

**UNIVERSIDAD DE SANTIAGO DE CHILE**  
**FACULTAD DE CIENCIA**  
Departamento de Matemática y Ciencia de la Computación



**Consumo energético de contramedidas para Side-Channel Attack Pasivo  
en Curva Elíptica sobre cuerpos primos**

**Sebastián Alonso Holloway Inostroza**

**Profesor Guía:** Dr. Rodrigo Abarzúa Ortiz

Trabajo de titulación presentado en conformidad  
a los requisitos para obtener el título de Analista  
en Computación Científica

**Santiago - Chile**  
**2019**

© **Sebastian Alonso Holloway Inostroza**

Se autoriza la reproducción parcial o total de esta obra, con fines académicos, por cualquier forma, medio o procedimiento, siempre y cuando se incluya la cita bibliográfica del documento. Queda prohibida la reproducción parcial o total de esta obra en cualquier forma, medio o procedimiento sin permiso por escrito del autor.

# Resumen

Hoy en día la mayoría de los dispositivos electrónicos que se utilizan están conectados por medio de lo que se conoce como el Internet de las Cosas (IoT por sus siglas en inglés), no obstante, la información que se transmite entre estos dispositivos podría ser vulnerada. Para prevenir esto, se puede utilizar criptosistemas que brindan seguridad a la información, tales como el RSA o Curva Elíptica (ECC). Dado que ECC necesita una clave de menor longitud, es ideal para microprocesadores de baja capacidad. Sin embargo, ECC posee vulnerabilidades conocidas como Side Channel Attacks (SCA), estos ataques utilizan las fugas físicas de información para obtener dicha clave (Por ejemplo, radiación electromagnética o análisis de energía). Este problema impulsó la creación de contramedidas para los distintos tipos de SCA. En general, los estudios relacionados a estas contramedidas se enfocan en el ataque y los costos de la contramedida, y no de un análisis en la seguridad ni en el consumo energético de ellas. Hasta la fecha no son muchas las investigaciones (McCann, Eder, & Oswald, 2015) sobre los costos energéticos de las contramedidas. Por lo que esta tesis busca analizar que contramedida es la más eficiente respecto al consumo de energía y la protección que brinda.

**Palabras clave:** Criptografía, Arquitectura de 8-bits, Curva Elíptica, Contramedidas Side Channel Attack.

# Tabla de contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivo General . . . . .	2
1.2.1. Objetivos Específicos . . . . .	2
<b>2. Criptografía en Curva Elíptica</b>	<b>3</b>
2.1. Curva Elíptica . . . . .	3
2.2. Leyes de Grupo . . . . .	4
2.3. Multiplicación escalar . . . . .	5
2.4. Longitud de Claves . . . . .	6
2.5. Sistemas de Coordenadas . . . . .	6
2.5.1. Coordenadas Proyectivas $\mathcal{P}$ . . . . .	7
2.5.2. Coordenadas Jacobianas $\mathcal{J}$ . . . . .	7
2.6. Fórmulas Explícitas Jacobianas . . . . .	8
2.6.1. Doblado Jacobiano . . . . .	8
2.6.2. Adición Jacobiana . . . . .	8
<b>3. Side Channel Attack en ECC</b>	<b>9</b>
3.1. Simple Power Analysis . . . . .	10
3.2. Differential Power Analysis . . . . .	10
3.3. Address-Bit DPA . . . . .	10
3.4. RPA, ZPA y SVA . . . . .	11
3.4.1. RPA . . . . .	11
3.4.2. ZPA . . . . .	11
3.4.3. SVA . . . . .	11
<b>4. Contramedidas para Side Channel Attack</b>	<b>12</b>
4.1. Contramedidas para Simple Power Analysis . . . . .	13

4.1.1.	Fórmula Unificada de Brier-Joye . . . . .	13
4.1.2.	Always Double and Add de Coron . . . . .	13
4.1.3.	Montgomery Ladder de Brier-Joye . . . . .	14
4.1.4.	Double-Add de Joye . . . . .	15
4.1.5.	Add-Only de Joye . . . . .	15
4.1.6.	Signed Digit Method . . . . .	16
4.1.7.	Bloques Atómicos . . . . .	16
4.2.	Construcciones para Differential Power Analysis . . . . .	18
4.2.1.	Primera Construcción de Coron . . . . .	18
4.2.2.	Construcción de Clavier-Joye . . . . .	18
4.2.3.	Segunda Construcción de Coron . . . . .	18
4.2.4.	Tercera Construcción de Coron . . . . .	19
4.2.5.	Método 2P* de Ciet-Joye . . . . .	19
4.3.	Construcciones para Address-bit DPA . . . . .	20
4.3.1.	Montgomery Powering Ladder . . . . .	20
4.4.	Construcciones para Ataques Múltiples . . . . .	21
4.4.1.	BRIP . . . . .	21
4.4.2.	Construcción de Kim . . . . .	21
4.4.3.	Construcción de Ha . . . . .	22
<b>5.</b>	<b>Implementación de Códigos</b>	<b>24</b>
5.1.	Implementación . . . . .	24
5.1.1.	Librería Nano-ECC . . . . .	24
5.1.2.	Implementaciones . . . . .	29
5.1.3.	Método de comprobación de las implementaciones . . . . .	30
5.2.	Limitaciones . . . . .	31
5.2.1.	Generación de un punto aleatorio . . . . .	31
5.2.2.	Limitaciones de memoria . . . . .	31
<b>6.</b>	<b>Resultados del análisis</b>	<b>33</b>
6.1.	Método de medición . . . . .	33
6.2.	Resultados de la medición . . . . .	34
6.2.1.	Left to Right . . . . .	35
6.2.2.	Always Double and Add . . . . .	35
6.2.3.	Modified Montgomery Ladder . . . . .	36
6.2.4.	Double Add de Joye . . . . .	37
6.2.5.	Add-Only . . . . .	37

6.2.6. Randomized Montgomery Ladder . . . . .	38
6.2.7. Signed Digit . . . . .	38
6.2.8. Clavier-Joye . . . . .	39
6.2.9. Segunda Contramedida de Coron: Blinding Point . . . . .	39
6.2.10. Tercera Contramedida de Coron . . . . .	40
6.2.11. Método 2P* de Ciet-Joye . . . . .	40
6.2.12. Montgomery Powering Ladder . . . . .	41
6.2.13. BRIP . . . . .	41
6.2.14. Contramedida de Kim . . . . .	42
6.3. Resumen de Resultados . . . . .	42
6.3.1. Resultados para Simple Power Analysis . . . . .	42
6.3.2. Resultados para Differential Power Analysis . . . . .	43
6.3.3. Resultados para Address-bit DPA . . . . .	43
6.3.4. Resultados para Ataque Múltiples . . . . .	44

<b>Bibliografía</b>	<b>47</b>
---------------------	-----------

# Índice de tablas

2.1. Comparación de la cantidad de bits requeridas en RSA y ECC para diferentes niveles de seguridad . . . . .	6
4.1. Bloque Atómico para Doblado Jacobiano . . . . .	17
5.1. Implementaciones de Algoritmos . . . . .	30
6.1. Resultados Left to Right . . . . .	35
6.2. Resultados Always Double and Add . . . . .	36
6.3. Resultados Modified Montgomery Ladder . . . . .	36
6.4. Resultados Double Add de Joye . . . . .	37
6.5. Resultados Add-Only de Joye . . . . .	37
6.6. Resultados Randomized Montgomery Ladder . . . . .	38
6.7. Resultados Signed Digit Method: Left-to-Right . . . . .	38
6.8. Resultados Contramedida de Clavier-Joye . . . . .	39
6.9. Resultados Segunda Contramedida de Coron: Blinding Point . . . . .	39
6.10. Resultados Tercera Contramedida de Coron . . . . .	40
6.11. Resultados Método 2P* de Ciet-Joye . . . . .	40
6.12. Resultados Montgomery Powering Ladder . . . . .	41
6.13. Resultados BRIP . . . . .	41
6.14. Resultados Contramedida de Kim . . . . .	42
6.15. Resumen Resultados SPA . . . . .	43
6.16. Resumen Resultados DPA . . . . .	43
6.17. Resumen Resultados Address-bit DPA . . . . .	43
6.18. Resumen Resultados Ataques Múltiples . . . . .	44

# Índice de figuras

2.1. Operaciones de Grupo . . . . .	5
3.1. Esquema de un SCA . . . . .	9
3.2. Gráfica de energía de un SPA . . . . .	10
5.1. Características de NanoECC . . . . .	25
5.2. Tiempos de ejecución Left-to-Right 128 bits . . . . .	26
5.3. Tiempos de ejecución Left-to-Right 192 bits . . . . .	27
5.4. Tiempos de ejecución Left-to-Right 256 bits . . . . .	28
5.5. Tiempos de ejecución Left-to-Right 384 bits . . . . .	29
6.1. Esquema del circuito implementado . . . . .	34



# Capítulo 1

## Introducción

### 1.1. Motivación

Hoy en día es común ver que los dispositivos embebidos puedan conectarse por medio del desarrollo de sistemas o a través protocolos de comunicación, como lo son las redes de sensores inalámbricas o la identificación de radiofrecuencia. Esto es lo que se conoce como el internet de las cosas (IoT por sus siglas en inglés), esta busca una interconexión digital de los objetos cotidianos con la internet. Las características de la internet de las cosas y la versatilidad de sus aplicaciones en dispositivos permiten abrir líneas de investigación en distintas áreas, incluso en el área de la seguridad de la información, la cual representa un problema (Jing, Vasilakos, Wan, Lu, & Qiu, 2014; Strobel, Oswald, Richter, Schellenberg, & Paar, 2014). Un elemento común que tienen los dispositivos embebidos de la internet de las cosas, son los microcontroladores, por lo que es importante que posean algún mecanismo de seguridad debido a la importancia de estos. En este escrito se enfoca en la criptografía de clave pública (PKC por sus siglas en inglés), la cual esta basada en la criptografía asimétrica. PKC consiste en la utilización de un par de claves, una de estas claves es de conocimiento público en un canal de comunicación, a esta se le conoce como clave pública. Y la otra clave solo es conocida por el usuario a quien se le asigna, esta es la clave privada (Wenger, & Großschadl, 2012). Algunos ejemplos de PKC son el sistema de clave pública RSA y el criptosistema de curva elíptica (ECC) propuesto por Koblitz (1987) y Miller (1985), ambos sistemas proveen la misma seguridad criptográfica, sin embargo, los requisitos de las claves que utilizan son distintos. Por ejemplo, una clave de 256 bits en ECC posee el mismo nivel de seguridad que una clave de 3072 bits en RSA (Giry, & Quisquater, 2011). Debido a esto, ECC presenta una ventaja por sobre el RSA en dispositivos embebidos (Giry, & Quisquater, 2011). Por lo que ECC es una buena opción para implementar en dispositivos embebidos. Sin embargo, es importante el desarrollo de técnicas de protección para los algoritmos criptográficos, ya que

al implementarse en sistemas embebidos, puede haber riesgos de generar ataques conocidos como Side Channel Attack, los que ponen en peligro la seguridad de las aplicaciones usadas en el internet de las cosas.

Los Side Channel Attack pasivos explotan el proceso criptográfico para deducir los bits de la clave secreta. Algunos ejemplos son los ataques de tiempo (Kocher, 1996), los análisis de energía (Kocher, Jaffe, & Jun, 1999) o la radiación electromagnética (Joye, & Quisquater, 2001; Gandolfi, Mourtel, & Olivier, 2001).

Existen dos estrategias generales para estos ataques, el Simple Side Channel Analysis (SSCA) (Kocher, 1996) el cual analiza las diferencias en las fugas físicas de información de un dispositivo en una sola multiplicación escalar. Y por otro lado esta el Differential Side Channel Analysis (DSCA) (Kocher, Jaffe, & Jun, 1999), el cual necesita de varias multiplicaciones escalares para obtener la clave, ya que este utiliza técnicas estadísticas para obtener la información (Popp, Mangard, & Oswald, 2007).

En general, hay varios estudios respecto a Side Channel Attack y sus contramedidas en curva elíptica, algunos de ellos son los reportes de Avanzi (2005), libros (Cohen, y otros, 2005), surveys (Fan & Verbauwhede, 2012; Fan, y otros, 2010). Estos trabajos previos, en general, presentan estudios desde un punto de vista de los ataques y sus contramedidas, y no de un análisis en la seguridad ni en el consumo energético de las diferentes contramedidas.

## **1.2. Objetivo General**

Analizar el consumo energético y tiempo de ejecución de las contramedidas para Side Channel Attack pasivo para el criptosistema de curva elíptica sobre cuerpos primos, implementadas en una arquitectura de 8-bits.

### **1.2.1. Objetivos Específicos**

- Realizar un estudio respecto al estado del arte de las distintas contramedidas de Side Channel Attack Pasivos.
- Seleccionar contramedidas para su implementación en un dispositivo embebido.
- Medir el consumo energético de las contramedidas.
- Generar un documento con los resultados obtenidos.

