That is, scalar multiplication computing point doubling in *modified* Jacobian projective coordinates and general point addition in Jacobian projective coordinates was considered. We also investigated the possibility to profit from trading field multiplication for field squaring in the resource constrained environment. Again, we introduce a usage of two patterns for this situation which can reduce $0.22 - 0.65\%$ of a cost per bit of when computing scalar multiplication.

Our third contribution is new atomic patterns applicable to binary-ternary base scalar multiplication involving point tripling. We examined atomic structures proposed in [2] and found that they could be more compact. As a result of rearranging those field operations, as well as converting some operations into another, we successfully reduced four dummy field negations for point doubling, one non-dummy and five dummy field negations for point addition, and two non-dummy and six dummy field negations for point tripling.

To summarize, we propose new atomic patterns for the following cases:

- Left-to-right binary using general point doubling and mixed point addition
- Right-to-left binary using modified Jacobian projective coordinates point doubling and general point addition
- Special case scalar multiplication when $a = -3$ (also applicable to binary-ternary base)

Our first two contributions are for scalar d represented as binary number. Thus, point arithmetic operations involved are point doubling and point addition. On the other hand, our third contribution is for scalar d either represented as binary number or double-base chain using mixed power of two and three. Hence, point arithmetic involved might include point tripling if double-base chain representation is used.

2. Preliminaries

2.1 Elliptic Curve Cryptography

An elliptic curve E over a field K is defined by the equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

where $a_1, a_2, a_3, a_4, a_6 \in K$.

If prime field is used, equation (1) can be simplified to

$$y^2 = x^3 + ax + b \quad (2)$$

where $a, b \in K$, and $\Delta = 4a^3 + 27b^2 \neq 0$.

Scalar multiplication, the operation that computes $[d]P = P + \cdots + P$ (d times) for a given point P on the elliptic curve E and a secret scalar d, is one of the main operations in elliptic curve cryptography. A commonly used method to compute scalar multiplication is double-and-add algorithm which consists of two point arithmetic operations, namely, point doubling and point addition. If the scalar d is expressed in double-base chain representation [4] [5] using mixed power of two and three, algorithms computing scalar multiplication also involve another point arithmetic operation called point tripling.

2.2 Point Arithmetic Operation

2.2.1 Doubling in Jacobian Coordinates

Let $P_1 = (X_1, Y_1, Z_1)$ be a point in Jacobian coordinates on the elliptic curve E. Point doubling $[2]P_1 = (X_2, Y_2, Z_2)$ can be computed as follows:

$$X_2 = \alpha_2^2 - 2\beta_2, \quad Y_2 = \alpha_2(\beta_2 - X_2) - 8Y_1^4, \quad Z_2 = 2Y_1Z_1$$

where $\alpha_2 = 3X_1^2 + aZ_1^4$, $\beta_2 = 4X_1Y_1^2$

The cost of point doubling is $4M + 6S$ where M and S are cost of field multiplication and field squaring respectively. In a special case of $a = -3$, α_2 can be computed more efficiently as follows:

$$3X_1^2 + aZ_1^4 = 3(X_1 + Z_1^2)(X_1 - Z_1^2)$$

We shall refer to point doubling that uses the above equation as *fast point doubling*. Accordingly, the cost of fast point doubling is decreased to $4M + 4S$.

2.2.2 Addition in Jacobian Coordinates

Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points in Jacobian coordinates on the elliptic curve E. Point addition $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$ can be computed as follows:

$$X_3 = \alpha_3^2 - \beta_3^3 - 2X_1Z_2^2\beta_3^2,$$
$$Y_3 = \alpha_3(X_1Z_2^2\beta_3^2 - X_3) - Y_1Z_2^3\beta_3^3$$
$$Z_3 = Z_1Z_2\beta_3$$

where $\alpha_3 = Y_2Z_1^3 - Y_1Z_2^3$, $\beta_3 = X_2Z_1^2 - X_1Z_2^2$

Thus, the cost of point addition is $12M + 4S$.

2.2.3 Addition in Mixed Jacobian-Affine Coordinates

Cohen, Miyaji, and Ono [6] introduced a computation in mixed coordinate systems. Addition in mixed Jacobian-affine coordinates or *mixed point addition* is computed by adding one point in Jacobian coordinates to the other point in affine coordinates.

Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be points on the elliptic curve E in Jacobian and affine coordinates respectively. Point addition $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$ can be computed as follows:

$$X_3 = \alpha_3^2 - \beta_3^3 - 2X_1\beta_3^2,$$
$$Y_3 = \alpha_3(X_1\beta_3^2 - X_3) - Y_1\beta_3^3$$
$$Z_3 = Z_1\beta_3$$

where $\alpha_3 = Y_2 Z_1^3 - Y_1$, $\beta_3 = X_2 Z_1^2 - X_1$

Since point P_2 is in affine coordinates which has its Z coordinate or Z_2 equals to 1, computing $Z_2^2, Z_2^3, Z_1 Z_2, X_1 Z_2^2$ can be omitted. Hence, the cost of mixed point addition is reduced to $8M + 3S$.

2. 2. 4 Tripling in Jacobian Coordinates

Let $P_1 = (X_1, Y_1, Z_1)$ be a point in Jacobian coordinates on the elliptic curve E. Point tripling $[3]P_1 = (X_3, Y_3, Z_3)$ can be computed as follows:

$$X_3 = 8Y_1^2(\beta_3 - \alpha_3) + X_1\omega_3^2,$$
$$Y_3 = Y_1[4(\alpha_3 - \beta_3)(2\beta_3 - \alpha_3) - \omega_3^3],$$
$$Z_3 = 2Z_1\omega_3$$

where $\alpha_3 = \theta_3\omega_3$, $\beta_3 = 8Y_1^4$, $\theta_3 = 3X_1^2 + aZ_1^4$, $\omega_3 = 12X_1Y_1^2 - \theta_3^2$

The cost of point tripling is $9M + 7S$. Again, in a special case of $a = -3$, θ_3 can be computed more efficiently as follows:

$$3X_1^2 + aZ_1^4 = 3(X_1 + Z_1^2)(X_1 - Z_1^2)$$

We shall refer to point tripling that uses the above equation as *fast point tripling*. Therefore, the cost of fast point tripling is decreased to $9M + 5S$.

