A comparison of simple side-channel analysis countermeasures for variable-base elliptic curve scalar multiplication

Erick Nascimento[1], Rodrigo Abarzúa[2], Julio López[1], Ricardo Dahab[1]

[1] Institute of Computing, University of Campinas,
Av. Albert Einstein 1251, Campinas, Brazil.

ra032483@students.ic.unicamp.br, {jlopez, rdahab}@ic.unicamp.br
[2]Departamento de Matemática y Ciencia de la Computación,
Universidad de Santiago de Chile, Av. B. O'Higgins 3363, Santiago, Chile.

rodrigo.abarzua@usach.cl

Abstract. *Side-channel attacks are a growing threat to implementations of cryptographic systems. This article examines the state of the art of algorithmic countermeasures against simple side-channel attacks on elliptic curve cryptosystems defined over prime fields. We evaluate the security versus computation cost trade-offs of SSCA countermeasures for variable-base scalar multiplication algorithms without precomputation. The expected performance impact of each countermeasure is analyzed regarding their computational cost in terms of finite field operations.*

1. Introduction

Elliptic Curve Cryptography (ECC) is a class of public-key cryptosystems proposed by Neal Koblitz [Koblitz 1987] and Victor Miller [Miller 1985], which provides significant advantages in several situations, including implementations on specialized microprocessors. For example, some industry standards require 2048-bit integers [1] for the RSA system, whereas the equivalent security for ECC requires finite fields of just 160 bits. Given the restricted power consumption, storage and processing capacities of embedded microprocessors, ECC-based cryptosystems are an interesting option.

Passive side-channel attacks exploit physical leakages of a cryptographic process executing on a device, for example: timing [Kocher 1996], power consumption [Kocher et al. 1999] and electromagnetic radiation [Quisquater and Samyde 2001, Gandolfi et al. 2001]. These attacks present a realistic threat to cryptographic applications, and have been demonstrated to be very effective against smart cards without proper countermeasures [Mangard et al. 2007]. There are two general strategies for these attacks: *Simple Side-Channel Analysis* (SSCA) [Kocher 1996], which analyzes the measurements obtained during a single scalar multiplication, based on the differences in the measured quantity depending on the value of the secret key; and *Differential Side-channel Analysis* (DSCA) [Kocher et al. 1999], which is based on statistical techniques to retrieve information about the secret key based on measurements from several scalar multiplications.

[1] For the modulus n.

The aim of this paper is to show the landscape of solutions that implementers can choose to protect implementations of elliptic curve scalar multiplication algorithms against simple side-channel attacks, targeted at very restricted embedded devices. The SSCA countermeasures are evaluated from the security and computational cost (number of finite field operations) perspectives, providing a security versus computational cost comparison.

These tight device capabilities limited the scope of the paper to countermeasures to SSCA that require the minimum: additional data space at runtime, additional code space, time overhead and energy usage. Therefore, we have chosen countermeasures that do not make use of precomputation tables, usually to store elliptic curve points.

This paper is organized as follows. Section 2 provides an introduction to two kinds of simple side-channel analysis (SSCA): simple power analysis (SPA) and timing analysis (TA). Section 3 introduces the scalar multiplication problem and classic algorithms to solve it. Section 4 presents known variable-base scalar multiplication algorithms without precomputation and protected against SSCA, discussing their computational cost and known attacks. The performance comparison of the countermeasures is provided in Section 5. Finally, Section 6 concludes the paper.

2. Simple Side-channel Analysis (SSCA)

There are several types of side-channel analysis within the class of simple side-channel analysis, but two of them are commonly considered for software-based implementation of public-key cryptographic algorithms: simple power analysis (SPA) and timing analysis (TA).

2.1. Simple Power Analysis (SPA)

Power analysis in general, and simple power analysis in particular, exploit the fact that the instantaneous power consumption of a device depends on both: the data processed and the operation performed [Mangard et al. 2007, Kocher et al. 1999].

Power analysis countermeasures for both SPA and DPA are based on the reduction or elimination of the dependency between the power consumption of a cryptographic device and the intermediate values used by the algorithm, and are classified in two main groups: hiding and masking [Mangard et al. 2007].

The fundamental principle of *hiding* countermeasures is to remove the dependency of the data into the power consumption. In software implementations the goals are usually to randomize the algorithm control flow (time dimension) or the kind of instructions used on each run (amplitude dimension), without changing the input data or any other intermediate value processed by it, so that it is impossible to recognize the dependency between data and power consumption.

The concept of *masking* is to randomize the intermediate values processed by the cryptographic device, i.e., a masking operation is applied over these values before the original algorithm execution occurs. Sometime later, the resultant (masked) intermediate values are then unmasked. The goal of masking is to make the power consumption required to process the intermediate values on the masked implementation *independent* of the original (unmasked) values. To achieve this goal, masking countermeasures act on the amplitude dimension of the leakage, by means of data randomization.