## Capítulo 2

# Criptografía en Curva Elíptica

El presente capítulo tiene por finalidad explicar sobre el trasfondo matemático relacionado a la investigación propuesta.

Se explicará sobre el criptosistema de curva elíptica, la multiplicación escalar con el método Left-to-Right, los tipos de coordenadas que se utilizarán dentro de la investigación y las fórmulas explícitas con las que trabajan las contramedidas para los ataques de canal lateral (SCA)

### 2.1. Curva Elíptica

El criptosistema de curva elíptica (ECC) fue propuesto por Koblitz (1987) y Miller (1985), este sistema define una curva elíptica  $E$  sobre un cuerpo, generalmente primo,  $\mathbb{F}_p$  por medio de la ecuación de Weierstrass:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

donde  $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_p$  y  $\Delta \neq 0$ , donde  $\Delta$  es el discriminante de  $E$  tal que:

$$d_2 = a_1^2 + 4a_2$$

$$d_4 = 2a_4 + a_1a_3$$

$$d_6 = a_3^2 + 4a_6$$

$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

Esta ecuación puede ser reducida mediante cambio de variables. Particularmente, si  $\mathbb{F}_p$  tiene característica diferente de 2 o 3, cada punto afín  $(x, y)$  se puede reemplazar de la siguiente manera:

$$(x, y) \rightarrow \left( x - \frac{a_2}{3} - \frac{a_1^2}{12}, y - \frac{a_1}{2}x - \frac{a_3}{2} + \frac{a_1^3}{24} + \frac{a_1 a_2}{6} \right)$$

lo cual reduce la ecuación de Weierstrass a la ecuación corta de Weierstrass :

$$y^2 = x^3 + ax + b$$

donde  $a, b \in \mathbb{F}_p$ . Entonces el discriminante de la curva es  $\Delta = -16(4a^3 + 27b^2)$ , con  $4a^3 + 27b^2 \neq 0$  (Menezes, Vanstone, & Hankerson, 2004).

## 2.2. Leyes de Grupo

Sea  $E$  una curva elíptica definida sobre el cuerpo primo  $\mathbb{F}_p$  con característica diferente de 2 o 3 y sean los puntos  $P = (x_1, y_1)$  y  $Q = (x_2, y_2)$  pertenecientes a esa curva.

El elemento neutro de la curva es el punto al infinito ( $P_\infty$ ). La identidad,  $P + P_\infty = P_\infty + P = P$  para todo  $P \in E(\mathbb{F}_p)$ . Inverso aditivo, si  $P = (x, y) \in E(\mathbb{F}_p)$ , entonces  $(x, y) + (x, -y) = P_\infty$ , donde el punto  $(x, -y)$  esta denotado por  $-P$ .  $-P$  también es un punto de  $E(\mathbb{F}_p)$ , además,  $-P_\infty = P_\infty$ . Luego están las operaciones de grupo, las cuales son la adición (Add) y doblado (Double), las cuales están dadas por las siguientes ecuaciones:

Sea  $P = (x_1, y_1)$  y  $Q = (x_2, y_2)$

$$P + Q = (x_3, y_3) = \lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1 ; \lambda = \frac{y_1 - y_2}{x_1 - x_2}$$

$$2P = (x_3, y_3) = \lambda^2 - 2x_1, \lambda(x_1 - x_3) - y_1 ; \lambda = \frac{3x_1^2 - a}{2y_1}$$

$$y(x_1, y_1) + (x_1, -y_1) = P_\infty$$

Una representación de estas ecuaciones se muestra en la Figura 2.1.

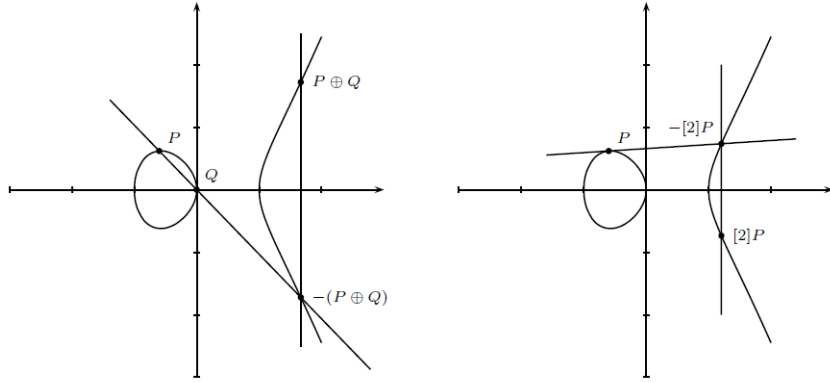


Figura 2.1: Operaciones de Grupo

Fuente: Cohen et al., (2005)

Donde se puede apreciar la utilización del método de la secante tangente para formar las operaciones de grupo (Cohen et al., 2005; Menezes, Vanstone, & Hankerson, 2004).

### 2.3. Multiplicación escalar

Con las operaciones definidas, un algoritmo fundamental en ECC es la multiplicación escalar  $[k]P$ , donde  $k \in \mathbb{N}$  y  $P \in E(\mathbb{F}_p)$ . Entonces podemos computar el  $[k]P$  con la forma:

$$[k]P = \underbrace{P + P + \dots + P}_{k \text{ veces}}$$

Si bien existen varios métodos de realizar la multiplicación escalar, el escrito se enfoca en el método de Double-and-Add, el cual posee dos versiones, el Left-to-Right y el Right-to-Left (Verneuil, 2012; Menezes, Vanstone, & Hankerson, 2004). Particularmente se utilizará el algoritmo Left-to-Right que se muestra en el Algoritmo 1.