2. 3 Side-Channel Analysis

To break the cryptosystem, apart from investigating mathematical or theoretical weaknesses of the algorithms, there exists another type of attack which analyses information leakage through physical devices. This kind of attack is referred to as *side-channel attack* or *side-channel analysis* or SCA.

Simple SCA refers to processes where only a single input is used to obtain side-channel information and to derive the underlying secret. *Differential* SCA, on the other hand, uses several inputs together with statistical techniques to analyse patterns observed from the leakage.

Chevallier-Mames, Ciet, and Joye [1] proposed a method to prevent simple side-channel analysis. Instead of making all the processes indistinguishable from one other, they rewrote those processes in forms of sequences of indistinguishable blocks of instructions or *side-channel atomic block*. Dimitrov, Imbert, and Mishra [5] derived new formulas for (consecutive) point tripling(s) and also showed the atomic patterns for those formulas. Longa [2] further improved those patterns. Giraud and Verneuil [3] introduced atomic patterns for embedded devices.

3. Overlapping Atomic Block

It has previously been described in [3], [7] that if using random curve isomorphism as a countermeasure for differential SCA, it is not possible to use fast point doubling because value a is random. This implies that general point doubling must be used instead. After examining the general point doubling and mixed point addition formulas, we observed that the number of field multiplications plus field squarings in those two formulas differs by one and the total number of field operations differs by four.

To be more precisely, general point doubling requires 4 field multiplications, 6 field squarings, 13 field additions, and 3 field negations which sum up to 26 field operations; mixed point addition requires 8 field multiplications, 3 field squarings, 7 field additions, and 4 field negations which sum up to 22 field operations. Because field squaring can be computed using field multiplication, we first focused our attention to the sum of field multiplication and field squaring necessary. By replacing some field squarings by field multiplications, it is possible to rearrange field arithmetic operations of general point doubling and mixed point addition into similar sequences.

Another important observation that we noticed was mixed point addition is never consecutively computed twice. In other words, mixed point addition never follows mixed point addition. It always follows general point doubling. Also, scalar multiplication never starts with mixed point addition. It always starts with general point doubling.

Since the total number of field multiplications plus field squarings is *nearly* but *not exactly* the same, we did not want to add any extra field multiplications to make both formulas have the total number of field multiplications plus field squarings equals. Instead, we defined *two* atomic patterns having the number of field multiplications plus field squarings equals to 10 and 11. We shall refer to as pattern 1: $M + S = 10$ and pattern 2: $M + S = 11$ respectively.

It is obvious that general point doubling can be expressed by using either of those patterns because the total number of field multiplications plus field squarings required for point doubling is 10 and either of those patterns has at least 10. However, mixed point addition cannot complete its computation unless it uses pattern 2 having $M + S = 11$.

Due to the fact that mixed point addition is never computed consecutively, that is, mixed point addition is always computed after general point doubling, the $M + S = 11$ pattern is never needed for two consecutive blocks. In this way, we can simply use those two patterns alternately. Figure 1 (Top) illustrates the idea of using two patterns alternately.

If general point doubling is executed using pattern $M+S = 11$, we would use that extra field multiplication to compute one multiplication from mixed point addition formula as if mixed point addition is really executed afterwards. In a lucky case where mixed point addition is computed afterwards, we waste no computation. In a not-so-lucky case

where mixed point addition is not computed afterwards, we waste one ahead-computation of field multiplication. Figure 1 (Bottom) illustrates the idea of ahead-computation if point doubling executed using pattern $M + S = 11$.

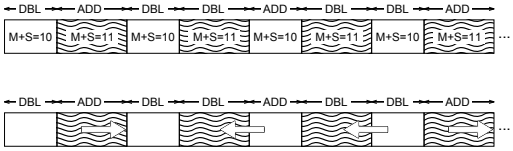

図 1 (Top)Using two patterns alternately
(Bottom)Computing ahead if doubling using $M + S = 11$

Table 1 shows the atomic patterns for general point doubling both pattern 1 and 2. These two patterns are quite similar for point doubling. The atomic pattern 1: $M + S = 10$ consists of 29 field operations while the atomic pattern 2: $M + S = 11$ consists of 30 field operations. The difference between these two patterns is at the last field operation shown in parenthesis. That is, pattern 1 does not contain this operation. Please note that the last operation ($T_6 = T_2 \times Z_1 = Y_2 Z_1$) is for mixed point addition. If mixed point addition is not performed afterwards, this values would be discarded. In our patterns, $\star$ indicates dummy operations.

表 1 General point doubling

Input: $P = (X_1, Y_1, Z_1)$
Output: $2P = (X_2, Y_2, Z_2)$
$T_1 \leftarrow X_1, \quad T_2 \leftarrow Y_1, \quad T_3 \leftarrow Z_1$

$T_4 = T_1^2$	(X_1^2)
$T_5 = T_2^2$	(Y_1^2)
$\star$	
$T_6 = T_4 + T_4$	$(2X_1^2)$
$T_7 = T_3^2$	(Z_1^2)
$T_7 = T_7 \times T_7$	(Z_1^4)
$T_7 = a \times T_7$	(aZ_1^4)
$\star$	
$T_4 = T_4 + T_6$	$(3X_1^2)$
$T_4 = T_4 + T_7$	$(\alpha) \quad *1$
$T_6 = T_4^2$	(α^2)
$T_7 = T_1 \times T_5$	$(X_1 Y_1^2)$
$\star$	
$T_7 = T_7 + T_7$	$(2X_1 Y_1^2)$

$T_3 = T_2 \times T_3$	$(Y_1 Z_1)$
$T_7 = T_7 + T_7$	$(4X_1 Y_1^2) = \beta$
$T_7 = -T_7$	$(-\beta)$
$T_3 = T_3 + T_3$	$(2Y_1 Z_1 = Z_2)$
$T_1 = T_7 + T_7$	(-2β)
$T_1 = T_1 + T_6$	$(\alpha^2 - 2\beta = X_2)$
$T_6 = T_1 + T_7$	$(X_2 - \beta)$
$T_6 = -T_6$	$(\beta - X_2)$
$T_5 = T_5^2$	(Y_1^4)
$T_4 = T_4 \times T_6$	$(\alpha(\beta - X_2))$
$T_5 = T_5 + T_5$	$(2Y_1^4)$
$T_5 = T_5 + T_5$	$(4Y_1^4)$
$T_5 = T_5 + T_5$	$(8Y_1^4)$
$T_5 = -T_5$	$(-8Y_1^4)$
$T_2 = T_4 + T_5$	$(Y_2) \quad *2$
$(T_6 = T_2 \times Z_1)$	$(Y_2 Z_1)$

$*1 : 3X_1^2 + aZ_1^4 = \alpha$

$*2 : \alpha(\beta - X_2) - 8Y_1^4 = Y_2$

Table 2 shows the atomic pattern for mixed point addition using pattern 1: $M+S = 10$. Due to the fact pattern 1 and 2 are used alternately and there is no consecutive mixed point additions, the previous operation before this mixed point addition using pattern $M + S = 10$ must be general point doubling using pattern $M + S = 11$. In this case, the ahead-computed value $Y_2 Z_1$ in the previous general point doubling will be used in this atomic block of mixed point addition.

