Revisiting Atomic Patterns for Scalar Multiplications on Elliptic Curves

Franck Rondepierre

Oberthur Technologies Crypto Group

Cardis 2013

Outline

1. Introduction
2. Elliptic Curve Background
3. Side-Channel Analysis
   - Simple Power Attack
   - State-of-the-Art Countermeasures
4. Our Contribution
5. Conclusion

Discrete Log Problem (DLP)

Given two elements G, P of a cyclic group, find a scalar k such that:

$$\underbrace{G + \cdots + G}_{k \text{ times}} = [k]G = P$$

- DLP is a assumed to be a hard problem
- Elliptic Curve Cryptography (ECC) is based on this problem
- ECC implementations must not reveal secret scalars

Outline

Outline

1 Introduction

2 Elliptic Curve Background

3 Side-Channel Analysis
   - Simple Power Attack
   - State-of-the-Art Countermeasures

4 Our Contribution

5 Conclusion

Elliptic Curve Definition

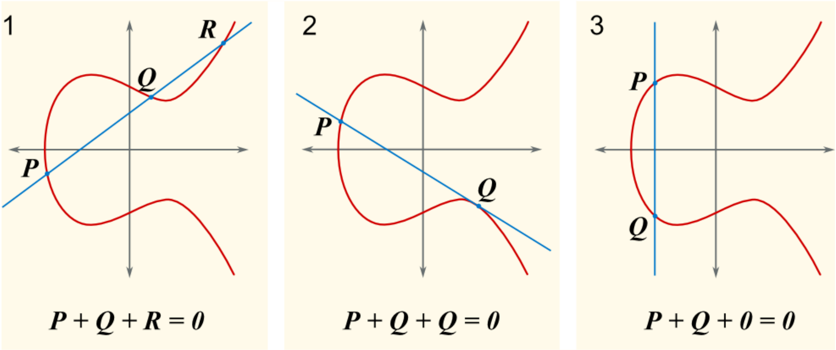
Short Weïerstrass equation

$p > 3, \quad \{a, b\} \subset \mathbb{F}_p, \quad 4\,a^3 + 27\,b^2 \neq 0.$

$$(\mathcal{E}) : y^2 = x^3 + a\,x + b\,.$$

$(\mathcal{E}\,(\mathbb{F}_p) \cup \mathcal{O}, +)$ abelian group.

Elliptic Curve Group Law



Scalar Multiplication

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

Scalar Multiplication

Right-to-Left Evaluation ◀

$$[k_0]P$$

Left-to-Right Evaluation ▶

Right-to-Left Evaluation ◀

$$[k_0]P + [k_1]2P$$

Left-to-Right Evaluation ▶

Right-to-Left Evaluation ◀

$$[k]P = [k_0]P + [k_1]2P + \ldots + [k_{\ell-1}]2^{\ell-1}P$$

Left-to-Right Evaluation ▶

Scalar Multiplication

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

$$[k_{\ell-1}]P$$

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

$$2(\,[k_{\ell-1}]P) + [k_{\ell-2}]P$$

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

$$[k]P = 2(\ldots 2(2(\,[k_{\ell-1}]P) + [k_{\ell-2}]P) + \ldots) + [k_0]P$$

Scalar Multiplication

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

- A doubling is performed for every scanned bit

Scalar Multiplication

Right-to-Left Evaluation ◀

Left-to-Right Evaluation ▶

- A doubling is performed for every scanned bit
- An addition is performed only for non-zero bit

Improved Techniques

- Pre/Post-computations
- RAM consumption
- Reduce the number of doublings and additions

Improved Techniques

Window Techniques ◀▶

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad \color{red}{k_1} \quad \color{red}{k_0}$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀ ▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀ ▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Window Techniques ◀▶

$$k: \quad k_{\ell-1} \quad \ldots \quad k_5 \quad k_4 \quad k_3 \quad k_2 \quad k_1 \quad k_0$$

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Improved Techniques

Window Techniques ◀▶

Straus-Shamir Trick ▶

$$\text{k:}\quad \begin{array}{ccccc} k_{\ell-1} & k_{\ell-2} & k_{\ell-3} & \dots & k_{\ell/2} \\ k_{\ell/2-1} & k_{\ell/2-2} & k_{\ell/2-3} & \dots & k_0 \end{array}$$