---

**Algorithm 1:** Left-to-Right

---

**Input:** Punto  $P \in E(F_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```

1 begin
2    $R_0 \leftarrow P_\infty; R_1 \leftarrow P$ 
3   for  $i \leftarrow n - 1 \dots 0$  do
4      $R_0 \leftarrow 2R_0$ 
5     if  $k_i = 1$  then
6        $R_0 \leftarrow R_0 + R_1$ 
7   return  $R_0$ 
```

---

Este método utiliza la representación binaria del escalar  $k = \sum_{i=0}^{n-1} k_i 2^i$  para realizar la multiplica-

ción, tal que:

$$[k]P = (k_{n-1}2^{k-1} + \dots + k_02^0)P$$

Luego utilizando la regla de Horner (GeeksforGeeks, s.f.) se obtiene

$$[k]P = (k_0 + 2(k_1 + (\dots 2(k_{n-2} + 2(k_{n-1}))))))P$$

El método en cuestión requiere de  $n$  doblados (D) y un promedio de  $\frac{n}{2}$  adiciones (A), lo cual se denota como  $(\frac{n}{2})A + nD$ . Por lo tanto, el método de Double-and-Add es el método más básico y óptimo para realizar una multiplicación escalar (Cohen et al., 2005).

## 2.4. Longitud de Claves

ECC y el RSA son criptosistemas que utilizan el problema del logaritmo discreto para las multiplicaciones de grupo en un cuerpo finito. Curva elíptica aplica este problema a un grupo de puntos, mientras que el RSA utiliza la factorización de números enteros muy grandes. Además, ambos criptosistemas ofrecen el mismo nivel de seguridad, con la principal diferencia que ECC ocupa una longitud de clave más corta que el RSA (Giry, & Quisquater, 2011). Esta se puede apreciar en la Tabla 2.1

Nivel de seguridad	80	112	128	192	256
ECC	160	224	256	384	512
RSA	1024	2048	3072	8192	15360

Tabla 2.1: Comparación de la cantidad de bits requeridas en RSA y ECC para diferentes niveles de seguridad

En particular, este escrito trabajará con una clave de largo de 256-bits para ECC.

## 2.5. Sistemas de Coordenadas

Dada la ecuación de Weierstrass la cual define una curva elíptica  $E$  en un cuerpo primo  $\mathbb{F}_p$ . Se denotan las coordenadas afines como el conjunto de puntos  $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \cup \{P_\infty\}$  tal que satisfagan la ecuación de Weierstrass.

Las operaciones presentadas anteriormente para el doblado y la adición, requieren de Inversiones ( $I$ ), Multiplicaciones ( $M$ ) y Cuadrados ( $S$ ) para ser computadas, más precisamente el doblado requiere de  $I + 2M + 2S$  y la adición requiere de  $I + 2M + S$ . Sin embargo, realizar una inversión es muy costoso, ya que  $1I = 100M$ , (Cohen et al., 2005). Para evitar esta inversión se puede utilizar otros sistemas de coordenadas, las Proyectivas ( $\mathcal{P}$ ) y las Jacobianas ( $\mathcal{J}$ ).

Entonces, sea  $\mathbb{F}_p$  un cuerpo finito, y sea  $c$  y  $d$  dos enteros positivos. Se define una relación de equivalencia denotada por  $\sim$  en el espacio  $\mathbb{F}_p^3 \setminus (0, 0, 0)$  tal que se obtenga una tripleta sobre  $\mathbb{F}_p$  dada por:

$$(X_1, Y_1, Z_1) \sim (X_2, Y_2, Z_2)$$

Si  $X_1 = \lambda^c X_2$ ,  $Y_1 = \lambda^d Y_2$ ,  $Z_1 = \lambda Z_2$  para algún  $\lambda \in \mathbb{F}_p$ . Luego, la clase de equivalencia que contiene  $(X, Y, Z) \in \mathbb{F}_p^3$  esta dada por:

$$(X : Y : Z) = \{(\lambda^c X, \lambda^d Y, \lambda Z) : \lambda \in \mathbb{F}_p\}$$

Se utilizará esta forma para definir las coordenadas Proyectivas ( $\mathcal{P}$ ) y Jacobianas ( $\mathcal{J}$ ) basados en los valores de  $c$  y  $d$ .

### 2.5.1. Coordenadas Proyectivas $\mathcal{P}$

Si se selecciona  $c = 1$  y  $d = 1$ , entonces se denotará  $x = X/Z$  y  $y = Y/Z$ , con  $Z \neq 0$ , se obtiene la ecuación de Weierstrass Proyectiva de una curva elíptica  $E$ :

$$Y^2 Z = X^3 + aXZ^2 + bZ^3$$

Donde cada punto afín  $(x, y)$  se representa por una coordenada Proyectiva  $(\lambda x : \lambda y : \lambda)$ , con  $\lambda \in \mathbb{F}_p$  y  $\lambda \neq 0$ . Además, cada punto representado por  $(X : Y : Z)$ ,  $Z \neq 0$  tiene una coordenada afín de la forma  $(x, y) = (X/Z, Y/Z)$ . El opuesto de un punto  $(X : Y : Z)$  es  $(X : -Y : Z)$  y el punto al infinito ( $P_\infty$ ) esta dado por  $(0 : \lambda : 0)$ ,  $\lambda \in \mathbb{F}_p$ .

### 2.5.2. Coordenadas Jacobianas $\mathcal{J}$

Sea  $c = 2$  y  $d = 3$ , del mismo modo que el anterior, se denotará  $x = X/Z^2$  e  $y = Y/Z^3$ , con  $Z \neq 0$ , entonces se obtiene la ecuación de Weierstrass Jacobiana de una curva elíptica  $E$ :

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

Cada punto afín  $(x, y)$  se representa por una coordenada Jacobiana de la forma  $(\lambda^2 x : \lambda^3 y : \lambda)$ , con  $\lambda \in \mathbb{F}_p$  y  $\lambda \neq 0$ . Además, cada punto representado por  $(X : Y : Z)$ ,  $Z \neq 0$ , tiene una coordenada afín  $(x, y) = (X/Z^2, Y/Z^3)$ . El opuesto del punto  $(X : Y : Z)$  es  $(X : -Y : Z)$  y el punto al infinito esta dado por  $(\lambda^2 : \lambda^3 : 0)$ , con  $\lambda \in \mathbb{F}_p$ .

## 2.6. Fórmulas Explícitas Jacobianas

Estas fórmulas permiten realizar las operaciones de grupo dentro de las coordenadas Jacobianas, estas operaciones son el doblado Jacobiano  $2P$ , la adición Jacobiana  $P + Q$  (Menezes, Vanstone, & Hankerson, 2004). Particularmente en este escrito se utilizará un  $a = -3$  ya que así se puede optimizar la cantidad de operaciones (Brier, & Joye, 2003) para las implementaciones de las contramedidas.

### 2.6.1. Doblado Jacobiano

Dado un punto en coordenadas Jacobianas  $P = (X_1, Y_1, Z_1)$  se puede obtener un  $2P = (X_3, Y_3, Z_3)$  mediante:

$$\begin{aligned} a &= 3(X_1^2 - Z_1^4) & b &= 4X_1Y_1^2 & X_3 &= a^2 - 2a \\ Y_3 &= a(b - X_3) - 8Y_1^4 & Z_3 &= 2Y_1Z_1 \end{aligned}$$

con un costo total de 4 multiplicaciones y 4 adiciones.

### 2.6.2. Adición Jacobiana

Dado dos puntos en coordenadas Jacobianas  $P = (X_1, Y_1, Z_1)$  y  $Q = (X_2, Y_2, Z_2)$ , se puede obtener  $P + Q = (X_3, Y_3, Z_3)$  con un costo de 12 multiplicaciones y 4 adiciones.

$$\begin{aligned} a &= Z_1^3Y_2 - Z_2^3Y_1 & b &= Z_1^2X_2 - Z_2^2X_1 & Z_3 &= Z_1Z_2b \\ X_3 &= a^2 - b^3 - 2Z_2^2X_1b^2 & Y_3 &= a(Z_2^2X_1b - X_3) - Z_2^3Y_1b^3 \end{aligned}$$



## Capítulo 3

# Side Channel Attack en ECC

Los Side Channel Attack (SCA), son ataques no invasivos los cuales buscan explotar las fugas físicas de información que existen tanto en las implementaciones algorítmicas como en los dispositivos utilizados para encriptar información. La Figura 3.1 muestra un esquema de un SCA.

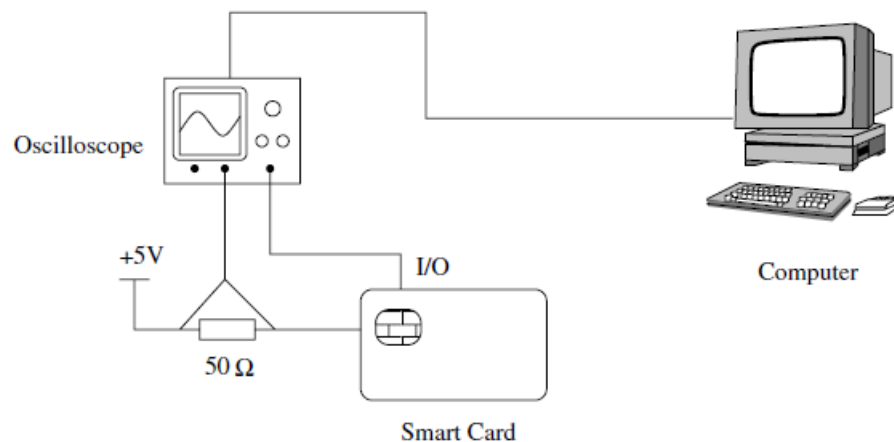


Figura 3.1: Esquema de un SCA

Fuente: Cohen et al., (2005)

En general existen varios tipos de ataque tales como el Simple Power Analysis (Joye & Olivier, 2011), el Differential Power Analysis (Kocher, Jaffe, & Jun, 1999), el Address-Bit DPA (Itoh, Izu, & Takenaka, 2002), los Ataques Múltiples (Goubin, 2002; Akashita, & Takagi, 2005; Murdica, Guilley, Danger, Hoogvorst, & Naccache, 2012), entre otros que no se utilizarán dentro de este escrito como los Fault Attack o ataques que modifiquen el cuerpo.

### 3.1. Simple Power Analysis

En los tipos de SCA, el más básico es el Simple Power Analysis (SPA). Este ataque consiste en observar directamente la energía consumida por el algoritmo durante la ejecución, entonces mediante el comportamiento que se muestra durante la ejecución se puede obtener la información necesaria para descubrir la clave secreta (Cohen et al., 2005; Joye & Olivier, 2011). En la Figura 3.2 se muestra gráficamente un SPA.

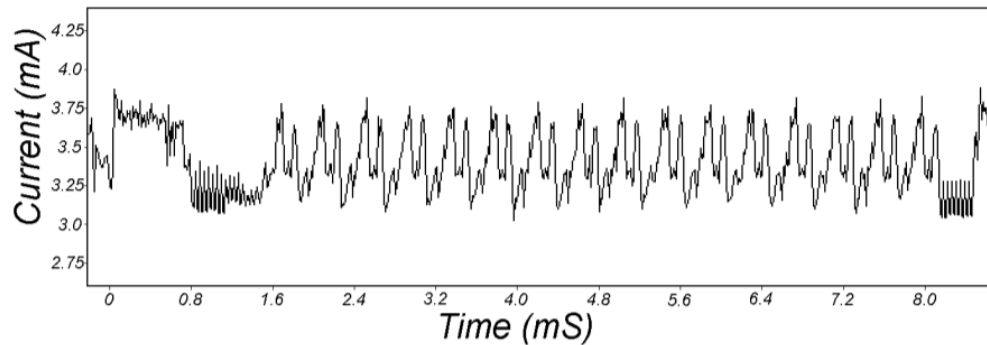


Figura 3.2: Gráfica de energía de un SPA

Fuente: Kocher, Jaffe, & Jun (1999)

### 3.2. Differential Power Analysis

Los Differential Power Analysis (DPA) son ataques que utilizan un análisis estadístico para obtener la información, dada una larga muestra de trazas obtenidas durante una encriptación controlada. En general este tipo de ataque necesita muchas ejecuciones, sin embargo, después de procesar y realizar el análisis estadístico, el DPA puede generar una reconstrucción completa de la clave secreta en pocos minutos (Kocher, Jaffe, & Jun, 1999; Caddy, 2011).

### 3.3. Address-Bit DPA

Itoh, Izu y Takenaka (2002) presentaron el Address-Bit DPA este explota las fugas de información que entrega cada bit individual de la dirección de memoria o los registros temporales. Por ejemplo, el algoritmo Modified-Montgomery Ladder presentado por Joye y Yen (2002), en el cual la dirección de memoria del punto doblado solo depende de un  $k_i$ . Como resultado, el  $k_i$  puede ser recuperado si el atacante puede distinguir la información desde  $R_0$  y  $R_1$ , donde  $R_0$  y  $R_1$  son registros temporales.

### 3.4. RPA, ZPA y SVA

Dentro de la gama de ataques hay algunos que intentan deducir la información encriptada mediante suposiciones o particularidades que se dan durante la ejecución del algoritmo. Estos 3 ataques suponen que es posible pedir al dispositivo de operar la multiplicación escalar (usando un escalar secreto constante) con un punto que será dado por el atacante.

#### 3.4.1. RPA

Goubin (2002) propuso un nuevo ataque el cual se conoce como Refined Power Analysis (RPA), este ataque se basa en que si el atacante conoce el bit más significativo del escalar secreto  $k$  y se utiliza un “punto especial” en la curva elíptica  $E(\mathbb{F}_p)$  de tal modo que una de las coordenadas del punto es 0, entonces él puede hacer una suposición del siguiente bit y mandar un punto para la multiplicación escalar que producirá la coordenada 0 si la suposición es correcta. Si esto ocurre, el punto utilizado dejara una diferencia significativa en la traza de energía consumida. Por lo que el atacante podrá conocer los bits del escalar secreto  $k$  después de varias ejecuciones recursivas.

#### 3.4.2. ZPA

Akishita y Takagi (2003; 2004) presentaron una generalización del ataque de Goubin teniendo como idea principal que el ataque utilice puntos especiales, tal que los valores de los registros temporales toman el valor cero, en particular para puntos  $P = (x, y)$  que satisfagan  $3x^2 + a = 0$  o  $5x^4 + 2ax^2 - 4bx + a^2 = 0$  tal que no puedan ser aleatorizados por coordenadas proyectivas o curvas elípticas isomorfas o cuerpos isomorfos. El ataque depende de cómo se implemente la fórmula de la adición, ya que Akashita uso el registro zero-value para la formula del doblado. A este ataque se le conoce como Zero-Value Point Attack (ZPA).

#### 3.4.3. SVA

Murdica, Guilley, Danger, Hoogvorst y Naccache (2012) presentaron un ataque llamado Same Value Analysis (SVA), la lógica de este ataque se basa en buscar puntos que sean iguales en su valor durante el doblado o la adición, luego internamente utiliza un análisis de poder comparativo para detectar si el punto especial aparece durante la multiplicación escalar.

## Capítulo 4

# Contramedidas para Side Channel Attack

Como se vio en el capítulo anterior, en el Criptosistema de Curva Elíptica existen diferentes tipos de ataques que explotan fugas específicas que tiene el hardware o bien el mismo algoritmo para realizar la encriptación. Como el objetivo de la criptografía es proteger la información, cada tipo de ataque tiene contramedidas que impide al atacante obtener dicha información encriptada. Los algoritmos  $[k]P$  son sencillos de computar para los Simple Power Analysis, esto es debido a que la operación de adición y de doblado son en esencia distintos (ya que poseen costos computacionales distintos). Sin embargo, hay contramedidas que son eficientes para este tipo de ataque, pero solo si se aplican a curvas elípticas especiales, como las de Edwards (Bernstein & Lange, 2007; Bessalov & Kovalchuk, 2019), Curvas de Jacobi (Billet & Joye, 2003), Curvas de Weierstrass (Brier & Joye, 2002). En general estas contramedidas son eficientes para encriptar, aun así, al trabajar en un cuerpo primo grande se recomienda el uso de curvas elípticas de orden primo (Brown, 2010; NIST, 2013). Para evitar ataques se puede utilizar instrucciones que siempre repitan el mismo patrón de sumas y doblados. Por ejemplo la Fórmula Unificada de Brier-Joye (2002), el Always Double and Add de Coron (1999), Montgomery Ladder (Izu & Takagi, 2002), el Double-Add y Add-Only de Joye (2007), el Signed Digit Method (Goundar, Joye, Miyaji, & Rivain, 2011) y el Bloque Atómico (Chevallier-Mames, Ciet, & Joye, 2004; Abarzúa & Thériault, 2012) los cuáles serán trabajados en este escrito.

## 4.1. Contramedidas para Simple Power Analysis

### 4.1.1. Fórmula Unificada de Brier-Joye

La Fórmula Unificada usa el mismo cuerpo de operaciones para la adición y el doblado de un punto. Para las curvas de Weierstrass, se tiene el siguiente algoritmo para la adición (Brier & Joye, 2002). El cual tiene un costo computacional de 13 Multiplicaciones (M) + 5 Cuadrados (S). Dado dos puntos en coordenadas Proyectivas  $P = (X_1, Y_1, Z_1)$  y  $Q = (X_2, Y_2, Z_2)$ , entonces la operación  $P + Q = R = (X_3, Y_3, Z_3)$  esta dada por:

$$\begin{aligned} U_1 &= X_1 Z_1, & U_2 &= X_2 Z_1, & S_1 &= Y_1 Z_2, & S_2 &= Y_2 Z_1, \\ T &= U_1 + U_2, & M &= S_1 + S_2, & Z &= Z_1 Z_2, & F &= ZM, \\ L &= MF, & G &= TL, & R &= T^2 - U_1 U_2 + aZ^2, & W &= R^2 - G, \\ X_3 &= 2FW, & Y_3 &= R(G - 2W) - L^2, & Z_3 &= 2F^3. \end{aligned}$$

Si bien este algoritmo permite ocultar la operación de adición y de doblado también es susceptible a los ataques de Izu y Takagi (2003), al ataque de Walter (2004), el ataque de Amiel, Feix, Tunstall, Whelan y Marnane (2008), al Ataque Combinado Pasivo-Activo (PACA) (Amiel, Villegas, Feix, & Marcel, 2007), al Horizontal Collision Correlation Analysis (HCCA) (Bauer, Jaulmes, Prouff, Reinhard, & Wild, 2014) y a los SVA.

### 4.1.2. Always Double and Add de Coron

Este algoritmo propuesto por Coron (1999) se basa en la idea de siempre hacer un doblado y una adición en cada iteración sin utilizar una condición que indique cuando realizar una adición, de este modo no se puede suponer si el bit  $k_i$  que se esta trabajando es un 0 o 1.

---

#### Algorithm 2: Always Double and Add de Coron

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

---

```

1 begin
2    $R_0 \leftarrow P_\infty$ 
3   for  $i \leftarrow n - 1 \dots 0$  do
4      $R_0 \leftarrow 2R_0$ 
5      $R_1 \leftarrow R_0 + P$ 
6      $R_0 \leftarrow R_{k_i}$ 
7   return  $R_0$ 
```

---

Este algoritmo otorga una protección contra los SPA, sin embargo, es posible vulnerar su seguridad si se le aplica un ZPA (Akashita & Takagi, 2003), RPA (Goubin, 2002), Doubling Attack

(Fouque & Valette, 2003), C-Safe Fault Attack (Yen, & Joye, 2000), M-Safe Fault Attack (Yen, Kim, Lim, & Moon, 2002), Ataques 2-Torsion para cuerpos de característica 2 (Yen, Lien, Moon, & Ha, 2005) y Correlation Collision Attack on the Horizontal Setting (Hanley, Kim, & Tunstall, 2015).

#### 4.1.3. Montgomery Ladder de Brier-Joye

Montgomery (1987) propuso un algoritmo que se conoce como Montgomery Ladder (ML) el cual está diseñado para un tipo especial de curvas llamada Curvas de Montgomery. Así como el Always Double and Add, este se asegura que cada bit del escalar realice ambas operaciones de doblado y adición, pero con una condición suplementaria para ambas operaciones tengan un impacto en la salida final. Brier y Joye (2002) utilizaron esta idea con las Curvas de Weierstrass. Más tarde se generalizó esta idea para las curvas de Weierstrass definidas en diferentes cuerpos (Brier & Joye, 2002; Goundar, Joye, Miyaji, & Rivain, 2011; López & Dahab, 1999), y para la multiplicación escalar Right-to-Left (Joye & Yen, 2002). El costo computacional es  $9M + 2S$  para el algoritmo de la adición y  $6M + 3S$  para el algoritmo del doblado. Debido a que el ML clásico es susceptible a ataques (Joye & Yen, 2002; Yen, Kim, Lim, & Moon, 2001) Joye y Yen (2002) propusieron una modificación al Montgomery Ladder el cual brinda una protección natural contra los SPA y los Safe-Error Attacks.

---

##### Algorithm 3: Modified Montgomery Ladder

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$   
**Output:**  $Q = [k]P$

