Moroccan Journal of Algebra and Geometry with Applications
Supported by Sidi Mohamed Ben Abdellah University, Fez, Morocco
Volume 2, Issue 2 (2023), pp 209-217
ISSN: 2820-7114

Title :

A Fast and SPA secure scalar Multiplication for Elliptic Curve Cryptography

Author(s):

Amadou Tall

Moroccan Journal of Algebra and Geometry with Applications
Vol. 2(2) (2023), 209–217

Supported by Sidi Mohamed Ben Abdellah University, Fez, Morocco ISSN: 2820-7114

A Fast and SPA secure scalar Multiplication for Elliptic Curve Cryptography

Amadou Tall
Université Cheikh Anta DIOP de Dakar, Senegal.
e-mail: *amadou7.tall@ucad.edu.sn*

Communicated by Abdenacer Makhlouf
(Received 19 December 2022, Revised 16 July 2023, Accepted 24 July 2023)

Abstract. This paper aims to give a new fast and secure scalar multiplication technique for elliptic curves. The technique also provides protection against side channel attacks, particularly simple power analysis. The method proposed for scalar multiplication is based on Lucas addition-subtraction chains [28].

Key Words: addition chain, addition-subtraction chain, Lucas chains, Lucas addition-subtraction chains, elliptic curve cryptography, scalar multiplication, side-channel attacks.
2010 MSC: Primary 11Y55.

1 Introduction

Elliptic curve cryptography was introduced in 1985 independently by Miller and Koblitz [1, 13, 18]. Given a point P on an elliptic curve over a finite field, computing the scalar multiple kP is central to the actual implementation of elliptic curve cryptography. Various methods have been proposed to speed up and secure this computation. Exponentiation algorithms have been shown to be vulnerable to side-channel analysis, where an attacker observes the power consumption [5]. This attack is known as *Simple Power Analysis* (SPA). There are several algorithms that have been proposed in the literature [5, 10, 8, 12, 21] to resist against SPA. It should be noted that differential side-channel analysis will not be considered in this paper.

In this paper, we give a new fast and secure point multiplication algorithm, which resists SPA. The algorithm is based upon a particular kind of addition-subtraction chain known as *Lucas addition-subtraction chains*. Addition-subtraction chains and Lucas chains have both been studied in connection with speeding up scalar multiplication [23, 16, 26, 1, 25, 6, 8, 24, 17, 18, 11, 27]. However, Lucas addition-subtraction chains have not yet been used before. The Lucas addition-subtraction algorithm we propose is much simpler and as fast as known algorithms that resist SPA.

This paper is organized as follows. In the next section, we provide a brief background on elliptic curves and review Lucas addition-subtraction chains. In Section 3, we present the new scalar multiplication algorithm based on Lucas addition-subtraction chains and show it resists SPA. In Section 4, we compare our scalar multiplication to the classical double-and-add, and NAF scalar multiplication algorithms. A deeper comparison will be done with some scalar multiplications that resist the SPA. Finally, we conclude in the last section.

2 Background

In this section, we first give a brief overview of addition on elliptic curves. For more details, the reader should consult [26]. We then review Lucas addition-subtraction chains [28].

2.1 Elliptic curves

Definition 2.1. An elliptic curve E over a finite field K is given by an equation

$$E(K): \quad y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{1}$$

where $a_1,\ a_2,\ a_3,\ a_4,\ a_6 \in K$ are such that for each point $(x,\ y)$ on E, the partial derivatives do not simultaneously vanish.

In practice, if the characteristic of K is not 2 or 3, then equation for an elliptic curve is usally simplified into $y^2 = x^3 + ax + b$. Here, $a,\ b \in K$, with $4a^3 + 27b^2 \neq 0$. The set $E(K)$ of the rational points of an elliptic curve E (defined over K) is an abelian group where the identity element is a special point $\mathcal{O}$, called the point at infinity.

2.2 The addition law

The set of points of an elliptic curve forms a group under a certain addition rule. We now give this rule explicitly. Let E be as in (1), and let $P = (x_1,\ y_1)$, $Q = (x_2,\ y_2)$ be two points of E, neither of which is $\mathcal{O}$. The inverse of the point P is given by

$$-P = (x_1,\ -y_1 - a_1x_1 - a_3),$$

and their sum $P + Q = (x_3,\ y_3)$ is defined as follows:

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2,$$

$$y_3 = \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3,$$

where λ is:

$$\lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, & \text{if } \ P \neq \pm Q, \\ \\ \frac{3{x_1}^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{if } \ P = Q. \end{cases}$$

2.3 Lucas addition-subtraction chains

Before giving the definition of Lucas addition-subtraction chains, we first define addition chains, Lucas addition chains, and addition-subtraction chains. As can be inferred from their name, Lucas addition-subtraction chains combine these three types of chains. For more details on these various types of chains, see [23, 29, 22, 30, 20, 8, 3, 28].