2.2. Timing Analysis (TA)

Timing attacks against implementations of cryptographic algorithms exploit the fact that usually, in implementations, the elapsed time for the execution of an algorithm is variable and depends on the input data being processed on the particular run, be it fixed data (the key) or variable data (the plaintext).

In general, if an implementation is vulnerable to timing attacks it is also vulnerable to power attacks, but the converse is not necessarily true [Schindler 2002]. Timing analysis can often be combined with power analysis, to conceive powerful attacks. According to [Aciiçmez and Koç 2009], timing analysis can be classified in the following major groups: cache analysis, branch prediction analysis, and shared functional unit analysis.

3. Scalar Multiplication Algorithms

There are several classes of algorithms for multiplying a point P by an integer scalar k. In this paper we focus on variants of the double-and-add method. The double-and-add method is similar to square-and-multiplication method for modular exponentiation. The binary representation of k is denoted by $k_{n-1}2^{n-1} + \cdots + k_0 2^0$, then the scalar multiplication $[k]P = (k_{n-1}2^{n-1} + \cdots + k_0 2^0)P$ is computed using Horner's rule, resulting in the double-and-add method, $[k]P = [k_0 + 2(k_1 + 2(\ldots(k_{n-2} + 2k_{n-1})\ldots))]P$, which requires $n = \lfloor \log_2(k) \rfloor + 1$ point doublings and, in average, $\frac{n}{2}$ point additions $\left(nD + \frac{n}{2}A\right)$ [2]. The double-and-add method is optimal [Cohen et al. 2010].

4. Countermeasures for Variable-Base Scalar Multiplication without Precomputation against Simple Side-Channel Attacks

Elliptic curve scalar multiplication is particularly vulnerable to simple side-channel analysis because the operations of doubling and addition of points are intrinsically different. Very efficient countermeasures are known but they are only applicable to specific models of elliptic curves (e.g. Edwards curves [Bernstein et al. 2008, Hisil et al. 2008]). Although it is possible to select an elliptic curve from a model where efficient countermeasures are known, in practice, it is very likely that curves established by a standard will be selected. For example, NIST [NIST 2000] and SEC 2 [Certicom 2010] standards.

The most commonly used algorithm for computing $Q = [k]P$ on an elliptic curve is the *double-and-add* algorithm, in the *left-to-right* or *right-to-left* versions [3]. Suppose that the doubling of a point and the addition of two different points are implemented with different formulas. Then, these two operations can be distinguished by SSCA [Kocher 1996, Kocher et al. 1999]. When the power trace shows a point doubling followed by a point addition, the current bit, say k_i, is equal to 1; and when the power trace shows a doubling followed by another doubling, then $k_i = 0$. The usual approach to prevent SSCA consists in always repeating the same pattern of instructions, whatever the processed data is.

Several proposals have been made to protect scalar multiplication against these attacks. For example, the *double-and-add-always* algorithm of Coron [Coron 1999] ensures

[2] In this notation, D stands for point doubling and A stands for point addition.

[3] In the left-to-right version, the scalar bits are scanned from the most (MSB) to the least (LSB) significant bit. In the right-to-left version, the order is reversed.

the sequence of operations to compute a scalar multiplication is independent of the value of the secret scalar by inserting a dummy point addition between consecutive doublings (i.e., when the bit of the scalar is 0).

A second countermeasure is to use *unified formulas* which use similar sets of field operations for both additions and doubling operations. These formulas exist for Weierstrass curves [Brier and Joye 2002], and special curves, such as Edwards [Edwards 2007] and inverted Edwards [Bernstein and Lange 2007] curves, among others [4].

Another countermeasure is the *Montgomery ladder* [Montgomery 1987], a technique designed for a special type of curve in large characteristic fields. It makes sure that every bit of the scalar corresponds to both a doubling and an addition, and that both operations have an impact on the output of the scalar multiplication. The addition formula for curves on the Montgomery model is also much simpler than that for curves on the Weierstrass model, contributing to make the scalar multiplication faster in this curve model. However, it is not always possible to convert a curve in the Weierstrass model to one in the Montgomery model, because, among other reasons, the number of points in a Montgomery curve is always divisible by 4. Nevertheless, the converse is true [Cohen et al. 2010].

Elliptic curve cryptography standards recommend curves on the Weierstrass form over $\mathbb{F}_p$ (prime fields) or $\mathbb{F}_{2^m}$ (binary extension fields), where $p > 2$ is a prime number and m is an integer. None of the NIST recommended elliptic curves [NIST 2000] over prime fields can be converted to the Montgomery form, because all of them have a prime number of points (the cofactor is always 1).

A fourth approach consists in using *regular* representations of the scalar [Thériault 2006, Joye 2007], with the same fixed sequence of group operations for all scalars. Yet another countermeasure [Goundar et al. 2011] is the use of signed-digit representations of the scalar, particularly the zero-less signed-digit (ZSD) representation [5].

Finally, side-channel atomicity [Chevallier-Mames et al. 2004]) splits point operations into small homogeneous blocks of basic field operations, making it hard to distinguish between atomic blocks of point doublings from those of point additions.