```

1 begin
2    $R_0 \leftarrow P_\infty; R_1 \leftarrow P$ 
3   for  $i \leftarrow n - 1 \dots 0$  do
4      $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$ 
5      $R_{k_i} \leftarrow 2R_{k_i}$ 
6   return  $R_0$ 
```

---

En general existen varias versiones de este algoritmo, las cuales tienen distintos costos computacionales de Multiplicaciones (M), Inversiones (I) y Cuadrados (S). En estas, están el Montgomery Ladder de Brier-Joye (2002) el cual tiene un costo computacional de  $n(12M + 13S) + 1I + 3M + 1S$ . Esta el  $X$ -only Montgomery Ladder (Izu & Takagi, 2002; Brier & Joye, 2002) el cual solo utiliza el registro  $X$  del punto  $P$  y tiene un costo computacional de  $n(9M + 7S) + 1I + 14M + 3S$ . Y el  $(X, Y)$ -only Co-Z Montgomery Ladder (Goundar, Joye, Miyaji, & Rivain, 2011) el cual tiene un costo de  $n(8M + 6S) + 1I + 1M$ . Si bien este algoritmo permite una protección contra los SPA, es vulnerable contra los RPA, ZPA, Address-bit DPA y a los Doubling Attack (Yen, Ko, Moon, & Ha, 2006)

#### 4.1.4. Double-Add de Joye

El algoritmo Double-Add de Joye (2007) al igual que el Montgomery Ladder para el Right-to-Left, siempre repite el mismo patrón de operaciones. En el siguiente Algoritmo de muestra el Double-Add de Joye resistente contra SPA.

---

**Algorithm 4:** Double-Add de Joye resistente contra SPA

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```

1 begin
2    $R_0 \leftarrow P_\infty$ 
3    $R_1 \leftarrow P$ 
4   for  $i \leftarrow 0 \dots n - 1$  do
5      $R_{1-k_i} \leftarrow 2R_{1-k_i} + R_{k_i}$ 
6   return  $R_0$ 

```

---

El costo computacional de este algoritmo es  $n(13M + 8S) + 1I + 3M + 1S$ . Por el momento no se ha encontrado literatura respecto a un ataque para esta contramedida.

#### 4.1.5. Add-Only de Joye

El siguiente algoritmo propuesto por Joye (2007) consiste en repetir siempre el mismo patrón de adiciones, yendo del bit menos significativo al bit más significativo. Debido a que este algoritmo utiliza la adición como doblado para cada bit de la clave secreta es resistente contra ataques SPA. Además, el costo computacional del algoritmo es  $(2n)A$ .

---

**Algorithm 5:** Add-Only de Joye

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```

1 begin
2    $R_0 \leftarrow P_\infty$ ;  $R_1 \leftarrow P$ ;  $R_2 \leftarrow P$ 
3   for  $i \leftarrow 0 \dots n - 1$  do
4      $R_{1-k_i} \leftarrow R_{1-k_i} + R_2$ 
5      $R_2 \leftarrow R_0 + R_1$ 
6   return  $R_0$ 

```

---

Este algoritmo es vulnerable a un tipo de ataque conocido como Correlation Collision Attack on the Horizontal Setting (Hanley, Kim, Tunstall, 2015). Sin embargo, se propuso una contramedida (Le, Tan, & Tunstall, 2015) para este ataque, la cual se muestra en el algoritmo 6.

---

**Algorithm 6:** Randomized Montgomery Ladder

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```
1 begin
2    $b \xleftarrow{\text{Random}} \{0, 1\}, R_0 \leftarrow P_\infty$ 
3   if  $b = 0$  then
4      $R_1 \leftarrow P$ 
5   else
6      $R_1 \leftarrow P_\infty$ 
7   for  $i \leftarrow n - 1 \dots 0$  do
8     if  $b \oplus k_i = 1$  then
9        $b \xleftarrow{\text{Random}} \{0, 1\}$ 
10       $R_b \leftarrow R_0 + R_{b \oplus k_i}$ 
11       $R_{-b} \leftarrow R_b + P$ 
12  return  $R_0$ 
```

---

#### 4.1.6. Signed Digit Method

Este algoritmo permite prevenir los ataques SPA utilizando la expansión Zeroless Signed-Digit (ZSD) tal que, sea  $k$  un número impar, entonces se puede expresar con valores  $\{-1, 1\}$ . Esta idea fue presentada por Goundar, Joye, Miyaji, Rivain y Venelli (2011). El Algoritmo 7 muestra el Signed Digit Method aplicado al algoritmo Left-to-Right.

---

**Algorithm 7:** Signed Digit Method: Left-to-Right

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$  con  $k_0 = 1$

**Output:**  $Q = [k]P$

```
1 begin
2    $R_0 \leftarrow P; R_1 \leftarrow P$ 
3   for  $i \leftarrow n - 1 \dots 1$  do
4      $d \leftarrow (-1)^{1+k_i}$ 
5      $R_0 \leftarrow 2R_0 + (d)R_1$ 
6  return  $R_0$ 
```

---

Por el momento no se ha encontrado literatura sobre ataques a esta contramedida.

#### 4.1.7. Bloques Atómicos

En las contramedidas existentes contra los SPA, los Bloques Atómicos (Chevallier-Mames, Ciet, & Joye, 2004) consisten en particionar las operaciones de doblado y adición en bloques atómicos homogéneos, tal que no puedan ser distinguibles en un SPA. Esto debido a que el atacante no podrá saber que bloque es parte de la operación doblado o adición. Esta originalmente tiene la estructura de Multiplicación-Adición-Negación-Adición (*M.A.N.A*) en las operaciones del cuerpo primo. Además, los bloques atómicos asumían que las multiplicaciones ( $M$ ) son iguales



a los cuadrados ( $S$ ) desde la perspectiva de un SCA. Sin embargo, esto fue refutado por Amiel et. al. (2008) y Hanley, Tunstall y Marnane (2011). Debido a esto, surgió un concepto conocido como Hamming weight, el cual se utiliza para distinguir entre los bloques atómicos que contienen multiplicaciones y cuadrados, lo que reabrió la rama de los SCA para bloques atómicos. Por otro lado, esta distinción puede tener ciertos beneficios, ya que los cuadrados son menos costosos que las multiplicaciones (Giraud & Verneuil, 2010). Para poder adaptar las fórmulas existentes a la estructura de los bloques atómicos, Bernstein y Lange (2007) introdujeron una nueva metodología más flexible. Esta metodología permite modificar operaciones de punto para balancear la cantidad de cuadrados y multiplicaciones, lo cual facilitaría la presencia de los cuadrados ( $S$ ) en los bloques atómicos. Por otro lado, Longa y Miri (2008) presentaron una nueva estructura de bloque atómico basado en la secuencia  $(S, N, A, M, N, A, A)$ . Además, aplicaron esta estructura al doubling, tripling y la adición mixta en curvas elípticas con coordenadas Jacobianas. En 2012, Abarzúa y Thériault presentaron un nuevo conjunto de bloques atómicos para proteger las multiplicaciones escalares contra SPA y C-Safe Attack. La estructura del bloque atómico sigue la secuencia  $(S, N, A, A, M, A)$ . Además, estos bloques atómicos funcionan para el doubling, tripling, quintupling y la adición mixta.

A continuación, se presenta el doblado de un punto en coordenadas Jacobianas y la adición mixta en coordenadas Jacobiana-Afín: Sea  $P = (X_1, Y_1, Z_1)$  un punto con coordenadas Jacobianas en una curva elíptica  $E$ . La fórmula más eficiente (Abarzúa & Thériault, 2012) con salida en coordenadas Jacobianas requiere de  $4M + 4S$ . El resultado de los bloques atómicos, con entrada  $R_1 \leftarrow X_1, R_2 \leftarrow Y_1, R_3 \leftarrow Z_1$  y salida  $X_3 \leftarrow R_1, Y_3 \leftarrow R_2, Z_3 \leftarrow R_3$ , se puede ver a continuación:

Secuencia	Bloque 1	Bloque 2	Bloque 3	Bloque 4
S	$R_4 \leftarrow R_3^2$ $Z_1^2$	$R_6 \leftarrow R_2^2$ $Y_1^2$	$R_4 \leftarrow R_1^2$ $a^2$	$R_8 \leftarrow R_7^2$ $4Y_1^4$
N	$R_5 \leftarrow -R_4$ $-Z_1^2$	$R_7 \leftarrow -R_1$ $-X_1$	$R_5 \leftarrow -R_1$ $-a$	$R_2 \leftarrow -R_8$ $-4Y_1^4$
A	$R_6 \leftarrow R_1 + R_4$ $X_1 + Z_1^2$	$R_1 \leftarrow R_7 + R_7$ $-2X_1$	$R_8 \leftarrow R_6 + R_6$ $-2b$	$R_8 \leftarrow R_1 + R_6$ $X_3 - b$
A	$R_4 \leftarrow R_1 + R_5$ $X_1 - Z_1^2$	$R_7 \leftarrow R_6 + R_6$ $2Y_1^2$	$R_1 \leftarrow R_4 + R_8$ $X_3 = a^2 - 2b$	$R_4 \leftarrow R_2 + R_2$ $-8Y_1^4$
M	$R_5 \leftarrow R_6 R_4$ $X_1^2 - Z_1^4$	$R_6 \leftarrow R_1 R_7$ $-b$	$R_4 \leftarrow R_2 R_3$ $Y_1 Z_1$	$R_6 \leftarrow R_5 R_8$ $-a(X_3 - b)$
A	$R_4 \leftarrow R_5 + R_5$ $2(X_1^2 - Z_1^4)$	$R_1 \leftarrow R_5 + R_4$ $a$	$R_3 \leftarrow R_4 + R_4$ $Z_3 = 2Y_1 Z_1$	$R_2 \leftarrow R_6 + R_4$ $Y_3$

Tabla 4.1: Bloque Atómico para Doblado Jacobiano

Por otro lado, también esta la adición mixta en coordenadas Jacobiana-Afín, dados dos puntos  $P = (X_1, Y_1, Z_1)$  en coordenada Jacobianas y  $Q = (X_2, Y_2)$  en coordenadas Afín, ambos pertenecientes a la curva elíptica  $E$ . La fórmula más eficiente para la adición mixta  $P + Q = (X_3, Y_3, Z_3)$  requiere de  $8M + 3S$ .

Los bloques atómicos presentados por Abarzúa y Thériault (2012) son susceptibles al Horizontal Collision Correlation Analysis (Bauer, Jaulmes, Prouff, Reinhard, & Wild, 2014), a los Vertical Collision Correlation Analysis presentados por Murdica (2014), a los Doubling Attack (Fouque, & Valette, 2003) y a un ataque propuesto por Chen, Li, Wu y Yu (2009).

## 4.2. Contramedidas para Differential Power Analysis

A continuación, se mostrarán algunas contramedidas para los Differential Power Analysis

### 4.2.1. Primera Contramedida de Coron

En esta contramedida, Coron (1999) utiliza la cardinalidad de la curva  $\#E$  para computar un  $Q = [k]P$  de la siguiente manera. Primero seleccionar un número  $d$  aleatorio con un largo de  $n$  bits. Luego se computa un  $k' = k + d(\#E)$ . Y, finalmente realiza un  $Q = [k']P$ . El truco de esta contramedida esta en que  $[k']P = [k + d(\#E)]P = [k]P + [d(\#E)]P = [k]P$  ya que  $[d(\#E)]P = P_\infty$ . Sin embargo esta contramedida presenta problemas de seguridad. Algunos de ellos son el análisis de Okeya y Sakurai (2000), el Doubling Attack (Fouque, & Valette, 2003) y el análisis de Ciet-Joye (2003), al Carry Leakage (Fouque, Réal, Valette, & Drissi, 2008) y al ataque de Feix, Roussette y Venelli (2014).

### 4.2.2. Contramedida de Clavier-Joye

Clavier y Joye (2001) presentaron esta contramedida la cual utiliza un número aleatorio  $r$  de  $n$ -bits. Luego, se calcula un  $k' = k - r$  y se computan las multiplicaciones escalares  $[r]P$  y  $[k']P$ , tal que se obtenga la multiplicación  $[k]P$  por medio de la suma de ellas. Esta contramedida es costosa debido a que realiza dos multiplicaciones escalares para calcular el  $[k]P$ .