Definition 2.2. Let n be an integer. A sequence $c = \{1 = a_0,\ a_1, \ldots, a_l = n\}$ is called an addition chain for n if and only if for each $a_i \in c$, there exists j, k with $0 \leq j, k < i$ such that:

$$a_i = a_j + a_k.$$

Example 2.3. The sequence $\{1,\ 2,\ 3,\ 5,\ 7,\ 9,\ 14,\ 19\}$ is an addition chain for 19.

Lucas addition chains are a special case of addition chains.

Definition 2.4. An addition chain $c = \{a_0, a_1, \ldots, a_l\}$ is a Lucas addition chain if and only if:

$$\text{if } a_i = a_j + a_k \text{ for some } 0 \leq i, j, k \leq l, \text{ then } a_j = a_k \text{ or } |a_j - a_k| \in c.$$

Example 2.5. The sequence $\{1, 2, 3, 5, 7, 9, 14, 19\}$ is a Lucas addition chain for 19.

Notice that, in this example, 14 is obtained by $7+7$ and not $9+5$.

Example 2.6. The sequence $\{1, 2, 3, 5, 10, 12\}$ is an addition chain for, but not a Lucas addition chain.

Addition-subtraction chains are a generalization of addition chains.

Definition 2.7. A sequence $c = \{1 = a_0, a_1, \dots, a_l = n\}$ is called an addition-subtraction chain for an integer n if and only if for each $a_i \in c$, then $a_i > 0$ and there exists j, k with $0 \le j, k < i$ such that

$$a_i = a_j + a_k \text{ or } a_i = a_j - a_k.$$

Example 2.8. The sequence $\{1, 2, 4, 8, 16, 24, 22\}$ is an addition-subtraction chain for 22.

It is clear that a Lucas addition chain is an addition chain, and any addition chain is an addition-subtraction chain. We now define Lucas addition-subtraction chains.

Definition 2.9. Let n be a integer. A Lucas addition-subtraction chain for n is a sequence $c = \{a_0 = 1, a_1, \dots, a_l = n\}$ such that for each $a_i \in c$, there exists j, k with $0 \le j,\ k < i$ satisfying

$$a_i = \begin{cases} a_j + a_k & \text{and } |a_j - a_k| \in c \cup \{0\}, \\ & or \\ a_j + 1, & \\ & or \\ a_j - a_k. & \end{cases}$$

Example 2.10. Let F_k be the k^{th} Fibonacci number. That is

$$F_k = \begin{cases} 1, & \text{for } k = 0, 1, \\ F_{k-1} + F_{k-2}, & \text{for } k \ge 2. \end{cases}$$

Then $\{F_1, F_2, \dots, F_l\}$ is a Lucas addition-subtraction chain for F_l.

Example 2.11. $\{1, 2, 3, 5, 10, 20, 19\}$ is a Lucas addition-subtraction chain for 19.

Example 2.12. $\{1, 2, 3, 4, 7, 10, 11, 9\}$ is a Lucas addition-subtraction chain for 9.

Throughout the remainder of the paper, we will use the shorthand LASC to denote a Lucas addition-subtraction chain. We now give a simple way to create short LASCs with the following theorem.

Theorem 2.13. Let n be an integer. A Lucas addition-subtraction chain for n can be obtained recursively in the following way:

1. If n is even, then append n to a Lucas addition-subtraction chain for $\frac{n}{2}$.
2. If $n \equiv 1 \bmod 4$, then append n to a Lucas addition-subtraction chain for $n-1$.
3. If $n = a2^{k+1} + (2^k - 1)$ for some k, then append n to a Lucas addition-subtraction chain for $n+1$.

Proof. We need just to show that each of the three steps given above satisfy the criteria for LASCs. If c is a LASC for $\frac{n}{2}$, then appending n to c will still be a valid LASC as $n = \frac{n}{2} + \frac{n}{2}$. If instead c is a valid LASC for $n-1$, then the definition for LASCs allows for adding 1 to an element of c, so we can append n to c. Finally, if c is a LASC for $n+1$, then always we have $1 \in c$, and we may append $(n+1)-1 = n$ to c. □

For ease of notation, we label the steps in Theorem 2.13 as DBL (doubling), ADD (adding), and SUB (subtracting).
Notice a DBL step is a doubling of the previous final element of the chain, an ADD step is an addition of 1 to the previous final element of the chain, and a SUB step is a subtraction by 1. We illustrate the theorem in the next two examples.