Table 3 shows the atomic pattern for mixed point addition using pattern 2: $M + S = 11$. In this case, mixed point addition can complete its computation within its block. No value from previous block is used. In fact, there is no ahead-computation in the previous block because operation that comes before mixed point addition using pattern $M+S = 11$ is general point doubling using pattern $M + S = 10$ which has no extra field multiplication.

表 2 Mixed point addition using pattern 1: $M + S = 10$

Input: $P_1 = (X_1, Y_1, Z_1), \quad P_2 = (X_2, Y_2, 1)$
Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$
$T_1 \leftarrow X_1, \quad T_2 \leftarrow Y_1, \quad T_3 \leftarrow Z_1$
$T_4 \leftarrow X_2, \quad T_5 \leftarrow Y_2$

$T_7 = T_3^2$	(Z_1^2)
$T_4 = T_4 \times T_7$	$(X_2 Z_1^2)$
$T_1 = -T_1$	$(-X_1)$
$T_4 = T_1 + T_4$	$(\beta) \quad *1$
$T_8 = T_4^2$	(β^2)
$T_3 = T_3 \times T_4$	$(Z_1\beta = Z_3)$
$T_6 = T_6 \times T_7$	$(Y_2 Z_1^3)$
$T_2 = -T_2$	$(-Y_1)$
$T_6 = T_2 + T_6$	$(\alpha) \quad *2$
$\star$	
$T_7 = T_6^2$	(α^2)
$T_1 = T_1 \times T_8$	$(-X_1\beta^2)$
$T_8 = -T_8$	$(-\beta^2)$
$T_7 = T_1 + T_7$	$(\alpha^2 - X_1\beta^2)$

$T_8 = T_4 \times T_8$	$(-\beta^3)$
$T_7 = T_7 + T_8$	$(\alpha^2 - \beta^3 - X_1\beta^2)$
$\star$	
$T_7 = T_1 + T_7$	$(X_3) \quad *3$
$T_1 = T_1 + T_7$	$(X_3 - X_1\beta^2)$
$\star$	
$\star$	
$T_1 = -T_1$	$(X_1\beta^2 - X_3)$
$T_5 = T_1 \times T_6$	$(\alpha(X_1\beta^2 - X_3))$
$T_8 = T_2 \times T_8$	$(Y_1\beta^3)$
$\star$	
$\star$	
$T_8 = -T_8$	$(-Y_1\beta^3)$
$T_2 = T_5 + T_8$	$(Y_3) \quad *4$
$\star$	

$*1 : X_2 Z_1^2 - X_1 = \beta$

$*2 : Y_2 Z_1^3 - Y_1 = \alpha$

$*3 : \alpha^2 - \beta^3 - 2X_1\beta^2 = X_3$

$*4 : \alpha(X_1\beta^2 - X_3) - Y_1\beta^3 = Y_3$

According to [3], they assumed the cost of field operation to be $A/M = 0.2$ and $S/M = 0.8$, and they also used NAF representation, i.e., n-bit scalar having an average Hamming weight of $n/3$. Thus, it implies that point doubling is executed n times and point addition is executed $n/3$ times. With these assumptions, the best average cost per bit to compute left-to-right scalar multiplication is $17.7M$ [3].

Our proposed atomic pattern 1: $M + S = 10$ consists of 7 field multiplications, 3 field squarings, 13 field additions, and 6 field negations. Thus, the cost per one block is $(7 \times 1M) + (3 \times 0.8M) + (13 \times 0.2M) + (6 \times 0.1M) = 12.6M$. Our proposed atomic pattern 2: $M + S = 11$ consists of 8 field multiplications, 3 field squarings, 13 field additions, and 6 field negations. Thus, the cost per one block is $(8 \times 1M) + (3 \times 0.8M) + (13 \times 0.2M) + (6 \times 0.1M) = 13.6M$.

Consider atomic block that allows to compute both point doubling and point addition, we simply think of it as the average of these two patterns. Therefore, the average cost is computed as $\frac{1}{2}(12.6M + 13.6M) = 13.1M$. Now, we can roughly say that one block of our atomic pattern is sufficient

表 3 Mixed point addition using pattern 2: $M + S = 11$

Input: $P_1 = (X_1, Y_1, Z_1)$, $P_2 = (X_2, Y_2, 1)$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

$T_1 \leftarrow X_1$, $T_2 \leftarrow Y_1$, $T_3 \leftarrow Z_1$

$T_4 \leftarrow X_2$, $T_5 \leftarrow Y_2$

$T_6 = T_3^2$	(Z_1^2)
$T_4 = T_4 \times T_6$	$(X_2 Z_1^2)$
$T_1 = -T_1$	$(-X_1)$
$T_4 = T_1 + T_4$	(β) $*1$
$T_7 = T_4^2$	(β^2)
$T_6 = T_3 \times T_6$	(Z_1^3)
$T_5 = T_5 \times T_6$	$(Y_2 Z_1^3)$
$T_2 = -T_2$	$(-Y_1)$
$T_5 = T_2 + T_5$	(α) $*2$
$\star$	
$T_6 = T_5^2$	(α^2)
$T_1 = T_1 \times T_7$	$(-X_1\beta^2)$
$T_7 = -T_7$	$(-\beta^2)$
$T_6 = T_1 + T_6$	$(\alpha^2 - X_1\beta^2)$