Además, presenta problemas de seguridad, tales como el análisis de Ebeid (2007), el ataque de Muller y Valette (2006), el Carry Leakage Attack de Fouquet et al. (2008) y el análisis de Ha et al. (2007) utilizando el 2-Torsion Attack el cual solo funciona para cuerpos de característica 2.

### 4.2.3. Segunda Contramedida de Coron

Coron (1999) definió esta contramedida para la multiplicación escalar  $[k]P$ . Esta contramedida se basa en realizar una multiplicación escalar previa de un punto  $R$  aleatorio perteneciente a la curva el cual se define como  $S' = [k]R$ , luego se realiza una adición tal que se pueda conseguir la multiplicación  $k(P + R)$  y se resta el  $S'$  al resultado. Esta contramedida es eficaz contra los RPA, ZPA y SVA.

---

**Algorithm 8:** Segunda Contramedida de Coron: Blinding Point

---

**Input:** Punto  $P$  y punto  $R \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$  y  $S' = [k]R$

**Output:**  $Q = [k]P$

```
1 begin
2    $P \leftarrow P + R$ 
3    $S = \text{Double-and-Add}(P, k)$ 
4    $b \xleftarrow{\text{Random}} \{0, 1\}$ 
5   return  $S - S'$ 
```

---

En esta contramedida el  $S'$  y el  $R$  pueden ser actualizados para una nueva ejecución de la siguiente forma  $R = (-1)^b 2R$  y  $S' = (-1)^b 2S'$ , donde  $b$  es el bit aleatorio obtenido en el proceso. Esta contramedida es susceptible al Doubling Attack (Fouque, & Valette, 2003), al análisis de Okeya y Sakurai (2000).

#### 4.2.4. Tercera Contramedida de Coron

Esta tercera contramedida propuesta por Coron (1999), consiste en aleatorizar el punto  $P = (X, Y, Z)$  con un  $\lambda \in \mathbb{F}_p$ , con  $\lambda \neq 0$ , tal que  $P = (\lambda X, \lambda Y, \lambda)$  al utilizar coordenadas Proyectivas y  $P = (\lambda^2 X, \lambda^3 Y, \lambda)$  para las coordenadas Jacobianas. Luego para el cómputo de la multiplicación escalar  $Q = [k]P$  se utiliza este nuevo  $P$  aleatorizado. Los costos computacionales asociados a esta contramedida son bajos, ya que son requeridas  $3M$  para las coordenadas Proyectivas y  $4M + 1S$  para las coordenadas Jacobianas.

Esta contramedida es susceptible a los RPA, SVA y ZPA.

#### 4.2.5. Método 2P\* de Ciet-Joye

En 2003, Ciet y Joye presentaron una contramedida para los DPA, en la cual aplicaron el método de aleatorización propuesto por Coron (1999) al Algoritmo 1: Left-to-Right. La idea de esta contramedida es aleatorizar el  $[2]P$  utilizando coordenadas proyectivas. Esto permite seguir utilizando el  $P$  en coordenadas afín. Y por ende, computar la multiplicación escalar  $[k]P$  usando coordenadas mixtas. A continuación, en el Algoritmo 9 se muestra el pseudocódigo del método

2P\*.

---

**Algorithm 9:** Método 2P\*

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$   
**Output:**  $Q = [k]P$

```

1 begin
2    $P^* \leftarrow \Upsilon(P)$  [base-point randomization]
3    $R_0 \leftarrow [2]P^*$ 
4   for  $i \leftarrow n - 2 \dots 1$  do
5      $b \leftarrow \neg k_i$ 
6      $R_b \leftarrow R_b + P$ 
7      $R_0 \leftarrow [2]R_0$ 
8    $b \leftarrow \neg k_i$ 
9    $R_b \leftarrow R_b + P$ 
10  return  $\Upsilon^{-1}(R_0)$ 

```

---

A la fecha no se han presentado ataques a esta contramedida.

### 4.3. Contramedidas para Address-bit DPA

#### 4.3.1. Montgomery Powering Ladder

Inicialmente esta contramedida fue propuesta por Itoh, Izu y Masahiko (2003), pero Izumi, Ikegami, Sakiyama y Ohta (2010) observaron que existía una diferencia de energía cuando se comparaban dos trazas de energía, esta diferencia se daba cuando la sobreescritura de una data era igual o distinta. Si la data era igual se podía observar un consumo bajo de energía, lo cual implica que la dirección de memoria  $a$  sea igual a la dirección de memoria  $b$ . A continuación, en el Algoritmo 10 se muestra la contramedida propuesta por Izumi et al.

---

**Algorithm 10:** Montgomery Powering Ladder method with randomized address

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2$  y  $r = (r_{n-1}, \dots, r_1, r_0)_2 \in \mathbb{N}$   
**Output:**  $Q = x([k]P)$

```

1 begin
2    $R_{r_{n-1}} \leftarrow x(2P)$ 
3    $R_{1 \oplus r_{n-1}} \leftarrow x(P)$ 
4   for  $i \leftarrow n - 2 \dots 0$  do
5      $R_2 \leftarrow 2(R_{k_{i+1} \oplus k_i \oplus r_{i+1}})$ 
6      $R_{1 \oplus r_i} \leftarrow R_0 + R_1$ 
7      $R_{r_i} \leftarrow R_2$ 
8   return  $R_{k_0 \oplus r_0}$ 

```

---

El problema que Izumi encontró es que en la operación  $R_1 \leftarrow R_{1+(k_i \oplus r_i)}$  del algoritmo de Itoh (2003) se puede establecer una relación del  $i$ -ésimo bit del escalar  $k$  con el bit  $i$ -ésimo del número aleatorio  $r$  tal que  $R_1 \leftarrow R_1$  si es que  $k_i = r_i$ . Si no  $R_1 \leftarrow R_0$ . Para comprobar esto Izumi realizó un ataque que analizaba la debilidad en el  $i$ -ésimo loop con la operación  $1 = R_1 \leftarrow R_{1+(k_i \oplus r_i)}$  y el comienzo del  $(i - 1)$ -ésimo loop con la operación  $2 = R_2 \leftarrow 2(R_{k_{i-1} \oplus r_i})$ . En la operación 1 se puede asumir que la traza de poder correspondiente a  $k_i \oplus r_i = 1$  esta en el grupo  $A = \{\text{las trazas}$

de poder que están por sobre el límite} y las trazas de poder que están relacionadas al  $k_{i-1} \oplus r_i = 0$  están en el grupo  $B = \{\text{las trazas de poder que están por debajo del límite}\}$ . El valor de la dirección  $k_{i-1} \oplus r_i$  del registro fuente  $R_{k_{i-1} \oplus r_i}$  en la operación 2 en el  $(i - 1)$ -esimo loop es calculado de la siguiente forma. Para el grupo  $A$ ,  $k_{i-1} \oplus r_i$  se convierte en  $\overline{k_{i-1} \oplus r_i}$  si es que  $r_i = \overline{k_i}$ . De otro modo,  $k_{i-1} \oplus r_i$  se convierte  $\overline{k_{i-1} \oplus k_i}$  si es que  $r_i = k_i$  en el grupo  $B$ . Como resultado el  $r_i$  se cancela al salir de la operación 2. Entonces es posible aplicar un DPA para calcular la diferencia de poder ( $P_\omega$ ) entre el grupo  $A$  y el  $B$ . Resultando  $P_\omega (1 \rightarrow \overline{k_{i-1} \oplus k_i}) - P_\omega (1 \rightarrow k_{i-1} \oplus k_i)$ . Ya que  $k_i$  y  $k_{i-1}$  son constantes, se puede distinguir si  $k_i$  es igual a  $k_{i-1}$  o no por medio de un Address-bit DPA (ADPA). Esta contramedida propuesta por Izumi et al. no presenta problemas de seguridad a la fecha.

## 4.4. Contramedidas para Ataques Múltiples

### 4.4.1. BRIP

Mamiya, Miyaji y Morimoto (2004) presentaron una contramedida resistente a ataques SPA, DPA, RPA, ZPA y SVA llamada BRIP. Este método utiliza un punto aleatorio  $R$  perteneciente a la curva, ya que este brinda una protección natural contra los DPA, RPA, ZPA y SVA, utilizando en principio del algoritmo Double-and-Add Always de Coron. El Algoritmo 11 computa un  $[k]P + R$  y al final de la ejecución resta el  $R$  a modo de obtener el  $[k]P$ .

---

#### Algorithm 11: BRIP

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$   
**Output:**  $Q = [k]P$

```

1 begin
2    $R \leftarrow \text{randompoint}(); R_0 \leftarrow R; R_1 \leftarrow -R; R_2 \leftarrow P - R$ 
3   for  $i \leftarrow n - 1 \dots 0$  do
4      $R_0 \leftarrow 2R_0$ 
5     if  $k_i = 0$  then
6        $R_0 \leftarrow R_0 + R_1$ 
7     else
8        $R_0 \leftarrow R_0 + R_2$ 
9   return  $R_0 + R_1$ 

```

---

Si bien esta contramedida protege de varios tipos de ataque, es susceptible a ataque como el 2-torsion propuesto por Yen, Lien, Moon y Ha (2005) y los Doubling Attacks (Fouque & Valette, 2003).

### 4.4.2. Contramedida de Kim

Esta contramedida (Kim et al., 2005) tiene la capacidad de proteger contra ataques como los DPA, RPA, SVA, ZPA, Doubling Attacks y 2-torsion Attacks. El concepto de esta contrame-

dida se basa en computar:  $[k]P = k(P + R) + (\#E - k)R = \sum_i k_i(P + R) + s_i R = \sum_i k_i P + \sum_i (k_i + s_i)R = [k]P + (k + r)R = [k]P + (\#E)R$ , donde  $R$  es un punto aleatorio y  $(\#E)R$  es el punto al infinito. El costo computacional para los precálculos es de  $2A + 1D$  y  $nD + nA$  para el computo de la multiplicación escalar  $[k]P$ . Para este algoritmo existe un caso particular donde se puede aplicar un SPA, esto es cuando en la línea 7 del Algoritmo 12, el  $k_i s_i = 00$  por lo que se tendría  $Q = Q + T_{k_i s_i} = Q + T_{00} = Q + P_\infty$ .

---

**Algorithm 12:** Contramedida de Kim

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```

1 begin
2    $R \leftarrow \text{randompoint}(); s \leftarrow \#E - k$ 
3    $T_{00} \leftarrow P_\infty; T_{01} \leftarrow R; T_{10} \leftarrow P + R; T_{11} \leftarrow P + 2R$ 
4    $Q \leftarrow T_{00}$ 
5   for  $i \leftarrow n - 1 \dots 0$  do
6      $Q \leftarrow 2Q$ 
7      $Q \leftarrow Q + T_{k_i s_i}$ 
8   return  $Q$ 
```

---

#### 4.4.3. Contramedida de Ha

Ha, Park, Moon y Yen (2007) presentaron una contramedida mejorada utilizando la técnica del mensaje blindado y el Truco de Shamir. La contramedida propuesta puede proteger de los SPA, DPA, Doubling Attack, RPA, SVA, ZPA, 2-torsion attack y los ADPA. La idea de esta contramedida es blindar un punto  $P$  utilizando un punto aleatorio  $R$ , ambos pertenecientes a la curva. Luego se asume que  $\#E$  es la cantidad de puntos de la curva  $E$  y tiene  $n$ -bits de largo. Entonces computando  $t(P + R) + sR + (2^n - 1)(P + R)$  en lugar de  $[k]P$ , con  $t$  y  $s$  enteros positivos de largo  $n$ -bits. Se obtendrá el  $[k]P$  por medio del siguiente computo:  $[k]P = (d\#E + k - (2^n - 1))(P + R) + (E - k)R + (2^n - 1)(P + R) = \sum_i^{n-1} 2^i (t_i(P + R)) + s_i R + (P + R)$ , donde  $(\#E)R = P_\infty$ . Sea  $t = d\#E + k - (2^n - 1)$  y  $s = \#E - k$ , ambos enteros de  $n$ -bits, entonces se escoge el entero más pequeño  $d$  tal que  $(d - 1)E + k < (2^n - 1) < dE + k$ , por esto  $d$  puede ser 1 o 2. A

continuación, en el Algoritmo 13 se muestra la contramedida propuesta.

---

**Algorithm 13:** Contramedida de Ha

---

**Input:** Punto  $P \in E(\mathbb{F}_p)$ ,  $k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$

**Output:**  $Q = [k]P$