Example 2.14. Using Theorem 8, A Lucas addition-subtraction chain for 124 can be obtained as follows:

$$\begin{aligned} \mathbf{124} &= 62 \cdot 2, \\ \mathbf{62} &= 31 \cdot 2, \\ \mathbf{31} &= 32 - 1, \\ 32 &= \mathbf{16} \cdot 2, \\ 16 &= \mathbf{8} \cdot 2, \\ &\vdots \\ 2 &= 1 \cdot 2. \end{aligned}$$

The corresponding Lucas addition-subtraction chain is:

$$\{1, 2, 4, 8, 16, 32, 31, 62, 124\}.$$

Example 2.15. Using Theorem 8, a Lucas addition-subtraction chain for 242 can be obtained as follows:

$$\begin{aligned} \mathbf{242} &= 121 \cdot 2, \\ \mathbf{121} &= 120 + 1, \\ \mathbf{120} &= 60 \cdot 2, \\ \mathbf{60} &= 30 \cdot 2, \\ 30 &= \mathbf{15} \cdot 2, \\ \mathbf{15} &= 16 - 1, \\ 16 &= \mathbf{8} \cdot 2, \\ &\vdots \\ \mathbf{2} &= 1 \cdot 2. \end{aligned}$$

The corresponding Lucas addition-subtraction chain is:

$$\{1, 2, 4, 8, 16, 15, 30, 60, 120, 121, 242\}.$$

We note that there are other approaches to finding Lucas addition-subtraction chains (see [28, **?**]). However, this algorithm is significant because of its simplicity and the short length of the chains produced. As seen above, each step is either a doubling (DBL), or an addition (ADD) or subtraction (SUB) by 1. An ADD or SUB step is always followed by a minimum of two successive doubling steps. This makes the computation very efficient.
We will see later that we generally have the same number of ADD and SUB (approximately) when we compute the LASC of a random prime p using the approach in Theorem 2.13.

3 The new scalar multiplication algorithm

As mentioned before, one of the key operations in the implementation of elliptic curve cryptography is computing scalar multiples of points. Let P be a point on an elliptic curve, and k be the scalar we wish to use. The following algorithm computes kP by constructing a Lucas addition-subtraction chain for k.

Algorithm 1 scalarMultiplication(k, P)

Require: k : integer, P: a point of an elliptic curve E
Ensure: kP : a point of E

```
if k even then
   return 2(scalarMultiplication(k/2, P))
else
   if ⌊k/2⌋ even then
      return 2(scalarMultiplication(⌊k/2⌋, P)) + P
   else
      return 2(scalarMultiplication(⌊k/2⌋ + 1, P)) − P
   end if
end if
```

The algorithm is at the *worst case* $2/3(\lambda(k))DBL + (\lambda(k)/3)ADD$ which is the average cost of the double-and-add scalar multiplication, where $\lambda(k) = \lfloor \log_2(k) \rfloor$.

3.1 Side-Channel Analysis

Side-channel attacks[15, 14] are any attacks based on *side-channel information*. Side-channel information refers to information that can be gained from the physical encryption device. This includes, for example, timing information, power consumption, and electromagnetic leaks. In particular, since the computational cost of addition and doubling of points on elliptic curves are distinguishable by measuring the power consumption, an attacker can use SPA to exploit this information.

Several counter-measures have been proposed against these attacks [5, 12, 7, 3]. In this work, the new scalar multiplication avoids simple power analysis (SPA) by taking advantage of the indistinguishability of addition and subtraction and that the ratio $\#ADD/\#SUB$ is very close to 1/2. We assume that an attacker can use the power consumption to determine the sequence of addition (or subtraction) and the doubling steps of our algorithm. However, this will not produce enough information about the binary expansion of the scalar k. If the attacker knows that there are m addition steps, then if we have used Algorithm 1 there are roughly 2^m possibilities for k. This follows because each addition step could be either an ADD or SUB step. The attacker cannot distinguish between any two possible candidates for k.

We illustrate this concept with an example. Suppose an SPA attack yields the sequence of doublings and additions/subtractions used to compute a Lucas addition-subtraction chain for an integer n. We list such a sequence in the second column below. The third and fourth columns show how knowing this sequence does not determine n. This example demonstrates that knowing when to double and when to add (or subtract) by itself does not help finding k because doubling and additions/subtractions occur during the same corresponding steps within the process for these two chains. In fact, this same sequence can also lead to chains for 1915, 1923, 1925, 2173, 2179, and 2181.