Sparse Representations ◀▶

Improved Techniques

Window Techniques ◀▶

Straus-Shamir Trick ▶

$$
\text{k:}\quad
\begin{array}{ccccc}
\color{red}{k_{\ell-1}} & k_{\ell-2} & k_{\ell-3} & \dots & k_{\ell/2} \\
\color{red}{k_{\ell/2-1}} & k_{\ell/2-2} & k_{\ell/2-3} & \dots & k_0
\end{array}
$$

Sparse Representations ◀▶

Window Techniques ◀▶

Straus-Shamir Trick ▶

$$\text{k:}\quad \begin{array}{ccccc} k_{\ell-1} & k_{\ell-2} & k_{\ell-3} & \dots & k_{\ell/2} \\ k_{\ell/2-1} & k_{\ell/2-2} & k_{\ell/2-3} & \dots & k_0 \end{array}$$

Sparse Representations ◀▶

Window Techniques ◀▶

Straus-Shamir Trick ▶

$$\text{k:}\quad \begin{matrix} k_{\ell-1} & k_{\ell-2} & \color{red}{k_{\ell-3}} & \dots & k_{\ell/2} \\ k_{\ell/2-1} & k_{\ell/2-2} & \color{red}{k_{\ell/2-3}} & \dots & k_0 \end{matrix}$$

Sparse Representations ◀▶

Improved Techniques

Window Techniques ◀▶

Straus-Shamir Trick ▶

$$\text{k:}\quad \begin{array}{ccccc} k_{\ell-1} & k_{\ell-2} & k_{\ell-3} & \ldots & \color{red}{k_{\ell/2}} \\ k_{\ell/2-1} & k_{\ell/2-2} & k_{\ell/2-3} & \ldots & \color{red}{k_0} \end{array}$$

Sparse Representations ◀▶

Window Techniques ◀▶

Straus-Shamir Trick ▶

Sparse Representations ◀▶

Aim at increasing zero digits with the help of negative digits:

0xF7 : 1 1 1 1 0 1 1 1

Window Techniques ◀▶

Straus-Shamir Trick ▶

Sparse Representations ◀▶

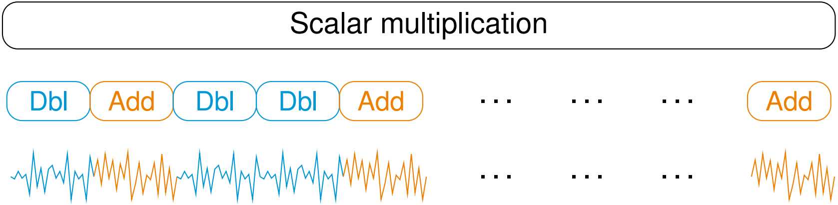
Aim at increasing zero digits with the help of negative digits:

$$\text{0xF7}: \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \bar{1} \quad 0 \quad 0 \quad \bar{1}$$

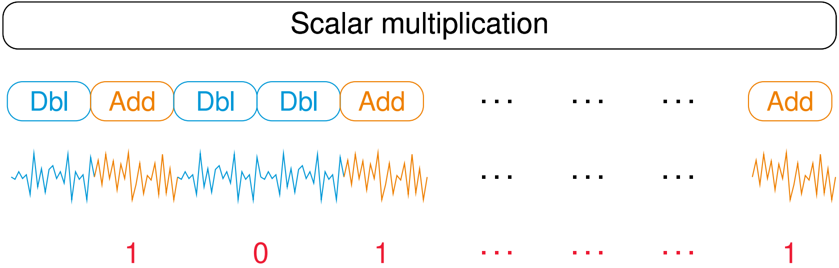
Outline

1. Introduction
2. Elliptic Curve Background
3. Side-Channel Analysis
    - Simple Power Attack
    - State-of-the-Art Countermeasures
4. Our Contribution
5. Conclusion