```

1 begin
2    $t = d\#E + k - (2^n - 1); s = \#E - d$ 
3    $R \leftarrow \text{randompoint}(); u, v \xleftarrow{\text{random}} \{0, 1\}$ 
4    $T_{00 \oplus uv} \leftarrow P + R; T_{01 \oplus uv} \leftarrow P + 2R; T_{10 \oplus uv} \leftarrow 2P + 2R; T_{11 \oplus uv} \leftarrow 2P + 3R$ 
5    $Q = T_{t_{n-1}s_{n-1} \oplus uv}$ 
6   for  $i \leftarrow n - 2 \dots 0$  do
7      $Q \leftarrow 2Q$ 
8      $Q \leftarrow Q + T_{t_i s_i \oplus uv}$ 
9   return  $Q$ 

```

---

A la fecha esta contramedida no presenta ataques.

## Capítulo 5

# Implementación de Códigos

Los algoritmos previamente mencionados se implementaron en una arquitectura de 8-bits debido a que el chip que utilizan las tarjetas inteligentes suele tener un microprocesador de 8-bits. Es por esto que para implementar estas contramedidas se utilizó un Arduino UNO, ya que este es similar respecto a las características del chip de las tarjetas inteligentes. Además, para comprobar la correcta implementación de estos se utilizó la calculadora online Magma (<http://magma.maths.usyd.edu.au/calc/>) la cual permite trabajar con curvas elípticas y realizar multiplicaciones  $[k]P$ .

### 5.1. Implementación

#### 5.1.1. Librería Nano-ECC

Para su implementación fue necesario utilizar la librería Nano-ECC (Ryan, 2013) la cual permite trabajar con la aritmética de las curvas elípticas de los niveles de seguridad de 128, 192, 256 y 384 bits sugeridos por el estándar de NIST (2013). Esta librería aprovecha el tipo de dato byte del Arduino, el cual utiliza 8 bits de espacio, para crear un arreglo de tamaño 32 con tal de obtener el largo de 256 bits. Además, tiene operaciones que permiten la implementación del doblado y la adición Jacobiana bajo la idea de que se está trabajando con elementos de 256 bits.

La librería consta de 2 archivos, ECC.c y ECC.h. El primero contiene las funciones que permiten trabajar la aritmética de la curva, es decir, la adición, multiplicación, sustracción, inversión y reducción modular, esto permite realizar los algoritmos previamente mencionados. La segunda contiene las cabeceras de las funciones y las variables globales.

A continuación, en la Figura 5.1 se muestran algunas características de la librería. Particularmente se muestra que utilizando un nivel de 192 bits, toma alrededor de 4034ms ejecutar el ECDH en



un Atmega328P.

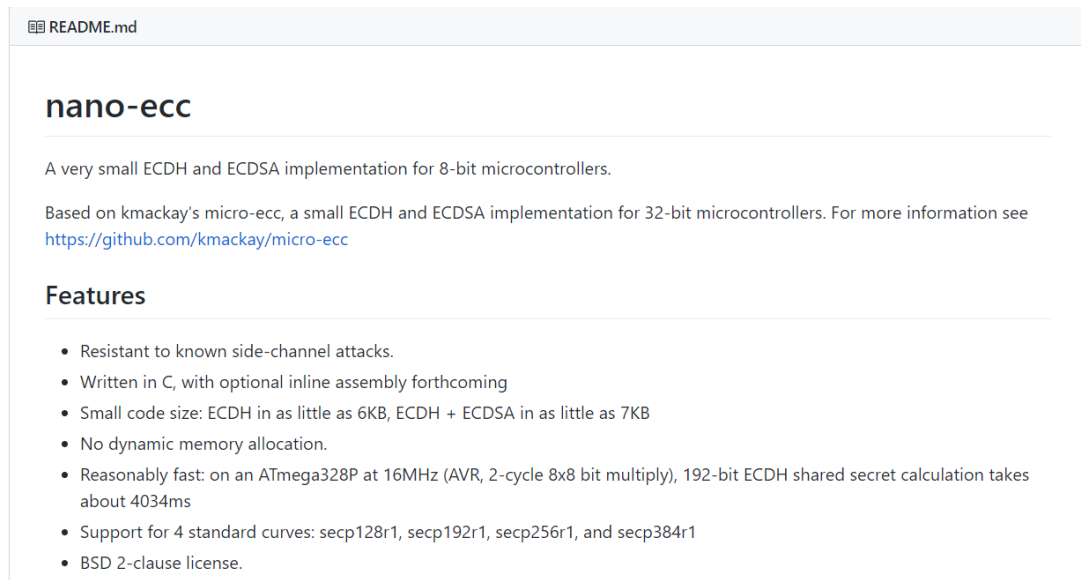
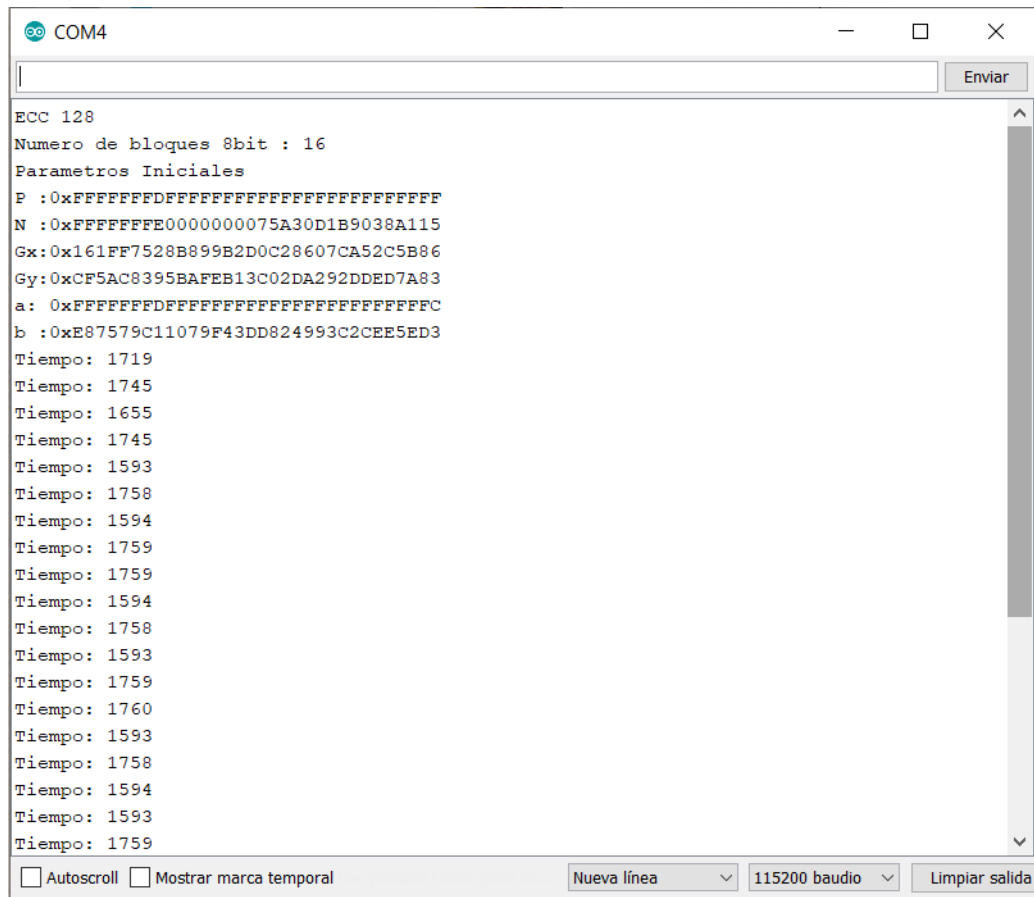


Figura 5.1: Características de NanoECC

Además, se realizaron algunas pruebas con esta librería ejecutando el Algoritmo 1 Left-to-Right con los distintos niveles de seguridad que ofrece la librería para corroborar el tiempo de ejecución que se muestra en la Figura 5.1.

En la Figura 5.2 se muestran los tiempos de ejecución del Algoritmo 1 Left-to-Right con un nivel de seguridad de 128 bits. Cabe recalcar que este nivel de seguridad ya no se utiliza y solo se utilizó como experimento para corroborar el tiempo de ejecución.



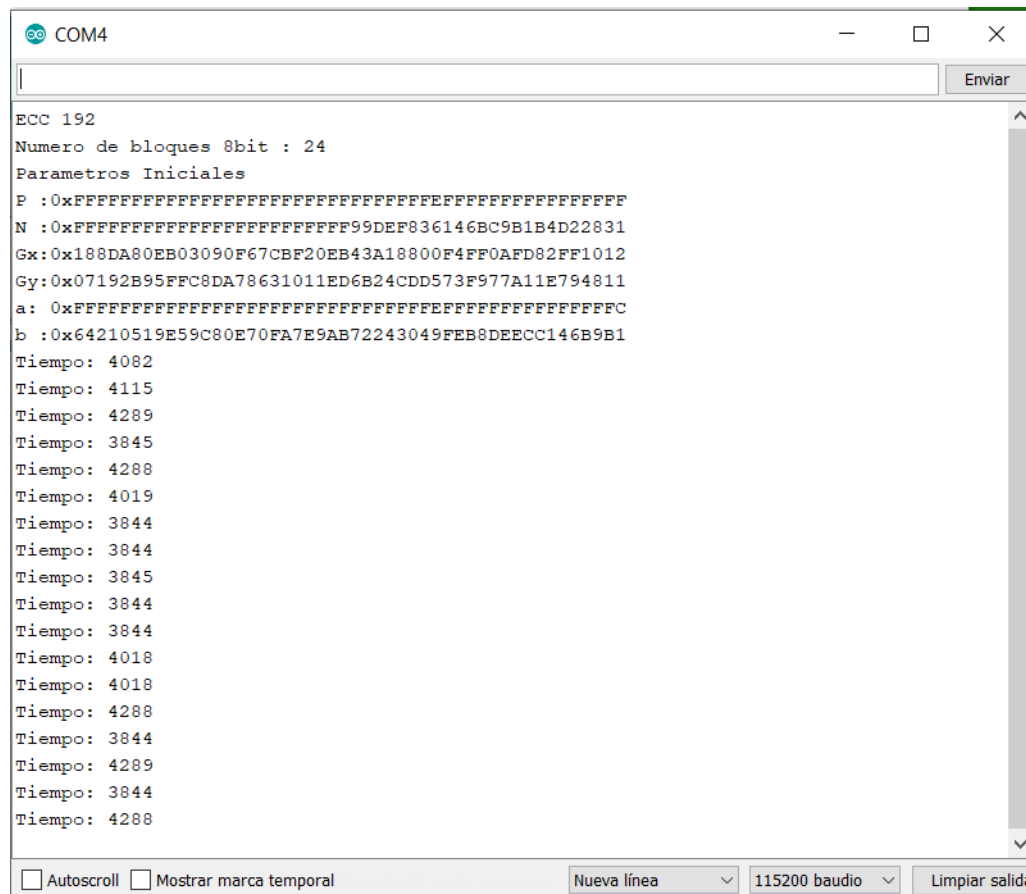
```
COM4
ECC 128
Numero de bloques 8bit : 16
Parametros Iniciales
P : 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
N : 0xFFFFFFFF0000000075A30D1B9038A115
Gx: 0x161FF7528B899B2D0C28607CA52C5B86
Gy: 0xCF5AC8395BAFEB13C02DA292DDED7A83
a : 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC
b : 0xE87579C11079F43DD824993C2CEE5ED3
Tiempo: 1719
Tiempo: 1745
Tiempo: 1655
Tiempo: 1745
Tiempo: 1593
Tiempo: 1758
Tiempo: 1594
Tiempo: 1759
Tiempo: 1759
Tiempo: 1594
Tiempo: 1758
Tiempo: 1593
Tiempo: 1759
Tiempo: 1760
Tiempo: 1593
Tiempo: 1758
Tiempo: 1594
Tiempo: 1593
Tiempo: 1759

☐ Autoscroll ☐ Mostrar marca temporal
Nueva línea 115200 baudio Limpiar salida
```

Figura 5.2: Tiempos de ejecución Left-to-Right 128 bits

Como se puede ver en la Figura 5.2, la ejecución demora entre 1593ms y 1759ms.

En la Figura 5.3 se muestran los tiempos de ejecución del Algoritmo 1 Left-to-Right con un nivel de seguridad de 192 bits.