An attacker can check all the possibilities (knowing when to make an addition or subtraction) and find a set of possible values of k, but this set will contain almost 2^m possible values.

Step	Operation	Chain for **1917**	Chain for **2171**
1	4 DBL	{1, 2, 4, 8, 16}	{1, 2, 4, 8, 16}
2	1 ADD	$15 = (1111)_2$	$17 = (10001)_2$
3	5 DBL	{30, 60, 120, 240, 480}	{34, 68, 136, 272, 544}
4	1 ADD	$479 = (111011111)_2$	$543 = (1000011111)_2$
5	2 DBL	{958, 1916}	{1086, 2172}
6	1 ADD	**1917**	**2171**

Figure 1: Two chains with the same doubling and addition/subtraction sequence.

We claim that we will have roughly the same number of ADD and SUB steps for a randomly chosen k. We expect the odd values in the chain computed by Algorithm 1 to be uniformly distributed mod 4. That is, we expect about half of them to be $\equiv 1 \bmod 4$, while half are $\equiv 3 \bmod 4$. From this it follows that the number of ADDs and SUBs will be approximately equal. The data in the next section supports this conclusion.

4 Comparisons with classical algorithms

In this section, our new proposed scalar multiplication algorithm will be compared to the classic double-and-add (binary) method, the non-adjacent form (NAF) method, and the FRLBM method [18]. The FRLBM method resists SPA under the assumption that there are the same number of DBL's and ADD's in a specific mixed coordinate. We will see that we obtain almost the same results whereas our algorithm is much simpler. We implemented each method with 1000000 random 160-bit primes, and display the average number of addition and doubling steps required. The next three tables do the same for 384-bit , 512-bit, and 1024-bit integers.

Method	binary	NAF	LASC	FRLBM
Addition	88	52	55	107
Doubling	159	160	156	106
Total	247	212	211	213

Figure 2: Table comparing scalar multiplication methods for 1000000 random 160 bit primes

Method	binary	NAF	LASC	FRLBM
Addition	202	117	130	256
Doubling	383	384	384	256
Total	585	501	524	512

Figure 3: Table comparing scalar multiplication methods for 1000000 random 384 bit primes

From the tables we see that Algorithm 1 (which uses LASC's) is comparable in efficiency to the NAF and FRLBM methods, while each are more efficient than the classical binary double–and–add technique.

We further analyzed our algorithm to determine the distribution of ADD versus SUB steps occurring in the addition steps. When we have roughly the same number of additions as subtractions, it

Method	binary	NAF	LASC	FRLBM
Addition	265	168	173	341
Doubling	511	512	511	341
Total	776	680	684	682

Figure 4: Table comparing scalar multiplication methods for 1000000 random 512 bit primes

Method	binary	NAF	LASC	FRLBM
Addition	530	350	455	682
Doubling	1023	1024	912	683
Total	1553	1374	1367	1365

Figure 5: Table comparing scalar multiplication methods for 1000000 random 1024 bit primes

decreases the chance of an attacker finding the right value for k. In each table we display the average number of DBL, ADD, and SUB steps required.

Operation	Average number of operations
ADD	28.06 additions (+1's)
DBL	159.33 doublings
SUB	26.73 subtractions (-1's)

Figure 6: 1000000 random prime numbers of 160–bits

Operation	Average number of operations
ADD	44.05
DBL	255.33
SUB	42.73

Figure 7: 1000000 random prime numbers of 256–bits

Operation	Average number of operations
ADD	65.06
DBL	383.33
SUB	65.73

Figure 8: 500000 random prime numbers of 384–bits

5 Conclusion

This paper has presented a new algorithm to compute scalar multiplication on elliptic curves. Our method is fast, much simpler, and as secure as previously known algorithms in protecting against SPA. The key tool used for the algorithm is Lucas addition-subtraction chains. Generally, these chains

Operation	Average number of operations
ADD	86.72
DBL	511.33
SUB	85.39

Figure 9: 250000 random prime numbers of 512–bits

have shorter length than the traditional Lucas addition chains [23, 28] and have the same properties. We leave it as future work to examine the potential use of Lucas addition-subtraction chains in the elliptic curve method ECM for factorization [4, 17, 18].

References

[1] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren. Handbook of Elliptic and Hyperelliptic Curve Cryptography, CRC Press, (2005).

[2] D. J. Bernstein, T. Lange, Faster addition and doubling on elliptic curves, Advances in Cryptology ASIACRYPT (2007), Lecture Notes in Computer Science, (4833), 29–50.