The following countermeasures are considered in this paper: *a*) Unified Formulas of Brier-Joye and Brier-Dechene-Joye; *b*) Double-and-add-always of Coron; *c*) Montgomery Ladder over prime fields of Brier-Joye and Izu-Takagi; *d*) Double-add of Joye; *e*) Zero-less signed-digit (ZSD) of Goundar; and *f*) Atomic Blocks of Chevallier-Mames, Longa-Miri and Abarzúa-Thériault.

In the following subsections we present these countermeasures in detail, analyzing their side-channel security and the expected performance based on the number of required finite field operations.

[4] More details can be found in the database of special elliptic curves [Tanja and Bernstein 2014]. Such special families of elliptic curves are not studied in this work.

[5] An odd integer k is represented in ZSD if its digits are in the set $\{-1, 1\}$.

4.1. Unified Formulas of Brier-Joye [Brier and Joye 2002]

Unified Formulas for point addition and point doubling using projective coordinates were presented by Brier and Joye [Brier and Joye 2002]. Let $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, with $x_i = X_i/Z_i$ and $y_i = Y_i/Z_i$, then $R = P + Q = (X_3, Y_3, Z_3)$ is given by:

$$\begin{array}{lllll} U_1 = X_1Z_2, & U_2 = X_2Z_1, & S_1 = Y_1Z_2, & S_2 = Y_2Z_1, & T = U_1 + U_2, \\ M = S_1 + S_2, & Z = Z_1Z_2, & F = ZM, & R = T^2 - U_1U_2 + aZ^2, & L = MF, \\ G = TL, & W = R^2 - G, & X_3 = 2FW, & Y_3 = R(G - 2W) - L^2, & Z_3 = 2F^3. \end{array}$$

This formula requires $13M + 5S$ [6]. The Unified Formulas of Brier-Joye are prone to the following attacks.

4.1.1. Izu-Takagi Attack [Izu and Takagi 2002b]

The unified formulas of Brier-Joye are only valid if $y_1 + y_2 \neq 0$, where $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Izu and Takagi [Izu and Takagi 2002b] presented an attack using two points such that $x_1 \neq x_2$ and $y_1 + y_2 = 0$. The main idea of the attack is to use an exceptional point, which causes an exceptional condition ($0^{-1} \notin \mathbb{F}_p$, i.e., a division by zero) on the underlying unified formula in affine coordinates. The secret scalar k is guessed from the error of the scalar multiplication $[k]P$ for different points P. If an attacker wants to know if the target calculated $[m]P + P$ with $2 \leq m < k$, he can use a point P such that $y(mP) + y(P) = 0$. If the device replies an error to the attacker, or does it in an implicit but detectable manner, then he knows the device calculated $[m]P + P$. Starting with $m = 2$ [7], and by following this process, the attacker is able to recover the secret scalar bit-by-bit, from the most to the least significant.

Brier, Dechene and Joye [Brier et al. 2004] presented a new unified formula to protect against Izu-Takagi attack. Nevertheless, their formula is prone to both Amiel's attack [Amiel et al. 2009] and PACA attack [Amiel et al. 2007].

4.1.2. Walter's Attack [Walter 2004]

Walter's attack [Walter 2004] is based on the fact that the conditional subtraction in a Montgomery modular multiplication (MMM) operation can be detected. Given a point $P = (X, Y, Z)$, in the point doubling using the projective formulas of Brier-Joye the computation of U_1 and U_2 are identical ($U_1 = U_2 = XZ$), and they exhibit identical behavior with respect to the occurrence of the final conditional subtraction in MMM. The same property holds for the computation of S_1 and S_2.

The behavior for point addition is different. Point addition involves the input point $P = (X_1, Y_1, Z_1)$, where the random (or randomized) coordinates mean that occasionally

[6] In expressions regarding computational costs in terms of the number of finite field operations, M stands for (field) multiplication, S for squaring and I for inversion.

[7] When $m = 2$ and the attacker knows whether $y(2P) + y(P) = 0$, then, if it is, $k_{n-2} = 1$; otherwise, $k_{n-2} = 0$

X_1 and Y_1 will both be small (i.e., close to 0) and Z_1 will be large. This means that the computations of U_1 and S_1 are less likely to include the additional subtraction in MMM, while the computations of U_2 and S_2 are more likely to include the additional subtraction. This difference in behavior can be detected by an attacker and can be used accordingly to recover the bits of the secret scalar.

4.1.3. Amiel *et al*'s Attacks [Amiel et al. 2009]

A common requirement for several countermeasures against simple side-channel attacks is that multiplication and squaring field operations are indistinguishable from the side-channel analysis point of view, i.e., both squaring and multiplication must be computed using the same algorithm. Particularly, atomic blocks [Chevallier-Mames et al. 2004, Chen et al. 2009, Giraud and Verneuil 2010] and *Unified formulas* [Brier and Joye 2002, Bernstein and Lange 2007, Joye et al. 2010] countermeasures rely on this property. However, that's not usually the case. Amiel's attack [Amiel et al. 2009] is based on distinguishing between multiplications and squarings using the instantaneous power consumption trace. This is possible because the Hamming weight probability distribution of the result of a multiplication is distinct from that of a squaring operation, and they can be distinguished in the power traces.