```
COM4
ECC 192
Numero de bloques 8bit : 24
Parametros Iniciales
P :0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
N :0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF99DEF836146BC9B1B4D22831
Gx:0x188DA80EB03090F67CBF20EB43A18800F4FF0AFD82FF1012
Gy:0x07192B95FFC8DA78631011ED6B24CDD573F977A11E794811
a: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC
b :0x64210519E59C80E70FA7E9AB72243049FEB8DEECC146B9B1
Tiempo: 4082
Tiempo: 4115
Tiempo: 4289
Tiempo: 3845
Tiempo: 4288
Tiempo: 4019
Tiempo: 3844
Tiempo: 3844
Tiempo: 3845
Tiempo: 3844
Tiempo: 3844
Tiempo: 4018
Tiempo: 4018
Tiempo: 4288
Tiempo: 3844
Tiempo: 4289
Tiempo: 3844
Tiempo: 4288

☐ Autoscroll ☐ Mostrar marca temporal
Nueva línea 115200 baudio Limpiar salida
```

Figura 5.3: Tiempos de ejecución Left-to-Right 192 bits

Como se puede ver en la Figura 5.3, la ejecución demora entre 3844ms y 4289ms, resultando en un número cercano al indicado en las características de la librería.

En la Figura 5.4 se muestran los tiempos de ejecución del Algoritmo 1 Left-to-Right con un nivel de seguridad de 256 bits.





Algoritmo	Implementado
Left-to-Right	✓
Left-to-Right con Bloque Atómico	✓
Left-to-Right con Fórmula Unificada	✓
Always Double and Add de Coron	✓
Modified Montgomery Ladder	✓
Double-Add de Joye	✓
Add-Only de Joye	✓
Randomized Montgomery Ladder	✓
Signed Digit Method: Left-to-Right	✓
Primera Contramedida de Coron	
Contramedida de Clavier-Joye	✓
Segunda Contramedida de Coron	✓
Tercera Contramedida de Coron	✓
Montgomery Powering Ladder method with randomized address	✓
2P* Ciet-Joye	✓
BRIP	✓
Contramedida de Kim	✓
Contramedida de Ha	

Tabla 5.1: Implementaciones de Algoritmos

En la sección 5.2.2. se explicaran los motivos de porque esta contramedida no se pudo implementar.

El Algoritmo 5 Add-Only de Joye es un caso particular, ya que solo se puede implementar con Fórmula Unificada.

### 5.1.3. Método de comprobación de las implementaciones

Para asegurar la correcta implementación de cada contramedida se utilizó la calculadora online Magma. Puesto que los resultados de las implementaciones con fórmulas explícitas Jacobianas y con bloque atómico quedan en coordenadas Jacobianas y los resultados de la implementación con fórmula unificada queda en coordenadas proyectivas, fue necesaria la conversión de ellas a coordenadas afín con el objetivo de corroborar mediante una resta con los resultados obtenidos de Magma. La conversión se realiza de la siguiente manera.

Sea  $(X, Y, Z)$  las coordenadas obtenidas del Arduino y  $(x, y)$  las coordenadas afín. Se utiliza la fórmula:

$$(x, y) = \left( \frac{X}{Z^2}, \frac{Y}{Z^3}, \frac{Z}{Z} \right)$$

para las salidas Jacobianas del Arduino y la fórmula

$$(x, y) = \left( \frac{X}{Z}, \frac{Y}{Z}, \frac{Z}{Z} \right)$$

para las salidas en coordenadas proyectivas.

Luego de la conversión, se realiza una resta entre el  $(x, y)$  obtenido y la variable  $[k]P$  creada en Magma, si dicha resta resulta en 0, quiere decir que ambas salidas son iguales y por lo tanto el resultado del Arduino es correcto.

## 5.2. Limitaciones

Mientras se realizaba la programación hubo ciertas limitaciones que dificultaron la implementación de las contramedidas, algunas ya sean por las características del Arduino o porque la librería utilizada no lo permite.

### 5.2.1. Generación de un punto aleatorio

Como es sabido, los puntos en curva elíptica se rigen por la ecuación de Weierstrass:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

particularmente se utilizó la versión corta de esta ecuación  $y^2 = x^3 + ax + b$ , con esta se crea un punto a partir de un  $x \in E(\mathbb{F}_p)$ , sin embargo, la librería no cuenta con una función que permita realizar una raíz cuadrada, y programar una función que permita calcularla en base al método de formación de un numero de 256 bits no es trivial. Por lo que, para solventar esta limitante de los puntos aleatorios, se utilizó la calculadora online Magma la cual permite generar puntos de una curva. Una vez generados estos puntos se creo una matriz para guardarlos en memoria del Arduino y utilizando un número aleatorio entre 0 y  $n - 1$ , con  $n$  como la cantidad de puntos generados, se seleccionaba al azar uno de ellos. Finalmente cabe decir que este guardado de datos no influye en los problemas de memoria, dado que el Arduino guarda los datos del código en la memoria FLASH la cual es de 32KB.

### 5.2.2. Limitaciones de memoria

El Arduino UNO posee una memoria RAM de solo 2KB, esto impide que se pueda usar libremente variables adicionales para implementar una contramedida, por este motivo, en cada contramedida se intento ocupar la menor cantidad de variables posibles, sin embargo, aun ocupando la menor cantidad de memoria posible hubieron contramedidas que no pudieron ser implementadas en Arduino UNO. A continuación, se mencionaran las contramedidas que presentaron problemas y una explicación del problema.

La contramedida de Ha no pudo ser implementada debido a que por si sola utiliza muchos registros temporales. Sin embargo, fue posible una implementación solo cuando se utilizaban datos en bruto dentro del código, es decir, tanto el escalar  $k$  como el punto  $P$  debían ser variables fijas dentro de las funciones del código. Lo cual no es permitido por la contramedida, ya que estos deben ser variables de entrada de la función.



## Capítulo 6

# Resultados del análisis

### 6.1. Método de medición

La recolección de datos relacionados a la energía se realizó utilizando un Arduino Atmega 2560 como dispositivo tipo DAQ, el cual realiza una captura de datos en tiempo real, con una frecuencia de 1 khz la cual es interpretada por el Arduino con una resolución máxima de 1024 bits (Arduino, 2020).

El dispositivo DAQ funciona midiendo la caída de tensión en la resistencia del circuito, que en este caso es de 1 ohm. Luego envía los valores de la caída de tensión a un pin análogo. Dichos valores corresponden al voltaje que va de 0 a 5 volts (V), sin embargo el Arduino lo interpreta como un valor que va del 0 al 1023. Por lo que se debe realizar una conversión de datos para obtener los valores en volts. Esta conversión se realiza mediante la ecuación:

$$v_R = \frac{v_A * 5}{1023}$$

donde  $v_R$  es el voltaje real en milivoltios (mV) y  $v_A$  es el valor numérico adimensional medido por el Arduino. En la Figura 6.1 se muestra el esquema del circuito implementado para medir el voltaje.

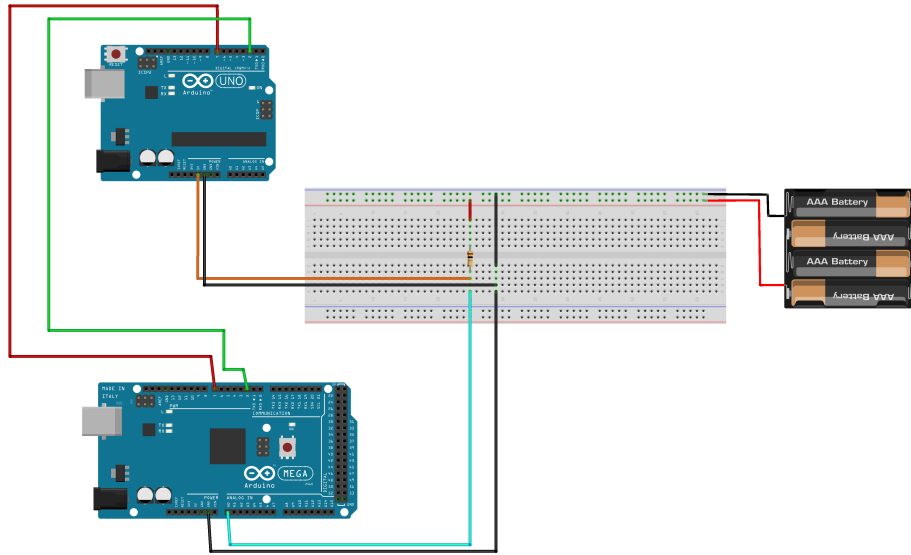


Figura 6.1: Esquema del circuito implementado

Para que este circuito funcione fue necesario una fuente externa de corriente de 5V para el Arduino UNO, ya que este es el que ejecuta las contramedidas. Mientras que el Arduino Atmega, que medirá el consumo de energía, está conectado directamente al ordenador.

Ambos dispositivos están conectados por medio de los pin 2 y 7, los cuales sirven como comunicadores para indicar cuando el Arduino Atmega debe comenzar a registrar la energía consumida por el Arduino UNO. Además, en el Arduino Atmega se utiliza la entrada analógica A0 para la lectura de datos.

Finalmente cabe mencionar que una medición consta de varios registros obtenidos por el Arduino Atmega donde la cantidad de registros varía según el tiempo que demore en terminar la ejecución de la contramedida. Además, para tener resultados más exactos, se realizaron 100 mediciones para cada contramedida.

## 6.2. Resultados de la medición

Puesto que el objetivo de esta tesis es analizar el consumo de energía de las contramedidas, es necesario explicar que se entiende por consumo energético.

El Joule o Julio ( $J$ ), es el trabajo necesario para producir un Watt ( $W$ ) de potencia en un segundo (Connor, 2019), es decir,  $J = W \cdot \text{seg}$ . Esto indica cuanto voltaje es necesario para que funcionen las contramedidas en el lapso de tiempo que les toma ejecutarse, dado que los Watts se calculan por medio de la fórmula  $W = V \cdot A$ , donde  $V$  es el voltaje y  $A$  es el amperaje o intensidad. Como los datos que entrega el Arduino se transforman en Voltaje, por medio de la Ley de Ohm  $V = A \cdot R$  se puede obtener el amperaje para calcular los Watts. La resistencia utilizada es de 1 Ohm, por

lo que la fórmula resulta en  $V = A$ , entonces los Watts se pueden calcular, en este caso, como  $W = V^2$ . Además, se calculará la varianza de la tensión para ver la dispersión de los datos, esta varianza es el promedio de las varianzas obtenidas de cada medición.

A continuación se muestran los resultados obtenidos por contramedida y luego un resumen general de los resultados.

### 6.2.1. Left to Right

En la Tabla 6.1 se muestran los resultados obtenidos del Algoritmo 1 Left to Right. Como se puede ver la implementación que toma menos tiempo en ejecutarse es la con Fórmulas Explícitas Jacobianas (FEJ) con 10,198 segundos, mientras que la que mas demora es la implementación con Fórmula Unificada con 16,610 segundos.

Como se puede ver en la tabla 6.1 la Energía indica que la implementación con Fórmula Unificada consume un 62.7 % más de energía en comparación a la implementación con Fórmulas Explícitas Jacobianas (FEJ). Mientras que la implementación con Bloques Atómicos consume un 5.8 % más de energía que la implementación con FEJ.

Left to Right			
Parámetro	FEJ	Bloque Atómico	Fórmula Unificada
Tiempo (seg)	10,198	10,790	16,610
Tensión Mínima ( $mV$ )	4,872	4,871	4,864
Tensión Promedio ( $mV$ )	4,889	4,888	4,886
Tensión Máxima ( $mV$ )	4,905	4,906	4,907
Potencia Mínima ( $\mu W$ )	23,734	23,722	23,661
Potencia Promedio ( $\mu W$ )	23,901	23,889	23,875
Potencia Máxima ( $\mu W$ )	24,062	24,072	24,079
Energía ( $\mu J$ )	243,736	257,760	396,569
Varianza	1,496E-05	1,769E-05	1,410E-05

Tabla 6.1: Resultados Left to Right

### 6.2.2. Always Double and Add

En la Tabla 6.2 se muestran los resultados obtenidos del Algoritmo 2 Always Double and Add de Coron.

Always Double and Add	
Parámetro	Resultado
Tiempo (seg)	15,553
Tensión Mínima ( $mV$ )	4,865
Tensión Promedio ( $mV$ )	4,883
Tensión Máxima ( $mV$ )	4,901
Potencia Mínima ( $\mu W$ )	23,665
Potencia Promedio ( $\mu W$ )	23,842
Potencia Máxima ( $\mu W$ )	24,024
Energía ( $\mu J$ )	370,800
Varianza	1,576E-05

Tabla 6.2: Resultados Always Double and Add

Como se puede ver el consumo energético de esta contramedida es casi un 52.13 % mayor a la implementación del Left-to-Right FEJ, y demora un 52.5 % más.

### 6.2.3. Modified Montgomery Ladder

En la Tabla 6.3 se muestran los resultados obtenidos del Algoritmo 3 Modified Montgomery Ladder.

Modified Montgomery Ladder	
Parámetro	Resultado
Tiempo (seg)	15,640
Tensión Mínima ( $mV$ )	4,840
Tensión Promedio ( $mV$ )	4,863
Tensión Máxima ( $mV$ )	4,883
Potencia Mínima ( $\mu W$ )	23,427
Potencia Promedio ( $\mu W$ )	23,647
Potencia Máxima ( $\mu W$ )	23,843
Energía ( $\mu J$ )	369,836
Varianza	4,067E-05

Tabla 6.3: Resultados Modified Montgomery Ladder