$T_7 = T_4 \times T_7$	$(-\beta^3)$
$T_6 = T_6 + T_7$	$(\alpha^2 - \beta^3 - X_1\beta^2)$
$\star$	
$T_6 = T_1 + T_6$	(X_3) $*3$
$T_1 = T_1 + T_6$	$(X_3 - X_1\beta^2)$
$\star$	
$\star$	
$T_1 = -T_1$	$(X_1\beta^2 - X_3)$
$T_5 = T_1 \times T_5$	$(\alpha(X_1\beta^2 - X_3))$
$T_2 = T_2 \times T_7$	$(Y_1\beta^3)$
$\star$	
$\star$	
$T_2 = -T_2$	$(-Y_1\beta^3)$
$T_2 = T_2 + T_5$	(Y_3) $*4$
$\star$	
$T_3 = T_3 \times T_4$	$(Z_1\beta = Z_3)$

$*1 : X_2 Z_1^2 - X_1 = \beta$

$*2 : Y_2 Z_1^3 - Y_1 = \alpha$

$*3 : \alpha^2 - \beta^3 - 2X_1\beta^2 = X_3$

$*4 : \alpha(X_1\beta^2 - X_3) - Y_1\beta^3 = Y_3$

to compute either point doubling or point addition. Regarding NAF representation, average cost per bit to compute left-to-right scalar multiplication using our atomic patterns is $13.1M + \frac{1}{3}(13.1M) = 17.46M$. Comparing this cost to the previous result, we achieved an improvement of 1.36%

4. Two-Pattern Sequence

Giraud and Verneuil [3] proposed new atomic patterns right-to-left scalar multiplication for embedded devices. They described that in embedded devices the cost of field addition could not be neglected and the ratio between modular addition and modular multiplication (A/M) could be as large as 0.3. Due to the non-negligible cost of field addition and the fact that three extra field additions were required for each S-M trade-off [2], they thought that the trade-off was unprofitable.

However, we examined their patterns and found that there were many dummy field additions in point addition pattern. We also noticed that in their doubling pattern, they used 6 field multiplications and 2 field squarings ($6M + 2S$) instead of 4 field multiplications and 4 field squarings ($4M + 4S$) as are commonly used. This means that some field squarings were unnecessarily replaced by field multiplications. Please note that they used modified Jacobian projective coordinates for point doubling, and they merged field negation and field addition into field subtraction. We also followed their setting.

Our idea is to utilize those dummy operations for three extra field additions so that S-M trade-off can still be profitable. We observed that the number of field multiplications plus field squarings needed for point addition is twice as many as for point doubling, i.e., 16 and 8. Hence, the number of atomic blocks required for point addition is two times that of point doubling. If S-M trade-off is applied to point addition, the number of field multiplications and field squarings required would change to 11 and 5 respectively. Since 11 and 5 are obviously not even numbers, we can not simply express atomic blocks for point doubling as half the number of the atomic blocks for point addition.

To cope with this problem, we defined *two* atomic patterns, that is, one containing 5 field multiplications and 3 field squarings ($5M + 3S$) and the other containing 6 field multiplications and 2 field squarings ($6M + 2S$). We shall now refer to as pattern 1 and pattern 2 respectively. It is trivial to see that point addition can be expressed using the combination of pattern 1 and pattern 2 because $(5M + 3S) + (6M + 2S) = 11M + 5S$ which is exactly the same number of field multiplications and field squarings required. It is also possible to express point doubling using either pattern 1: $5M + 3S$ or pattern 2: $6M + 2S$ because point doubling requires $4M + 4S$ and field squaring can always be executed using field multiplication.

Since we now have two different atomic patterns, in order to withstand the simple SCA, it is necessary to ensure that sequences of these two patterns reveal no information related to which patterns belong to which point operations. One solution is to use pattern 1 and pattern 2 alternately regardless of what point operation is being executed. Figure 2 illustrates the idea of alternately switching two patterns.

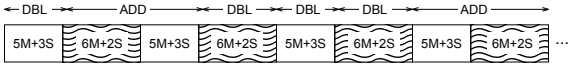

図 2 Using two patterns alternately regardless of what point operation is being executed to prevent simple SCA

In this way, point addition must be able to begin its execution with either pattern 1 or pattern 2. As a consequence, the following atomic patterns are required:

- Point doubling using $5M + 3S$
- Point doubling using $6M + 2S$
- Point addition using $5M + 3S$ then $6M + 2S$
- Point addition using $6M + 2S$ then $5M + 3S$

We shall now present our new atomic patterns for the above four cases. Table 4 shows the atomic patterns for point doubling. Table 5 shows the atomic pattern for point addition using pattern 1: $5M + 3S$ then pattern 2: $6M + 2S$. Table 6 shows the atomic pattern for point addition using pattern 2: $6M + 2S$ then pattern 1: $5M + 3S$. Again, $\star$

indicates dummy operations.

表 4 Two atomic patterns for point doubling

Input: $P = (X_1, Y_1, Z_1, W_1)$

Output: $2P = (X_2, Y_2, Z_2, W_2)$

$T_1 \leftarrow X_1, \quad T_2 \leftarrow Y_1, \quad T_3 \leftarrow Z_1, \quad T_4 \leftarrow W_1$

Pattern 1: $5M + 3S$			Pattern 2: $6M + 2S$	
$T_5 = T_1^2$	(X_1^2)	1	$T_5 = T_1^2$	(X_1^2)
$T_6 = T_2 + T_2$	$(2Y_1)$	2	$T_6 = T_2 + T_2$	$(2Y_1)$
$T_3 = T_3 \times T_6$	$(2Y_1Z_1 = Z_2)$	3	$T_3 = T_3 \times T_6$	$(2Y_1Z_1 = Z_2)$
$T_7 = T_5 + T_5$	$(2X_1^2)$	4	$T_7 = T_5 + T_5$	$(2X_1^2)$
$T_2 = T_2 \times T_6$	$(2Y_1^2)$	5	$T_2 = T_2 \times T_6$	$(2Y_1^2)$
$T_6 = T_2 + T_2$	$(4Y_1^2)$	6	$T_6 = T_2 + T_2$	$(4Y_1^2)$
$T_2 = T_2^2$	$(4Y_1^4)$	7	$T_2 = T_2 \times T_6$	$(8Y_1^4 = \gamma)$
$T_2 = T_2 + T_2$	$(8Y_1^4 = \gamma)$	8	$T_5 = T_5 + T_7$	$(3X_1^2)$
$T_5 = T_5 + T_7$	$(3X_1^2)$	9	$T_5 = T_5 + T_4$	$(3X_1^2 + W_1 = \alpha)$
$T_5 = T_4 + T_5$	$(3X_1^2 + W_1 = \alpha)$	10		
$T_7 = T_5^2$	(α^2)	11	$T_7 = T_5^2$	(α^2)
$T_6 = T_1 \times T_6$	$(4X_1Y_1^2 = \beta)$	12	$T_6 = T_1 \times T_6$	$(4X_1Y_1^2 = \beta)$
$T_4 = T_4 + T_4$	$(2W_1)$	13	$T_4 = T_4 + T_4$	$(2W_1)$
$T_7 = T_7 - T_6$	$(\alpha^2 - \beta)$	14	$T_7 = T_7 - T_6$	$(\alpha^2 - \beta)$
$T_4 = T_2 \times T_4$	$(2\gamma W_1 = W_2)$	15	$T_4 = T_2 \times T_4$	$(2\gamma W_1 = W_2)$
$T_1 = T_7 - T_6$	$(\alpha^2 - 2\beta = X_2)$	16	$T_1 = T_7 - T_6$	$(\alpha^2 - 2\beta = X_2)$
$T_6 = T_6 - T_1$	$(\beta - X_2)$	17	$T_7 = T_6 - T_1$	$(\beta - X_2)$
$T_6 = T_5 \times T_6$	$(\alpha(\beta - X_2))$	18	$T_5 = T_5 \times T_7$	$(\alpha(\beta - X_2))$
$T_2 = T_6 - T_2$	$(Y_2) \quad *1$	19	$T_2 = T_5 - T_2$	$(Y_2) \quad *2$