Notice that the computation of multiplications $Z = Z_1Z_2$ and U_1U_2 in $R = T^2 - U_1U_2 + aZ^2$, when a point addition is performed, are different from those when a point doubling operation is performed. The computation of $Z = Z_1^2$ and U_1^2 in R will be squaring operations and, hence, Amiel's attack can be applied to such implementations.

4.1.4. Passive and Active Combined Attack (PACA) [Amiel et al. 2007]

Amiel *et al.* [Amiel et al. 2007] presented a Passive and Active Combined Attack (PACA) on a (supposedly) side channel resistant implementation of the square and multiply algorithm. Although not a pure SSCA attack, because of the required fault insertion step (the active part of the attack), we present it here for completeness.

The main idea of the PACA attack is as follows. An attacker applies a fault in the register storing the Z coordinate of point P_1, say setting $Z_1 = 0$ after the fault. Then, in the Unified formulas of Brier-Joye, we have two different patterns for the calculation, $Z = Z_1 \cdot Z_1 = 0 \cdot 0$ (if it is a doubling) and $Z = Z_1 \cdot Z_2 = 0 \cdot Z_2$ (with $Z_2 \neq 0$, if it is an addition), and both can be identified in a power trace [Amiel et al. 2007, Schmidt et al. 2010]. This allows an SSCA attacker to distinguish between point doubling and addition operations and consequently to recover the secret scalar.

4.2. Double-and-add-always algorithm of Coron [Coron 1999]

The *double-and-add-always* algorithm of Coron [Coron 1999] (Algorithm 1) uses a dummy point addition when the scalar bit k_i is 0, such that the sequence of operations to compute the scalar multiplication is independent of the value of the secret scalar.

Therefore an adversary cannot guess the bit k_i by SPA. A drawback of this method is its low efficiency. It requires $nA + nD$ field operations, a 33% increase in the amount of field operations in comparison to the (unprotected) binary left-to-right algorithm.

Algorithm 1 *Double-and-add always* algorithm resistant against SPA

```
INPUTS: Point P $\in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
$R_0 \leftarrow P_\infty$
for $i$ from $n-1$ to $0$ do
    $R_0 \leftarrow 2R_0$
    $R_1 \leftarrow R_0 + P$
    $R_0 \leftarrow R_{k_i}$
end for
return $R_0$
```

The *Double-and-add always* algorithm of Coron is prone to the following attacks.

4.2.1. Fouque and Valette's Doubling Attack [Fouque and Valette 2003]

The doubling attack of Fouque-Valette [Fouque and Valette 2003] is based on the fact that it is possible to detect if two intermediate values are equal when the algorithm computes the scalar multiplication for points chosen points P and $2P$. Several algorithms protected against SPA are vulnerable to Fouque and Valette's attack, such as the classic binary left-to-right algorithm, including those derived from it, such as Coron's double-and-add-always algorithm.

In Coron's double-and-add-always algorithm (Algorithm 1), the partial sums are computed as follows: $S_m(P) = \sum_{i=1}^{m} k_{n-i} 2^{m-i} P = \sum_{i=1}^{m-1} k_{n-i} 2^{m-1-i}(2P) + k_{n-m}P = S_{m-1}(2P) + k_{n-m}P$. So, the intermediate result of the algorithm at step m when given input P will be equal to the intermediate result at step $m-1$ when given input $2P$, if and only if, $k_{n-m} = 0$. Therefore, an attacker can obtain the secret scalar by comparing the doubling computation at step $m+1$ for P and at step m for $2P$ to recover the bit k_{n-m}. If both computations are identical, $k_{n-m} = 0$, otherwise $k_{n-m} = 1$. It has been shown that with only two scalar multiplication requests chosen by the attacker, it is possible to recover all the bits of the scalar [8].

4.3. Montgomery Ladder of Brier-Joye [Brier and Joye 2002]

Another possible countermeasure is the *Montgomery ladder* method [Montgomery 1987], originally designed for a special type of curve, the so-called Montgomery curve in large characteristic. Brier and Joye [Brier and Joye 2002] extended this method to Weierstrass curves of large characteristic. Their algorithm requires $9M + 2S$ for point addition and $6M+3S$ for point doubling. The classic Montgomery powering ladder is prone to M safe-error fault attacks [9] [Sung-Ming et al. 2002], Joye and Yen [Joye and Yen 2003] proposed modifications in order to counteract them (Algorithm 2).

The modified Montgomery ladder makes it impossible to insert safe faults, thus providing a natural protection against SPA, M safe-error and C safe-error attacks [10] [Joye and Yen 2003]. An important observation is that this algorithm al-

[8] The attacker collects one power trace for the computation of kP and one for the computation of $k(2P)$. For each iteration $m = 1, ..., n$, he runs the attack as described and finds k_{n-m}.