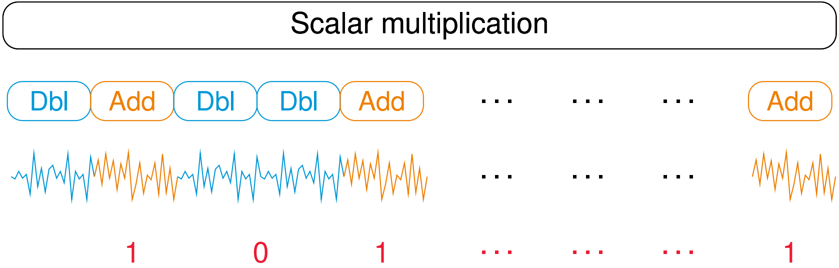
Scalar Multiplication Analysis



Scalar Multiplication Analysis

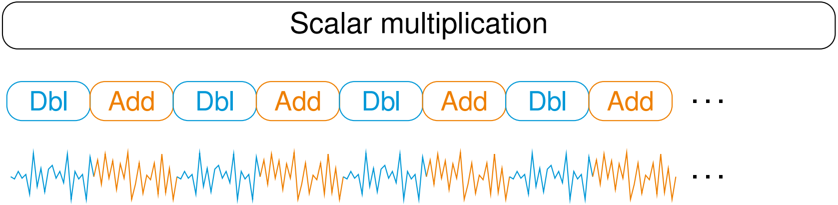


- The secret scalar *k* can be recovered



Regular Algorithms

- Operation flow independent of the secret
- Exemples: Double and Add Always, Montgomery Ladder,…



Atomicity Principle

- Introduced by Chevallier-Mames, Ciet, Joye [2003]
- One sequence of operations in $\mathbb{F}_p$.

$$\blacksquare \left[\begin{array}{l}\text{Multiplication}\\ \text{Addition}\\ \text{Negation}\\ \text{Addition}\end{array}\right.$$

Atomicity Principle

- Introduced by Chevallier-Mames, Ciet, Joye [2003]
- One sequence of operations in $\mathbb{F}_p$.
- Use this sequence with different operands.

$$\blacksquare \left[\begin{array}{l}\text{Multiplication}\\ \text{Addition}\\ \text{Negation}\\ \text{Addition}\end{array}\right.$$

Atomicity Principle

- Introduced by Chevallier-Mames, Ciet, Joye [2003]
- One sequence of operations in $\mathbb{F}_p$.
- Use this sequence with different operands.

$$\blacksquare \left[\begin{array}{l} \text{Multiplication} \\ \text{Addition} \\ \text{Negation} \\ \text{Addition} \end{array} \right.$$

Chevallier-Mames et al. EC Operations

Doubling:	■■■■■■■■■■	$10M + 20A + 10N$
Addition:	■■■■■■■■■■■■■■■■	$16M + 32A + 16N$

Atomicity Algorithms

■■ $\Longrightarrow$ ■ $2M + 3A + 2N$ Longa

Longa EC Operations ▶ [2007]

Doubling:	■■■■	$8M + 12A + 8N$
Addition:	■■■■■■■	$14M + 21A + 14N$

Giraud-Verneuil EC Operations ◀ [Cardis2010]

Atomicity Algorithms

$\Longrightarrow$	$2M + 3A + 2N$	Longa
$\Longrightarrow$	$2S + 6M + 10A$	Giraud-Verneuil

Longa EC Operations ▶ [2007]

Doubling: $8M + 12A + 8N$

Addition: $14M + 21A + 14N$

Giraud-Verneuil EC Operations ◀ [Cardis2010]

Doubling: $2S + 6M + 10A$

Addition: $4S + 12M + 20A$

Outline

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

Doubling:	$10M$
Addition:	$16M$

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

Doubling:	■■■■■■■■■■	$10M$
Addition:	■■■■■■■■■■■	$11M$

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

Doubling:	$11M$	$\Longrightarrow$	
Addition:	$11M$	$\Longrightarrow$	

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

Doubling:  $10M$ $\Longrightarrow$ 

Addition:  $10M$ $\Longrightarrow$ 

How to Improve Patterns?

- More operations in a pattern $\Longrightarrow$ less dummy operations.
- Efficient pattern for both addition and doubling?
- Optimize additions?