$*1 : \alpha(\beta - X_2) - 8Y_1^4 = Y_2$

$*2 : \alpha(\beta - X_2) - 8Y_1^4 = Y_2$

Note that the differences between patterns 1: $5M + 3S$ and pattern 2: $6M + 2S$ are at step 7 and 8. Field squaring at step 7 and field addition at step 8 of pattern 1: $5M + 3S$ are combined to field multiplication in step 7 of pattern 2: $6M + 2S$.

Again, we used NAF representation and assumed the cost of field operation to be $A/M = 0.2$ and $S/M = 0.8$. With these assumptions, average cost per bit to compute scalar multiplication using their atomic patterns is $16M$ [3].

In our atomic patterns, there are 5 field multiplications, 3 field squarings, and 11 field additions in pattern 1 and 6 field multiplications, 2 field squarings, and 10 field additions in pattern 2. Under the above assumption, cost per pattern for both patterns is $9.6M$. One pattern is needed for point doubling while two are needed for point addition. Therefore, the cost for point doubling and point addition are $9.6M$ and $19.2M$ respectively. With NAF representation, average cost per bit to compute scalar multiplication using our atomic patterns is $16M$.

The reason our average cost per bit equals to theirs even though we traded one field squaring for one field multiplication is that using 5 field multiplications for point doubling resulted in an extra field addition required, that is, 11 field additions are needed instead of 10. Thus, $0.2M$ gained from S-M trade-off was used up for the extra field addition.

表 5 Addition using pattern $5M + 3S$ then $6M + 2S$

Input: $P_1 = (X_1, Y_1, Z_1), \quad P_2 = (X_2, Y_2, Z_2)$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

$T_1 \leftarrow X_1, \quad T_2 \leftarrow Y_1, \quad T_3 \leftarrow Z_1$

$T_4 \leftarrow X_2, \quad T_5 \leftarrow Y_2, \quad T_6 \leftarrow Z_2$

Pattern 1: $5M + 3S$		Pattern 2: $6M + 2S$	
$T_7 = T_3^2$	(Z_1^2)	$T_7 = T_7^2$	$(4\beta^2)$
$T_8 = T_3 + T_6$	$(Z_1 + Z_2)$	$\star$	
$T_4 = T_4 \times T_7$	$(X_2Z_1^2)$	$T_2 = T_2 \times T_6$	$(Y_1Z_2^3)$
$\star$		$T_5 = T_5 - T_2$	$(\alpha) \quad *4$
$T_3 = T_3 \times T_7$	(Z_1^3)	$T_1 = T_1 \times T_7$	$(4X_1Z_2^2\beta^2)$
$\star$		$T_5 = T_5 + T_5$	(2α)
$T_9 = T_6^2$	(Z_2^2)	$T_7 = T_4 \times T_7$	$(4\beta^3)$
$\star$		$T_3 = T_7 + T_7$	$(8\beta^3)$
$\star$		$T_9 = T_1 + T_1$	$(8X_1Z_2^2\beta^2)$
$\star$			
$T_8 = T_8^2$	$((Z_1 + Z_2)^2)$	$T_6 = T_5^2$	$(4\alpha^2)$
$T_1 = T_1 \times T_9$	$(X_1Z_2^2)$	$T_2 = T_2 \times T_3$	$(8Y_1Z_2^3\beta^3)$
$T_8 = T_8 - T_7$	$(*1)$	$T_6 = T_6 - T_7$	$(4\alpha^2 - 4\beta^3)$
$T_8 = T_8 - T_9$	$(2Z_1Z_2) \quad *2$	$T_6 = T_6 - T_9$	$(X_3) \quad *5$
$T_6 = T_6 \times T_9$	(Z_2^3)	$T_8 = T_4 \times T_8$	$(2Z_1Z_2\beta = Z_3)$
$T_4 = T_4 - T_1$	$(\beta) \quad *3$	$T_1 = T_1 - T_6$	$(4X_1Z_2^2\beta^2 - X_3)$
$T_7 = T_4 + T_4$	(2β)	$\star$	
$T_5 = T_3 \times T_5$	$(Y_2Z_1^3)$	$T_5 = T_1 \times T_5$	$(*6)$
$\star$		$T_2 = T_5 - T_2$	$(Y_3) \quad *7$

$*1 : (Z_1 + Z_2)^2 - Z_1^2$

$*2 : (Z_1 + Z_2)^2 - Z_1^2 - Z_2^2 = 2Z_1Z_2$

$*3 : X_2Z_1^2 - X_1Z_2^2 = \beta$

$*4 : Y_2Z_1^3 - Y_1Z_2^3 = \alpha$

$*5 : 4\alpha^2 - 4\beta^3 - 8X_1Z_2^2\beta^2 = X_3$

$*6 : 2\alpha(4X_1Z_2^2\beta^2 - X_3)$

$*7 : 2\alpha(4X_1Z_2^2\beta^2 - X_3) - 8Y_1Z_2^3\beta^3 = Y_3$

However, our atomic patterns are still profitable if the cost of field addition plus field squaring is less than that of field multiplication, that is, $S + A < M$. According to [3], the A/M ratio on smart cards with crypto-coprocessor is less than 0.2 when the bit-length used is not shorter than 320. In other words, our atomic patterns show improvement when using at least 320-bit scalar. Table 7 shows the improvements according to the bit-length used.