[9] A M safe-error is a memory safe-error in which bits of a register are maliciously modified, and the change is temporary, i.e. the register can be overwritten later.

[10] A C safe-error stands for computational safe-error, and consists in timely induce a temporary fault in the ALU for determining whether an operation is dummy or effective.

lows one to compute scalar multiplication on elliptic curves using the x-coordinate only [Brier and Joye 2002, Fischer and Giraud 2002, Izu and Takagi 2002a]. In Table 1 we present the cost of this countermeasure, including its refined forms.

Algorithm 2 Montgomery ladder resistant against SPA and safe fault attacks

```
INPUTS: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
$R_0 \leftarrow P_\infty, R_1 \leftarrow P$
for $i$ from $n-1$ to $0$ do
    $b \leftarrow k_i,$
    $R_{1-b} \leftarrow R_{1-b} + R_b$
    $R_b \leftarrow 2R_b$
end for
return $R_0$
```

Table 1. Computing cost for algorithms based on Montgomery ladder

Algorithm	In	# regs.	Total cost
Classic Montgomery ladder of Brier-Joye	[Brier and Joye 2002]	8	$n(12M + 13S) + 1I + 3M + 1S$
$X-$only Montgomery ladder	[Brier and Joye 2002, Izu and Takagi 2002a]	7	$n(9M + 7S) + 1I + 14M + 3S$
$(X, Y)-$only co$-Z$ Montgomery ladder	Alg. 15 in [Goundar et al. 2011]	6	$n(8M + 6S) + 1I + 1M$

The Montgomery Ladder of Brier-Joye is prone to the following attacks.

4.3.1. Relative Doubling Attack of Yen *et al* [Yen et al. 2006]

Yen *et al.* [Yen et al. 2006] proposed the relative doubling attack, which uses the same chosen input (P and $2P$) as described in Fouque and Valettes's doubling attack (Section 4.2.1). In this attack it is just required to determine the relation between two adjacent secret scalar bits (i.e., if $k_i = k_{i-1} = 0$ or $k_i = k_{i-1} = 1$ holds), thereby decreasing the number of key candidates.

4.4. Double-add algorithm of Joye [Joye 2007]

Joye's double-add algorithm [Joye 2007] (Algorithm 3), like Montgomery ladder for right-to-left scalar multiplications, always repeats the same pattern of effective operations. Table 2 shows the cost for the classic Joye's double-add algorithm and the variant using the Co-Z technique [Goundar et al. 2011]. There is no known SSCA attack against this algorithm.

Algorithm 3 Joye's double-add resistant against SPA

```
INPUTS: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
$R_0 \leftarrow P_\infty, R_1 \leftarrow P$
for $i$ from $0$ to $n-1$ do
    $b \leftarrow k_i,$
    $R_{1-b} \leftarrow 2R_{1-b} + R_b$
end for
return $R_0$
```

Table 2. Computing cost of Joye's double-add

Algorithm	In	# regs.	Total cost
Classic Joye's double-add	Alg. 5 in [Joye 2007]	10	$n(13M + 8S) + 1I + 3M + 1S$
Co-Z Joye's double-add	Alg. 14 in [Goundar et al. 2011]	8	$n(9M + 7S) + 1I - 9M - 6S$

4.5. Signed Digit Methods of Goundar *et al.* [Goundar et al. 2011]

In order to prevent SPA-type attacks, Goundar *et al* [Goundar et al. 2011] proposed the use of the *zeroless signed-digit expansion* (ZSD) in the binary left-to-right or right-to-left algorithms. The odd scalar k, is recoded with digits in the set $\{-1, 1\}$. The recoding is on-the-fly, taking place inside the main loop (Algorithm 4).

The computational costs for the most efficient algorithms using the signed digit method are shown in Table 3: the right-to-left signed-digit algorithm using the Co-Z technique and the corresponding left-to-right algorithm. The latter also applies the (X,Y)-only technique, which simplifies some of the curve operations, rendering the Z coordinate (in projective coordinates) unnecessary and thus improving the cost.

Algorithm 4 Classic Signed-digit method: Left-to-right

```
INPUTS: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$ with $k_0 = 1$
OUTPUTS: $\mathbf{Q} = [k] \cdot \mathbf{P}$
$R_0 \leftarrow P; R_1 \leftarrow P$
for $i$ from $n - 1$ to $1$ do
    $\kappa \leftarrow (-1)^{1+k_i}$
    $R_0 \leftarrow 2R_0 + (\kappa)R_1$
end for
return $R_0$
```

Table 3. Computing cost of Signed-digit method

Algorithm	In	# regs.	Total cost
Right-to-left algorithm			
Co$-Z$ signed-digit algorithm	Alg. 17 in [Goundar et al. 2011]	8	$n(9M + 7S) + 1I - 9M - 6S$
Left-to-right algorithm			
$(X, Y)-$only co$-Z$ signed-digit algorithm	Alg. 16 in [Goundar et al. 2011]	6	$n(8M + 6S) + 1I - 5M - 4S$

4.6. Atomic Blocks of Chevallier-Mames *et al.* [Chevallier-Mames et al. 2004]

Atomic blocks [Chevallier-Mames et al. 2004] is a method to secure scalar multiplication against SSCA consisting in partitioning point operations into small homogeneous atomic blocks, which cannot be distinguished from each other through SSCA. The original atomic block of Chevallier-Mames has a (M, A, N, A) [11] structure of field operations.