Doubling:	$9M$	$\Longrightarrow$	
Addition:	$9M$	$\Longrightarrow$	

oberthur TECHNOLOGIES THE M COMPANY

New Atomic Patterns

All Curve Pattern ▶

Doubling:	■	$3S + 8M + 9A$
Addition:	■	$3S + 8M + 9A$

Most Curve Pattern ▶

Doubling:	■	$2S + 8M + 10A$
Addition:	■	$2S + 8M + 10A$

$a = 0$ Curve Pattern ▶

Doubling:	■	$2S + 7M + 8A$
Addition:	■	$2S + 7M + 8A$

New atomic Patterns

All Curve Pattern

This pattern can be used with all existing elliptic curves.

Most Curve Pattern

$a = 0$ Curve Pattern

New atomic Patterns

All Curve Pattern

This pattern can be used with all existing elliptic curves.

Most Curve Pattern

This pattern restricts the value $I^2 = -a3^{-1}$ to be a quadratic residue. Then we have:

$$3X^2 + aZ^4 = 3(X - IZ^2)(X + IZ^2)$$

$a = 0$ Curve Pattern

New atomic Patterns

All Curve Pattern

This pattern can be used with all existing elliptic curves.

Most Curve Pattern

This pattern restricts the value $I^2 = -a3^{-1}$ to be a quadratic residue. Then we have:

$$3X^2 + aZ^4 = 3(X - IZ^2)(X + IZ^2)$$

$a = 0$ Curve Pattern

For security and efficiency reasons, the curves with $a = 0$ have a dedicated pattern.

Implementation Limits

$[k]P$ for unknown P

- $\mathrm{NAF}_{w=4}$
- ℓ doublings and $\ell/5$ additions

$[k]G$ for fixed point G

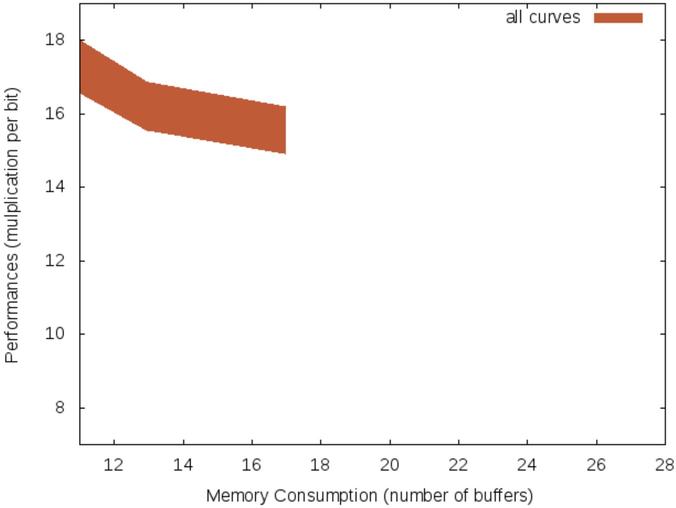
- Precompute $Q = \left[2^{\ell/2}\right]G$ once for all
- Split k and compute $[k]G = [k_0]G + [k_1]Q$
- JSF
- $\ell/2$ doublings and $\ell/4$ additions

oberthur TECHNOLOGIES THE M COMPANY

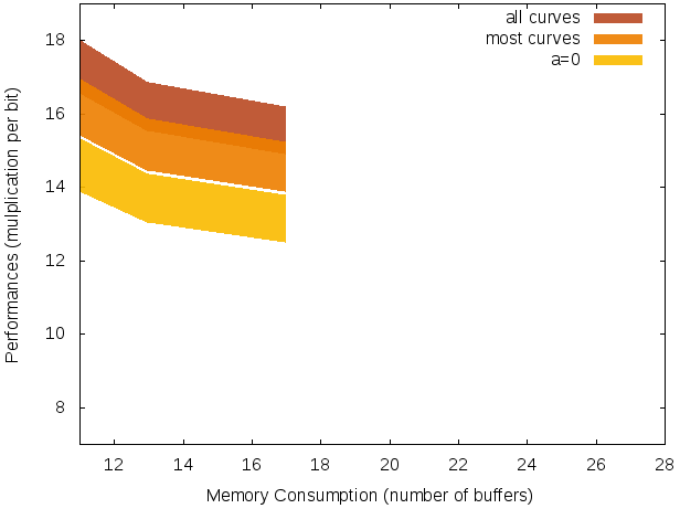
Implementation Characteristics

bit size	192	224	256	320	384	512	521
A/M	0.21	0.21	0.19	0.17	0.16	0.14	0.14
GV A/M	0.30	0.25	0.22	0.16	0.13	0.09	0.09