5. Special Case Point Operation

Longa and Miri [2] proposed new atomic patterns for the special case of $a = -3$. Their atomic structure for traditional scalar multiplication, namely, involving only point doubling and point addition was S-N-A-M-N-A-A. This structure composed of 1 field multiplication, 1 field squaring, 3 field additions, and 2 field negations. For scalar multiplication involving point tripling, their atomic structure was S-N-A-A-M-N-A-A. This structure had one more field addition, that is, 1 field multiplication, 1 field squaring, 4 field additions, and 2 field negations.

We studied their atomic structures and found that there were 4 dummy negations in point doubling and 5 dummy negations in point addition. We also observed that they used

表 6　Addition using pattern $6M + 2S$ then $5M + 3S$

Input: $P_1 = (X_1, Y_1, Z_1)$,　$P_2 = (X_2, Y_2, Z_2)$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

$T_1 \leftarrow X_1$,　$T_2 \leftarrow Y_1$,　$T_3 \leftarrow Z_1$

$T_4 \leftarrow X_2$,　$T_5 \leftarrow Y_2$,　$T_6 \leftarrow Z_2$

Pattern 2: $6M + 2S$		Pattern 1: $5M + 3S$	
$T_7 = T_3^2$	(Z_1^2)	$T_3 = T_3^2$	$(4\beta^2)$
$T_8 = T_3 + T_6$	$(Z_1 + Z_2)$	$T_5 = T_5 + T_5$	(2α)
$T_4 = T_4 \times T_7$	$(X_2 Z_1^2)$	$T_1 = T_1 \times T_3$	$(4X_1 Z_2^2 \beta^2)$
$\star$		$T_6 = T_1 + T_1$	$(8X_1 Z_2^2 \beta^2)$
$T_3 = T_3 \times T_7$	(Z_1^3)	$T_3 = T_3 \times T_4$	$(4\beta^3)$
$\star$		$T_{10} = T_3 + T_3$	$(8\beta^3)$
$T_5 = T_3 \times T_5$	$(Y_2 Z_1^3)$	$T_8 = T_8^2$	$((Z_1 + Z_2)^2)$
$\star$		$T_8 = T_8 - T_7$	$(*3)$
$\star$		$T_8 = T_8 - T_9$	$(2Z_1 Z_2)$　$*4$
		$\star$	
$T_9 = T_6^2$	(Z_2^2)	$T_7 = T_5^2$	$(4\alpha^2)$
$T_1 = T_1 \times T_9$	$(X_1 Z_2^2)$	$T_2 = T_2 \times T_{10}$	$(8Y_1 Z_2^3 \beta^3)$
$T_4 = T_4 - T_1$	(β)　$*1$	$T_7 = T_7 - T_3$	$(4\alpha^2 - 4\beta^3)$
$T_3 = T_4 + T_4$	(2β)	$T_7 = T_7 - T_6$	(X_3)　$*5$
$T_6 = T_6 \times T_9$	(Z_2^3)	$T_8 = T_4 \times T_8$	$(2Z_1 Z_2 \beta = Z_3)$
$\star$		$T_1 = T_1 - T_7$	$(4X_1 Z_2^2 \beta^2 - X_3)$
$\star$		$\star$	
$T_2 = T_2 \times T_6$	$(Y_1 Z_2^3)$	$T_5 = T_1 \times T_5$	$(*6)$
$T_5 = T_5 - T_2$	(α)　$*2$	$T_5 = T_5 - T_2$	(Y_3)　$*7$

$*1 : X_2 Z_1^2 - X_1 Z_2^2 = \beta$

$*2 : Y_2 Z_1^3 - Y_1 Z_2^3 = \alpha$

$*3 : (Z_1 + Z_2)^2 - Z_1^2$

$*4 : (Z_1 + Z_2)^2 - Z_1^2 - Z_2^2 = 2Z_1 Z_2$

$*5 : 4\alpha^2 - 4\beta^3 - 8X_1 Z_2^2 \beta^2 = X_3$

$*6 : 2\alpha(4X_1 Z_2^2 \beta^2 - X_3)$

$*7 : 2\alpha(4X_1 Z_2^2 \beta^2 - X_3) - 8Y_1 Z_2^3 \beta^3 = Y_3$

表 7　Improvement of atomic pattern for right-to-left scalar multiplication corresponding to bit-length used

Bit-length	A/M	Cost per bit		Improvement
		[3]	Our results	
320	0.16	15.33	15.3	0.22%
384	0.13	14.83	14.78	0.39%
512	0.09	14.17	14.08	0.65%
521	0.09	14.17	14.08	0.65%

4 and 6 atomic blocks to express point doubling and point addition respectively.

After carefully examining the number of field multiplications, field squarings, field additions, and field negations required for each point operation, we discovered that those operations can be rearranged so that point doubling and mixed point addition use $4M + 4S + 12A + 4N$ and $6M + 6S + 17A + 6N$ field operations respectively. For this reason, we introduce new atomic patterns for the traditional scalar multiplication consisting of 1 field multiplication, 1 field squaring, 3 field additions, and 1 field negation, that is, having the structure S-A-N-A-M-A.

For point tripling, we found that it could be expressed using $7M+7S+23A+7S$ field operations. It is easy to see that if each block composed of 1 field multiplication, 1 field squaring, 3 field additions, and 1 field negation, eight blocks would be required to compute point tripling. Moreover, there exists dependency among field operation sequences. That is, some certain operations must be computed before some certain operations. As a consequence, eight blocks were unavoidably required.

Our atomic structure for scalar multiplication involving point tripling has a structure of S-A-N-A-M-A-A. Note that there is one more field addition compared to the structure involving only point doubling and point addition.

Table 8, Table 9, and Table 10 show the atomic patterns for point doubling, point addition, and point tripling respectively.