[3] A. Brauer, On addition chains, Bull. Amer. Math. Soc. **45** (1939), 736–739.

[4] I. Chelli, Fully Deterministic ECM. Available at `http://caramel.loria.fr/sem-slides/200909251030.pdf`.

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

[6] M. Ciet, M. Joye, K. Lauter, P. L. Montgomery, Trading Inversions for Multiplications in Elliptic Curve Cryptography, Des. Codes Cryptogr., (2003) 189–206.

[7] V. Dimitrov, L. Imbert, and P. K. Mishra. Efficient and secure Elliptic Curve Point Multiplication Using Double-Base Chains, ASIACRYPT 2005, **3788** of Lecture Notes in Computer Science, 59–78.

[8] D.M. Gordon. A survey of fast exponentiation methods, Journal of Algorithms **27** (1) (1998) 129–146.

[9] D. Hankerson, A. J. Menezes, S. Vanstone, Guide to elliptic curve cryptography, Springer-Verlag New York, Inc, Secawcus NJ, USA (2003).

[10] M. Hedabou, P. Pinel and L. Beneteau. A comb method to render ECC resistant against side channel attacks, Report 2004/342, Cryptology ePrint Archive, 2004 **http;//eprint.iacr.org/2004/342**.

[11] M. Joye, Fast Point Multiplication on Elliptic Curves With Precomputation, J. von zur Gathen, Arithmetic of Finite Fields WAIFI 2008. (5130) 36–46, Lecture Notes in Computer Science, Springer 2008.

[12] M. Joye and J.-J. Quisquater, Hessian elliptic curves and side-channel attacks, Cryptographic Hardware and Embedded Systems-CHESS (2001), **2162** of Lecture Notes in Computer Science. Springer-Verlag.

[13] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. **48** (117) (1987) 203–209.

[14] P. C. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. Advances in Cryptology - CRYPTO 96, **1109** of Lecture Notes in Computer Science, 104-113. Springer-Verlag.

[15] P. Kocher, J. Jaffe and B. Jun. Differential power analysis, M.J. Wiener, editor, Advances in Cryptology – CRYPTO '99, volume 1666 of Lecture Notes in Computer Science **(Springer 1999)** 388–397.

[16] K. Koyama, Y Tsuruoka, Speeding elliptic cryptosystems using a signed binary window method. Advances in Cryptology **740** (1992) 345 - 357.

[17] A. Kruppa, Factoring into Large primes with $p-1$, $p+1$, and ECM. Available at `http://cado.gforge.inria.fr/workshop/slides/kruppa.pdf`.

[18] A. Kruppa, A software implementation of ECM for NFS. Available at `http://hal.archives-ouvertes.fr/docs/00/41/90/94/PDF/RR-7041.pdf`.

[19] D. P. Le, Fast quadrupling of a point in elliptic curve cryptography, preprint Information Processing Letters (23 March 2011).

[20] M. Mignotte, A. Tall, A note on addition chains, International Journal of Algebra **5** (6) (2011) 269–274.

[21] V. S. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology: Proceedings of Crypto 85. (218) 417–426 Lecture Notes in Computer Science.

[22] F. Morrain, J. Olivos Speeding up the computation on an elliptic curve using addition-subtraction chains, Informatique théorique et applications **24**, (6) (1990) 531–543.

[23] P. L. Montgomery. Evaluating recurrences of form $X_{m+n} = f(X_m, X_n, X_{m-n})$ via Lucas Chains, January 1992.

[24] F. Morain, J. Olivos. Speeding up the computations on an elliptic curve using addition-subtraction chains, RAIRO Informatique théoretique et application **24** (1990) 531–543 .

[25] Y. Sakai, K. Sakurai, Efficient scalar multiplication on elliptic curve with direct computations of several doublings, IEICE Transactions Fundamentals **E84-A(1)** (2001) 120–129.

[26] J. H. Silverman, The arithmetic of elliptic curves. Springer-Verlag (1986).

[27] M. Stam, On Montgomery-Like Representations for Elliptic Curves over $GF(2^k)$ Available at `http://www.iacr.org/archive/pkc2003/25670240/25670240.pdf`.

[28] A. Tall, A generalization of Lucas addition chains, Bull. Math. Soc. Sci. Math. **55** (103)(2012) 79–93.

[29] H. Volger. Some results on addition-subtraction chains, Information Processing Letters **20** (3)(8 April 1985) 155–160.

[30] C. Wang, C. Lin and C. Chang, A method for computing Lucas sequences, Computers and Mathematics with Applications **38** (1999) 187–196.