One important assumption was made in the atomic blocks of Chevallier-Mames: multiplication and squaring are indistinguishable from a side-channel perspective. This was later proved wrong by [Amiel et al. 2009] (see Section 4.1.3) and [Hanley et al. 2011].

Longa and Miri [Longa and Miri 2008] presented a new atomic block structure based on the sequence (S, N, A, M, N, A, A) [12] of field operations. Their atomic block

[11] Multiplication-Addition-Negation-Addition.

[12] Squaring-Negation-Addition-Multiplication-Negation-Addition-Addition.

Table 4. Costs of scalar multiplication using the atomic blocks of Abarzúa and Thériault [Abarzúa and Thériault 2012]

Algorithm	In	# regs.	Total cost
Right-to-left algorithm			
Modified Doubling and General Addition	[Abarzúa and Thériault 2012]	16	$n(8.5M + 8.5S) + 1I + 3M + 1S$
Left-to-right algorithm			
Doubling and Mixed Addition	[Abarzúa and Thériault 2012]	11	$n(7M + 7S) + 1I + 3M + 1S$

structure have been applied to doubling, tripling and mixed addition for elliptic curves in Jacobian coordinates.

Abarzúa and Thériault [Abarzúa and Thériault 2012] built new sets of atomic blocks designed to protect against both SSCA and C-safe fault attacks. These atomic blocks are structured with the sequence of field operations (S, N, A, A, M, A). They applied these atomic blocks to various operations in Jacobian coordinates: doubling, tripling, and quintupling, as well as mixed Jacobian-affine addition. Formulas were also given to the general Jacobian addition and Modified-Jacobian doubling for use in right-to-left scalar multiplication. Finally, they presented a variation of the Jacobian doubling formula that requires the same number of blocks as the mixed Jacobian-affine addition, essentially giving the atomic equivalent of unified formulas.

Table 4 summarizes the scalar multiplication costs using these atomic blocks in the right-to-left and left-to-right algorithms.

5. Computational cost versus security comparison

For the computational cost comparison between different algorithms, we consider the following cost ratio for the finite field operations: $S/M = 0.8$ and $I/M = 100$. We also consider that the scalar k is $n = 192$ bits in length. Tables 5 and 6 summarizes the expected (theoretical) computational cost and the security issues of the different countermeasures for scalar multiplication algorithms against SSCA.

Among the right-to-left algorithms, Joye's double-add and Goundar's signed-digit algorithm are tied as the most efficient. Abarzúa and Thériault's atomic blocks is the most efficient left-to-right (and overall, in fact) scalar multiplication algorithm protected against SSCA, and there is not any known attack against it.

Table 5. Comparison of protected *left-to-right* scalar multiplication algorithms

Countermeasure	**Coordinate Systems**	**Total Cost**	**Performance** $n = 192$	**Security Problem**
Unified Formulas for Weierstrass curves	$\mathcal{P}$	$n(13M + 5S) + 1I + 2M$	$3366M$	(ψ)
		$n(16M + 3S) + 1I + 2M$	$3634.8M$	(ϕ)
Double-and-Add-Always	$\mathcal{J}$	$n(10M + 9S) + 1I + 3M + 1S^{(\mathbf{a})}$	$3406.2M$	(φ)
Montgomery Ladder for Weierstrass curves	$\mathcal{J}$	$n(8M + 6S) + 1I + 1M^{(\mathbf{b})}$	$2558.6M$	-
		$n(9M + 7S) + 1I + 14M + 3S^{(\mathbf{c})}$	$2919.6M$	-
Signed-digit	$\mathcal{J}$	$n(8M + 6S) + 1I - 5M - 4S^{(\mathbf{d})}$	$2549.4M$	-
Atomic Blocks	$\mathcal{J}$	$n(7M + 7S) + 1I + 3M + 1S^{(\mathbf{e})}$	$2523M$	(ξ)

Detailed description of computational cost:

(a) [Abarzúa and Thériault 2012]: fast mixed addition $(7M + 4S)$ and fast doubling $(3M + 5S)$ with $a = -3$.
(b) [Goundar et al. 2011]: (X, Y)-only co-Z Montgomery ladder, $(8M + 6S)$ for each bit.
(c) [Brier and Joye 2002, Izu and Takagi 2002a]: X-only Montgomery ladder, $(9M + 7S)$ for each bit.
(d) [Goundar et al. 2011]: $(X, Y)-$only co-Z signed-digit algorithm, $(8M + 6S)$ for each bit.
(e) [Abarzúa and Thériault 2012]: addition $(6M + 6S)$ and doubling $(4M + 4S)$ [13].