表 8　Atomic point doubling

Input: $P = (X_1, Y_1, Z_1)$

Output: $2P = (X_2, Y_2, Z_2)$

$T_1 \leftarrow X_1$,　$T_2 \leftarrow Y_1$,　$T_3 \leftarrow Z_1$

$T_4 = T_3^2$	(Z_1^2)	$T_6 = T_5^2$	(α^2)
$T_5 = T_1 + T_4$	$(X_1 + Z_1^2)$	$T_6 = T_6 + T_7$	$(\alpha^2 - \beta)$
$T_4 = -T_4$	$(-Z_1^2)$	$T_5 = -T_5$	$(-\alpha)$
$T_6 = T_1 + T_4$	$(X_1 - Z_1^2)$	$T_6 = T_6 + T_7$	$(\alpha^2 - 2\beta = X_2)$
$T_5 = T_5 \times T_6$	$(X_1^2 - Z_1^4)$	$T_3 = T_2 \times T_3$	$(Y_1 Z_1)$
$T_6 = T_5 + T_5$	$(2(X_1^2 - Z_1^4))$	$T_3 = T_3 + T_3$	$(2Y_1 Z_1 = Z_2)$
$T_4 = T_2^2$	(Y_1^2)	$T_4 = T_4^2$	$(4Y_1^4)$
$T_4 = T_4 + T_4$	$(2Y_1^2)$	$T_4 = T_4 + T_4$	$(8Y_1^4)$
$T_1 = -T_1$	$(-X_1)$	$T_4 = -T_4$	$(-8Y_1^4)$
$T_1 = T_1 + T_1$	$(-2X_1)$	$T_1 = T_6 + T_7$	$(X_2 - \beta)$
$T_7 = T_1 \times T_4$	$(-\beta)$　$*1$	$T_5 = T_1 \times T_5$	$(\alpha(\beta - X_2))$
$T_5 = T_5 + T_6$	(α)　$*2$	$T_2 = T_4 + T_5$	(Y_2)　$*3$

$*1 : -4X_1 Y_1^2 = -\beta$

$*2 : 3(X_1^2 - Z_1^4) = \alpha$

$*3 : \alpha(\beta - X_2) - 8Y_1^4 = Y_2$

Atomic structure for scalar multiplication involving only point doubling and point addition proposed by Longa and Miri [2] is S-N-A-M-N-A-A. On the other hand, our new atomic structure is S-A-N-A-M-A. That is, we reduced one field negation for *each* block. To express point doubling and point addition, four and six atomic blocks are required. Therefore, we removed the total of *four* and *six* field negations for computing point doubling and point addition.

For scalar multiplication that involves point tripling, their atomic structure is S-N-A-A-M-N-A-A. Our atomic structure is S-A-N-A-M-A-A. In other words, we removed one field negation for *each* block. Four, six, and eight atomic blocks are required to express point doubling, point addition, and point tripling respectively. Hence, we decreased *four*, *six*, and *eight* field negations for computing point doubling, point addition, and point tripling respectively.

6. Conclusion

Side-channel atomic block has previously been proposed as a countermeasure for side-channel analysis. However, many

表 9　Atomic point addition

Input: $P_1 = (X_1, Y_1, Z_1)$,　$P_2 = (X_2, Y_2, 1)$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

$T_1 \leftarrow X_1$,　$T_2 \leftarrow Y_1$,　$T_3 \leftarrow Z_1$

$T_4 \leftarrow X_2$,　$T_5 \leftarrow Y_2$,　Y_2^2 is pre-computed

$T_6 = T_3^2$	(Z_1^2)	$T_3 = T_3^2$	$((Z_1+\beta)^2)$
$T_2 = T_2 + T_2$	$(2Y_1)$	$T_3 = T_3 + T_8$	$(2Z_1\beta = Z_3) * 4$
$T_1 = -T_1$	$(-X_1)$	$T_4 = -T_4$	$(-\beta)$
$\star$		$T_{10} = T_1 + T_1$	$(-4X_1\beta^2)$
$T_4 = T_4 \times T_6$	$(X_2 Z_1^2)$	$T_4 = T_4 \times T_7$	$(-2\beta^3)$
$T_4 = T_1 + T_4$	(β)　$*1$	$T_4 = T_4 + T_4$	$(-4\beta^3)$
$T_7 = T_4^2$	(β^2)	$T_7 = T_6^2$	(Z_1^6)
$T_8 = T_6 + T_7$	$(Z_1^2 + \beta^2)$	$T_1 = T_{10} + T_{10}$	$(-8X_1\beta^2)$
$T_9 = -T_2$	$(-2Y_1)$	$T_7 = -T_7$	$(-Z_1^6)$
$T_7 = T_7 + T_7$	$(2\beta^2)$	$T_5 = T_5 + T_7$	(2α)　$*5$
$T_6 = T_3 \times T_6$	(Z_1^3)	$T_2 = T_2 \times T_4$	$(-8Y_1\beta^3)$
$T_5 = T_5 + T_6$	$(Y_2 + Z_1^3)$	$T_4 = T_1 + T_4$	$(-4\beta^3 - 8X_1\beta^2)$
$T_5 = T_5^2$	$((Y_2 + Z_1^3)^2)$	$T_6 = T_5^2$	$(4\alpha^2)$
$T_5 = T_5 - Y_2^2$	$(*2)$	$T_6 = T_4 + T_6$	(X_3)　$*6$
$T_8 = -T_8$	$(-Z_1^2 - \beta^2)$	$T_5 = -T_5$	(-2α)
$T_5 = T_5 + T_9$	$(*3)$	$T_1 = T_6 + T_{10}$	$(*7)$
$T_1 = T_1 \times T_7$	$(-2X_1\beta^2)$	$T_5 = T_1 \times T_5$	$(*8)$
$T_3 = T_3 + T_4$	$(Z_1 + \beta)$	$T_2 = T_2 + T_5$	(Y_3)　$*9$