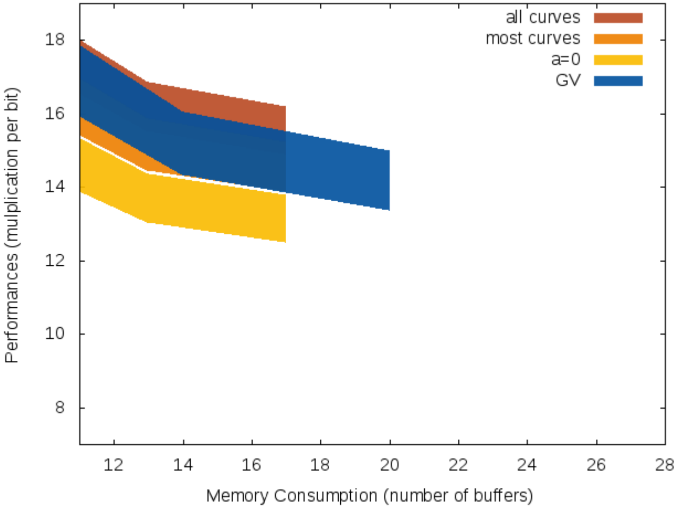
Performances



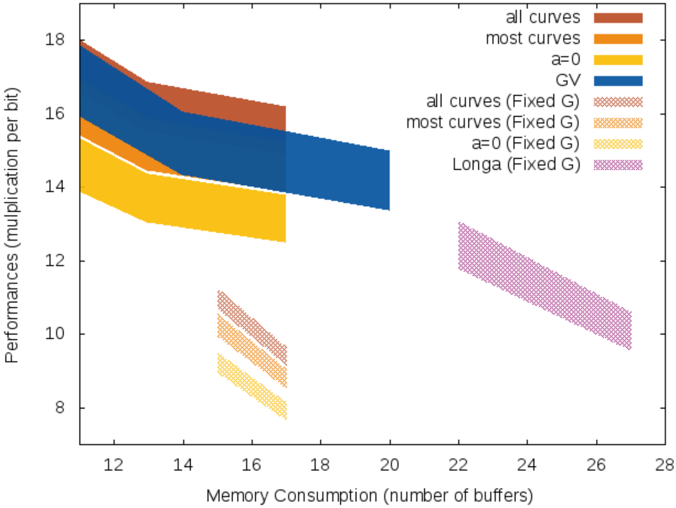
Performances



Performances

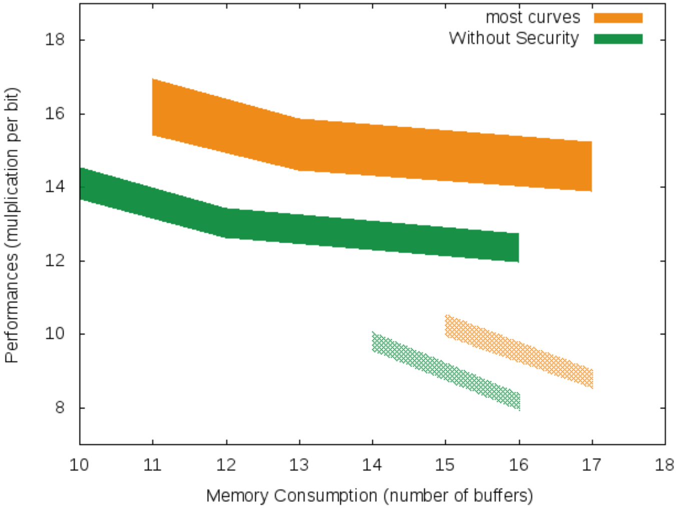


Performances



oberthur TECHNOLOGIES THE M COMPANY

Countermeasure Overhead



Conclusion

- Optimizing additions rather than doublings is a valid strategy for secure implementation.
- First proposition for secure multi-multiplication.
- Most EC protocols can benefit from multi-multiplication implementations.