Attacks:

(ψ) [Izu and Takagi 2002b, Walter 2004, Amiel et al. 2009, Schmidt et al. 2010].
(ϕ) [Stebila and Thériault 2006, Amiel et al. 2009, Amiel et al. 2007].
(φ) [Fouque and Valette 2003]: doubling attack.
(ξ) [Fouque and Valette 2003, Chen et al. 2009] [14]: these attacks do not apply to Abarzúa-Thériault atomic blocks.

Table 6. Comparison of protected right-to-left scalar multiplication algorithms

Countermeasure	Coordinate Systems	Total Cost	Performance $n = 192$	Security Problem
Joye's double-add	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{f})}$	$2889.4M$	-
Signed-digit	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{g})}$	$2889.4M$	-
Atomic Blocks	$\mathcal{J}$	$n(8.5M + 8.5S) + 1I + 3M + 1S^{(\mathbf{h})}$	$3041.4M$	(χ)

Detailed description of computational cost:

(f) [Goundar et al. 2011]: Co-Z Joye's double-add, $(9M + 7S)$ for each bit.
(g) [Goundar et al. 2011]: Co-Z signed-digit algorithm, $(9M + 7S)$ for each bit.
(h) [Abarzúa and Thériault 2012]: general addition $(9M+9S)$ and doubling $(4M+4S)$.

Attacks:

(χ) Chen's attack [Chen et al. 2009] [15]. This attack does not apply to Abarzúa-Thériault atomic blocks.

6. Conclusion and future work

Side-channel attacks are a growing threat to implementations of cryptographic systems. This article examined the state of the art of algorithmic countermeasures for variable-base scalar multiplication algorithms without precomputation. A comparison has been made between several classes of proposed countermeasures regarding their computational costs, claimed security properties and known attacks.

[13] In this case the algorithm performs $nD + \frac{n}{2}A$.

[14] These attacks works because the implementation does not avoid irregular breaks between atomic blocks within the same group operation and distinct group operations.

[15] This experimental attack applies because the implementation does not avoid irregular breaks between atomic blocks within the same group operation and distinct group operations.

It is known that theoretical computational costs based on finite field operation counts does not tell which algorithm is in fact the most efficient in practice. Implementations on the target platform are required to have a true picture of their real performance. This is even more true when side-channel protected implementations are required, because implementation-level protections (e.g., against timing analysis) for one algorithm may be more costly than those for another algorithm, as they may be dependent on the structure of the algorithm.

Besides the performance results from a real implementation, a security assessment of the selected algorithms is also required.

References

Abarzúa, R. and Thériault, N. (2012). Complete Atomic Blocks for Elliptic Curves in Jacobian Coordinates over Prime Fields. In Hevia, A. and Neven, G., editors, *Progress in Cryptology – LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 37–55. Springer Berlin / Heidelberg.

Aciiçmez, O. and Koç, c. K. (2009). Microarchitectural Attacks and Countermeasures. In *Cryptographic Engineering*, chapter 18. Springer.

Amiel, F., Feix, B., Tunstall, M., Whelan, C., and Marnane, W. P. (2009). Distinguishing multiplications from squaring operations. In *Selected Areas in Cryptography*, pages 346–360. Springer.

Amiel, F., Villegas, K., Feix, B., and Marcel, L. (2007). Passive and active combined attacks: Combining fault attacks and side channel analysis. In *Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on*, pages 92–102.

Bernstein, D. and Lange, T. (2007). Inverted edwards coordinates. In Boztaş, S. and Lu, H.-F., editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 4851 of *LNCS*, pages 20–27. Springer.

Bernstein, D. J., Birkner, P., Joye, M., Lange, T., and Peters, C. (2008). Twisted edwards curves. In *Progress in Cryptology–AFRICACRYPT 2008*, pages 389–405. Springer.

Brier, E., Dechene, I., and Joye, M. (2004). Unified Point Addition Formulae for Elliptic Curve Cryptosystems. In *Embedded Cryptographic Hardware: Methodologies and Archi- tectures*, pages 247–256. Nova Science Publishers.

Brier, E. and Joye, M. (2002). Weierstrass Elliptic Curves and Side-Channel Attacks. In *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems, vol 2274*, pages 335–345. Springer.

Certicom (2010). SEC 2: Recommended Elliptic Curve Domain Parameters, version 2.0. Technical report, Certicom Corp.

Chen, T., Li, H., Wu, K., and Yu, F. (2009). Countermeasure of ECC against Side-channel Attacks: Balanced Point Addition and Point Doubling Operation Procedure. *Asia-Pacific Conference on Information Processing*.

Chevallier-Mames, B., Ciet, M., and Joye, M. (2004). Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity. *Computers, IEEE Transactions on*, 53(6):760–768.

Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., and Vercauteren, F. (2010). *Handbook of elliptic and hyperelliptic curve cryptography*. Chapman and Hall/CRC.