$*1 : X_2 Z_1^2 - X_1 = \beta$

$*2 : (Y_2 + Z_1^3)^2 - Y_2^2$

$*3 : (Y_2 + Z_1^3)^2 - Y_2^2 - 2Y_1$

$*4 : (Z_1+\beta)^2 - Z_1^2 - \beta^2 = 2Z_1\beta$

$*5 : 2Y_2 Z_1^3 - 2Y_1 = 2\alpha$

$*6 : 4\alpha^2 - 4\beta^3 - 8X_1\beta^2 = X_3$

$*7 : X_3 - 4X_1\beta^2$

$*8 : 2\alpha(4X_1\beta^2 - X_3)$

$*9 : 2\alpha(4X_1\beta^2 - X_3) - 8Y_1\beta^3 = Y_3$

表 10　Atomic point tripling

Input: $P = (X_1, Y_1, Z_1)$

Output: $3P = (X_2, Y_2, Z_2)$

$T_1 \leftarrow X_1$,　$T_2 \leftarrow Y_1$,　$T_3 \leftarrow Z_1$

$T_4 = T_3^2$	(Z_1^2)	$T_5 = T_5^2$	$((\theta-\omega)^2)$
$T_5 = T_1 + T_4$	$(X_1 + Z_1^2)$	$T_1 = T_1 + T_1$	$(2X_1)$
$T_6 = -T_4$	$(-Z_1^2)$	$T_7 = -T_7$	$(-\theta^2 - \omega^2)$
$T_7 = T_1 + T_6$	$(X_1 - Z_1^2)$	$T_1 = T_1 + T_1$	$(4X_1)$
$\star$		$T_1 = T_1 \times T_6$	$(4X_1\omega^2)$
$T_2 = T_2 + T_2$	$(2Y_1)$	$T_5 = T_5 + T_7$	$(-2\theta\omega = -2\alpha) * 3$
$\star$		$\star$	
$T_8 = T_2^2$	$(4Y_1^2)$	$T_8 = T_8^2$	$(16Y_1^4 = 2\beta)$
$T_9 = T_8 + T_8$	$(8Y_1^2)$	$T_6 = T_5 + T_8$	$(2\beta - 2\alpha)$
$T_3 = -T_3$	$(-Z_1)$	$T_7 = -T_6$	$(2\alpha - 2\beta)$
$T_{10} = T_8 + T_9$	$(12Y_1^2)$	$T_9 = T_9 + T_9$	$(16Y_1^2)$
$T_5 = T_5 \times T_7$	$(X_1^2 - Z_1^4)$	$T_9 = T_6 \times T_9$	$(16Y_1^2(2\beta - 2\alpha))$
$T_7 = T_5 + T_5$	$(2(X_1^2 - Z_1^4))$	$T_1 = T_1 + T_9$	(X_2)　$*4$
$T_5 = T_5 + T_7$	(θ)　$*1$	$\star$	
$T_7 = T_5^2$	(θ^2)	$T_3 = T_3^2$	$((Z_1+\omega)^2)$
$T_2 = T_2 + T_2$	$(4Y_1)$	$T_6 = T_6 + T_8$	$(4\beta - 2\alpha)$
$T_{10} = -T_{10}$	$(-12Y_1^2)$	$T_4 = -T_4$	$(-Z_1^2 - \omega^2)$
$T_2 = T_2 + T_2$	$(8Y_1)$	$T_3 = T_3 + T_4$	$(2Z_1\omega = Z_2)$
$T_{10} = T_1 \times T_{10}$	$(-12X_1Y_1^2)$	$T_6 = T_6 \times T_7$	$(*5)$
$T_{10} = T_7 + T_{10}$	$(-\omega)$　$*2$	$T_6 = T_6 + T_{12}$	$(*6)$
$\star$		$\star$	
$T_6 = T_{10}^2$	(ω^2)	$(T_4 = T_3^2)$	$(Z_2^2)^{**}$
$T_7 = T_6 + T_7$	$(\theta^2 + \omega^2)$	$(T_5 = T_1 + T_4)$	$(X_2 + Z_2^2)^{**}$
$T_{11} = -T_{10}$	(ω)	$(T_6 = -T_4)$	$(-Z_2^2)^{**}$
$T_4 = T_4 + T_6$	$(Z_1^2 + \omega^2)$	$(T_7 = T_1 + T_6)$	$(X_2 - Z_2^2)^{**}$
$T_{12} = T_6 \times T_{10}$	$(-\omega^3)$	$T_2 = T_2 \times T_6$	(Y_2)　$*7$
$T_3 = T_3 + T_{10}$	$(-Z_1 - \omega)$	$(T_2 = T_2 + T_2)$	$(2Y_2)^{**}$
$T_5 = T_5 + T_{10}$	$(\theta - \omega)$	$\star$	

$*1 : 3(X_1^2 - Z_1^4) = \theta$

$*2 : \theta^2 - 12X_1Y_1^2 = -\omega$

$*3 : (\theta-\omega)^2 - \theta^2 - \omega^2 = -2\theta\omega$

$*4 : 16Y_1^2(2\beta - 2\alpha) + 4X_1\omega^2 = X_2$

$*5 : 4(\alpha-\beta)(2\beta-\alpha)$

$*6 : 4(\alpha-\beta)(2\beta-\alpha) - \omega^3$

$*7 : 8Y_1[4(\alpha-\beta)(2\beta-\alpha) - \omega^3] = Y_2$

** indicates a merge of last and first block when point tripling is computed consecutively

dummy operations were introduced to make the overall process appeared as a sequence of identical atomic blocks. To deal with this problem, we introduce the idea of two-pattern atomic block and ahead computation.

Two-pattern atomic block is a technique of using two patterns together in a systematic way to still preserve the property of indistinguishability observed from side-channel. By increasing the choice of patterns, we decreased the number of dummy operations required to balance each atomic block.

Ahead computation is a technique of bringing operations in following blocks into previous blocks and computing them. "Bringing operations into previous blocks" can be done by either replacing dummy operations or rearranging operations sequences.

With these two techniques, we successfully reduced a number of operations required to compute scalar multiplication. This result can be applied to many scalar multiplication related algorithms.

文　　献

[1] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity," IEEE Transactions on Computer, vol.53, pp.760–768, 2004.

[2] P. Longa and A. Miri, "Fast and flexible elliptic curve point arithmetic over prime fields," IEEE Transactions on Computer, vol.57, pp.289–302, 2008.

[3] C. Giraud and V. Verneuil, "Atomicity improvement for elliptic curve scalar multiplication," Proceedings of the 9th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Application, pp.80–101, 2010.

[4] V.S. Dimitrov, G.A. Jullien, and W.C. Miller, "Theory and applications of the double-base number system," IEEE Transactions on Computer, vol.48, pp.1098–1106, 1999.

[5] V.S. Dimitrov, L. Imbert, and P.K. Mishra, "The double-base number system and its application to elliptic curve cryptography," Mathematics of Computation, vol.77, pp.1075–1104, 2008.

[6] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, pp.51–65, 1998.

[7] M. Joye and C. Tymen, "Protections against differential analysis for elliptic curve cryptography," Proceedings of the Third International Workshop on Cryptographic Hardward and Embedded Systems, pp.377–390, 2001.