Coron, J.-S. (1999). Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems*, pages 292–302. Springer.

Edwards, H. (2007). A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44(3):393–422.

Fischer, W. and Giraud, C. (2002). Parallel scalar multiplication on general elliptic curves over Fp hedged against Non-Differential Side-Channel Attacks. *IACR Cryptology ePrint Archive*.

Fouque, P.-A. and Valette, F. (2003). The doubling attack – why upwards is better than downwards. In Walter, C., Koç, e., and Paar, C., editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 269–280. Springer.

Gandolfi, K., Mourtel, C., and Olivier, F. (2001). Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES 2001*, pages 251–261. Springer.

Giraud, C. and Verneuil, V. (2010). Atomicity improvement for elliptic curve scalar multiplication. In Gollmann, D., Lanet, J.-L., and Iguchi-Cartigny, J., editors, *Smart Card Research and Advanced Application*, volume 6035 of *LNCS*, pages 80–101. Springer.

Goundar, R., Joye, M., Miyaji, A., Rivain, M., and Venelli, A. (2011). Scalar multiplication on Weierstraßelliptic curves from Co-Z arithmetic. *Journal of Cryptographic Engineering*, 1(2):161–176.

Hanley, N., Tunstall, M., and Marnane, W. (2011). Using templates to distinguish multiplications from squaring operations. *International Journal of Information Security*, 10(4):255–266.

Hisil, H., Wong, K. K.-H., Carter, G., and Dawson, E. (2008). Twisted Edwards curves revisited. In *Advances in Cryptology-ASIACRYPT 2008*, pages 326–343. Springer.

Izu, T. and Takagi, T. (2002a). A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks. In *Public Key Cryptography – PKC 2002*, pages 280–296.

Izu, T. and Takagi, T. (2002b). Exceptional procedure attack on elliptic curve cryptosystems. In *Public Key Cryptography—PKC 2003*, pages 224–239. Springer.

Joye, M. (2007). Highly regular right-to-left algorithms for scalar multiplication. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 135–147. Springer.

Joye, M., Tibouchi, M., and Vergnaud, D. (2010). Huff's model for elliptic curves. In *Algorithmic Number Theory*, pages 234–250. Springer.

Joye, M. and Yen, S.-M. (2003). The Montgomery powering ladder. In *Cryptographic Hardware and Embedded Systems-CHES 2002*, pages 291–302. Springer.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209.

Kocher, P. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Koblitz, N., editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer Berlin / Heidelberg.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential Power Analysis. In Wiener, M., editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, page 789. Springer Berlin / Heidelberg.

Longa, P. and Miri, A. (2008). Fast and Flexible Elliptic Curve Point Arithmetic over Prime Fields. In *Computers, IEEE Transactions on*, volume 57, pages 289–302.

Mangard, S., Oswald, E., and Popp, T. (2007). *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer.

Miller, V. S. (1985). Use of Elliptic Curves in Cryptography. In *Advances in Cryptology -CRYPTO Proceedings*, pages 417–426. Springer.

Montgomery, P. L. (1987). Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264.

NIST (2000). FIPS 186-2: Digital Signature Standard. Technical report, NIST.

Quisquater, J.-J. and Samyde, D. (2001). Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*, pages 200–210. Springer.

Schindler, W. (2002). A Combined Timing and Power Attack. In Naccache, D. and Paillier, P., editors, *Public Key Cryptography SE - 19*, volume 2274 of *LNCS*, pages 263–279. Springer.

Schmidt, J.-M., Tunstall, M., Avanzi, R., Kizhvatov, I., Kasper, T., and Oswald, D. (2010). Combined implementation attack resistant exponentiation. In *Progress in Cryptology–LATINCRYPT 2010*, pages 305–322. Springer.

Stebila, D. and Thériault, N. (2006). Unified point addition formulæ and side-channel attacks. In Goubin, L. and Matsui, M., editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 354–368. Springer.

Sung-Ming, Y., Kim, S., Lim, S., and Moon, S. (2002). A countermeasure against one physical cryptanalysis may benefit another attack. In Kim, K., editor, *Information Security and Cryptology — ICISC 2001*, volume 2288 of *LNCS*, pages 414–427. Springer.

Tanja, L. and Bernstein, D. J. (2014). Explicit-Formulas Database. www.hyperelliptic.org/EFD/bib.html.

Thériault, N. (2006). Spa resistant left-to-right integer recodings. In Preneel, B. and Tavares, S., editors, *Selected Areas in Cryptography*, volume 3897 of *LNCS*, pages 345–358. Springer.

Walter, C. (2004). Simple power analysis of unified code for ecc double and add. In Joye, M. and Quisquater, J.-J., editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 191–204. Springer.

Yen, S.-M., Ko, L.-C., Moon, S., and Ha, J. (2006). Relative doubling attack against montgomery ladder. In Won, D. and Kim, S., editors, *Information Security and Cryptology - ICISC 2005*, volume 3935 of *LNCS*, pages 117–128. Springer.