Como se puede ver el consumo energético de esta contramedida aumenta en un 51.73 % en comparación con el consumo energético del Left-to-Right FEJ, mientras que el tiempo aumenta en un 53.35 %.

#### 6.2.4. Double Add de Joye

En la Tabla 6.4 se muestran los resultados obtenidos del Algoritmo 4 Double-Add de Joye resistente contra SPA.

Double-Add de Joye resistente contra SPA	
Parámetro	Resultado
Tiempo (seg)	15,606
Tensión Mínima ( $mV$ )	4,804
Tensión Promedio ( $mV$ )	4,849
Tensión Máxima ( $mV$ )	4,4,874
Potencia Mínima ( $\mu W$ )	23,076
Potencia Promedio ( $\mu W$ )	23,513
Potencia Máxima ( $\mu W$ )	23,754
Energía ( $\mu J$ )	366,950
Varianza	1.446E-04

Tabla 6.4: Resultados Double Add de Joye

Como se puede ver esta contramedida tiene un consumo energético de un 50.55 % más que la implementación del Left-to-Right FEJ. Por otro lado, el tiempo de ejecución aumenta en un 53.03 %.

#### 6.2.5. Add-Only

En la Tabla 6.5 se muestran los resultados obtenidos del Algoritmo 5 Add-Only de Joye.

Add-Only de Joye	
Parámetro	Resultado
Tiempo (seg)	22,208
Tensión Mínima ( $mV$ )	4,801
Tensión Promedio ( $mV$ )	4,843
Tensión Máxima ( $mV$ )	4,871
Potencia Mínima ( $\mu W$ )	23,050
Potencia Promedio ( $\mu W$ )	23,461
Potencia Máxima ( $\mu W$ )	23,730
Energía ( $\mu J$ )	521,013
Varianza	1.26976E-04

Tabla 6.5: Resultados Add-Only de Joye

De los resultados se puede apreciar que esta contramedida consume un 113.76 % más de energía que la implementación del Left-to-Right FEJ. Además, el tiempo de ejecución aumenta un

117.76 %.

### 6.2.6. Randomized Montgomery Ladder

En la Tabla 6.6 se muestran los resultados obtenidos del Algoritmo 6 Randomized Montgomery Ladder.

Randomized Montgomery Ladder	
Parámetro	Resultado
Tiempo (seg)	16,915
Tensión Mínima ( $mV$ )	4,793
Tensión Promedio ( $mV$ )	4,838
Tensión Máxima ( $mV$ )	4,870
Potencia Mínima ( $\mu W$ )	22,974
Potencia Promedio ( $\mu W$ )	23,411
Potencia Máxima ( $\mu W$ )	23,715
Energía ( $\mu J$ )	396,016
Varianza	1,409,E-04

Tabla 6.6: Resultados Randomized Montgomery Ladder

Como se puede ver en los resultados, esta contramedida en comparación a la implementación del Left-to-Right FEJ, consume un 62.47 % más de energía y demora un 65 % más.

### 6.2.7. Signed Digit

En la Tabla 6.7 se muestran los resultados obtenidos del Algoritmo 7 Signed Digit Method: Left-to-Right.

Signed Digit Method: Left-to-Right	
Parámetro	Resultado
Tiempo (seg)	15,484
Tensión Mínima ( $mV$ )	4,805
Tensión Promedio ( $mV$ )	4,846
Tensión Máxima ( $mV$ )	4,872
Potencia Mínima ( $\mu W$ )	23,092
Potencia Promedio ( $\mu W$ )	23,481
Potencia Máxima ( $\mu W$ )	23,733
Energía ( $\mu J$ )	363,574
Varianza	9,232E-05

Tabla 6.7: Resultados Signed Digit Method: Left-to-Right

De los resultados se puede ver que la contramedida consume un 49.16 % más de energía y demora un 51 % más que la implementación Left-to-Right FEJ.

### 6.2.8. Clavier-Joye

En la Tabla 6.8 se muestran los resultados obtenidos de la contramedida de Clavier-Joye.

Contramedida de Clavier-Joye	
Parámetro	Resultado
Tiempo (seg)	20,604
Tensión Mínima ( $mV$ )	4,775
Tensión Promedio ( $mV$ )	4,829
Tensión Máxima ( $mV$ )	4,847
Potencia Mínima ( $\mu W$ )	22,801
Potencia Promedio ( $\mu W$ )	23,324
Potencia Máxima ( $\mu W$ )	23,496
Energía ( $\mu J$ )	480,541
Varianza	1,358,E-04

Tabla 6.8: Resultados Contramedida de Clavier-Joye

Esta contramedida en comparación al Left-to-Right FEJ tiene un consumo de un 97.15 % más de energía y demora un 102.03 % más.

### 6.2.9. Segunda Contramedida de Coron: Blinding Point

En la Tabla 6.9 se muestran los resultados obtenidos del Algoritmo 8 Segunda Contramedida de Coron: Blinding Point.

Segunda Contramedida de Coron: Blinding Point	
Parámetro	Resultado
Tiempo (seg)	25,821
Tensión Mínima ( $mV$ )	4,769
Tensión Promedio ( $mV$ )	4,818
Tensión Máxima ( $mV$ )	4,847
Potencia Mínima ( $\mu W$ )	22,744
Potencia Promedio ( $\mu W$ )	23,214
Potencia Máxima ( $\mu W$ )	23,494
Energía ( $\mu J$ )	599,401
Varianza	1,270,E-04

Tabla 6.9: Resultados Segunda Contramedida de Coron: Blinding Point

Esta contramedida en comparación al Left-to-Right FEJ tiene un consumo de un 145.92 % más y demora un 153.1 % más

### 6.2.10. Tercera Contramedida de Coron

En la Tabla 6.10 se muestran los resultados obtenidos de la Tercera Contramedida de Coron.

Tercera Contramedida de Coron	
Parámetro	Resultado
Tiempo (seg)	10,487
Tensión Mínima ( $mV$ )	4,867
Tensión Promedio ( $mV$ )	4,886
Tensión Máxima ( $mV$ )	4,893
Potencia Mínima ( $\mu W$ )	23,691
Potencia Promedio ( $\mu W$ )	23,870
Potencia Máxima ( $\mu W$ )	23,943
Energía ( $\mu J$ )	250,331
Varianza	1,583E-05

Tabla 6.10: Resultados Tercera Contramedida de Coron

Esta contramedida en comparación al Left-to-Right FEJ tiene un consumo de un 2.7 % más de energía y demora un 2.83 % más.

### 6.2.11. Método 2P\* de Ciet-Joye

En la Tabla 6.11 se muestran los resultados obtenidos del Algoritmo 9 Método 2P\* de Ciet-Joye.

Método 2P* de Ciet-Joye	
Parámetro	Resultado
Tiempo (seg)	15,682
Tensión Mínima ( $mV$ )	4,877
Tensión Promedio ( $mV$ )	4,894
Tensión Máxima ( $mV$ )	4,902
Potencia Mínima ( $\mu W$ )	23,784
Potencia Promedio ( $\mu W$ )	23,949
Potencia Máxima ( $\mu W$ )	24,029
Energía ( $\mu J$ )	375,568
Varianza	9,915E-06

Tabla 6.11: Resultados Método 2P\* de Ciet-Joye



El Método 2P\* de Ciet-Joye tiene un consumo de un 54.08 % más de energía que el Left-to-Right FEJ, además, demora un 53.7 % más.

### 6.2.12. Montgomery Powering Ladder

En la Tabla 6.12 se muestran los resultados obtenidos del Algoritmo 10 Montgomery Powering Ladder method with randomized address.

Montgomery Powering Ladder	
Parámetro	Resultado
Tiempo (seg)	15,632
Tensión Mínima ( $mV$ )	4,772
Tensión Promedio ( $mV$ )	4,816
Tensión Máxima ( $mV$ )	4,844
Potencia Mínima ( $\mu W$ )	22,775
Potencia Promedio ( $\mu W$ )	23,198
Potencia Máxima ( $\mu W$ )	23,463
Energía ( $\mu J$ )	362,628
Varianza	1,197,E-04

Tabla 6.12: Resultados Montgomery Powering Ladder

Como se puede ver en la tabla 6.12, esta contramedida utiliza un 48,77 % más de energía que el Left-to-Right FEJ y consume un 53.2 % más de tiempo.

### 6.2.13. BRIP

En la Tabla 6.13 se muestran los resultados obtenidos del Algoritmo 11 BRIP.

BRIP	
Parámetro	Resultado
Tiempo (seg)	15,559
Tensión Mínima ( $mV$ )	4,796
Tensión Promedio ( $mV$ )	4,836
Tensión Máxima ( $mV$ )	4,865
Potencia Mínima ( $\mu W$ )	23,007
Potencia Promedio ( $\mu W$ )	23,388
Potencia Máxima ( $\mu W$ )	23,669
Energía ( $\mu J$ )	363,909
Varianza	1,164,E-04

Tabla 6.13: Resultados BRIP

De los resultados se puede ver que hay un consumo energético de un 49.30 % más y un aumento en el tiempo de ejecución de un 52.5 % en comparación al Left-to-Right FEJ.

#### 6.2.14. Contramedida de Kim

En la Tabla 6.14 se muestran los resultados obtenidos del Algoritmo 12 Contramedida de Kim et al.

Contramedida de Kim	
Parámetro	Resultado
Tiempo (seg)	14,240
Tensión Mínima ( $mV$ )	4,877
Tensión Promedio ( $mV$ )	4,889
Tensión Máxima ( $mV$ )	4,906
Potencia Mínima ( $\mu W$ )	23,786
Potencia Promedio ( $\mu W$ )	23,901
Potencia Máxima ( $\mu W$ )	24,067
Energía ( $\mu J$ )	340,318
Varianza	8,555E-06

Tabla 6.14: Resultados Contramedida de Kim

Los resultados muestran que esta contramedida tiene un consumo energético de un 39.62 % más en comparación al Left-to-Right FEJ. Y tiene un aumento en el tiempo de ejecución del 39.63 %.

### 6.3. Resumen de Resultados

A continuación, se presenta un resumen de los resultados según el tipo de ataque al que esta enfocado, con el fin de visualizar que implementación de la contramedida consume menos energía.

#### 6.3.1. Resultados para Simple Power Analysis

En la Tabla 6.15 se recopilan los resultados de las implementaciones de las contramedidas para los Simple Power Analysis.

Resumen SPA		
Contramedida	Energía ( $\mu J$ )	Tiempo (Seg)
Left to Right	243,736	10,198
Left to Right con Bloque Atómico	257,760	10,790
Left to Right con Fórmula Unificada	396,569	16,610
Always Double and Add	370,800	15,553
Modified Montgomery Ladder	369,836	15,640
Double-Add de Joye	366,950	15,606
Add-Only	521,013	22,208
Randomized Montgomery Ladder	396,016	16,915
Signed Digit	363,574	15,484

Tabla 6.15: Resumen Resultados SPA

Es posible apreciar que la contramedida con menor consumo energético es el Left-to-Right implementado con Bloques Atómicos. Por otro lado, la contramedida que más energía utiliza es el Add-Only.

### 6.3.2. Resultados para Differential Power Analysis

En la Tabla 6.16 se recopilan los resultados de las implementaciones de las contramedidas para los Differential Power Analysis.

Resumen DPA		
Contramedida	Energía ( $\mu J$ )	Tiempo (Seg)
Left to Right	243,736	10,198
Contramedida de Clavier-Joye	480,541	20,604
Segunda Contramedida de Coron	599,401	25,821
Tercera Contramedida de Coron	250,331	10,487
Método 2P*	375,568	15,682

Tabla 6.16: Resumen Resultados DPA

Es posible apreciar que la Tercera Contramedida de Coron es la que tiene el menor consumo energético en comparacion al Left-to-Right.

### 6.3.3. Resultados para Address-bit DPA

En la Tabla 6.17 se recopilan los resultados de las implementaciones de las contramedidas para los Address-bit DPA.

Resumen Address-bit DPA		
Contramedida	Energía ( $\mu J$ )	Tiempo (Seg)
Left to Right	243,736	10,198
Montgomery Powering Ladder	362,628	15,632

Tabla 6.17: Resumen Resultados Address-bit DPA

#### 6.3.4. Resultados para Ataque Múltiples

En la Tabla 6.18 se recopilan los resultados de las implementaciones de las contramedidas para los Ataques Múltiples.

Resumen Ataques Múltiples		
Contramedida	Energía ( $\mu J$ )	Tiempo (Seg)
Left to Right	243,736	10,198
BRIP	363,909	15,559
Contramedida de Kim	340,318	14,240

Tabla 6.18: Resumen Resultados Ataques Múltiples

De la tabla 6.18 se puede apreciar que la contramedida de Kim et. al, es la que tiene el menor consumo energético.

# Conclusiones

En este trabajo, se presentaron algunas contramedidas para los ataques de canal lateral en el criptosistema de curva elíptica con el fin de analizar cual es el consumo energético de ellas. Para ello fue necesaria la implementación de dichas contramedidas en una arquitectura que sea similar a la de los dispositivos embebidos que existen en la actualidad, esta arquitectura resulto ser de 8-bits por lo que la implementación tuvo que ser lo más eficiente posible en el uso de memoria. Las implementaciones de las contramedidas fue realizada mediante Fórmulas Explícitas Jacobianas, con las excepciones de la implementación del Left-to-Right el cual se implemento, adicionalmente, con Bloques Atómicos y Fórmula Unificada, y el Add-Only de Joye el cual solo se puede implementar con Fórmula Unificada. Dentro de las contramedidas, las únicas que no se pudieron implementar debido a las limitaciones de la librería o del Arduino, fueron la Primera Contramedida de Coron dado que la librería al realizar una operación automáticamente calcula de forma modular el resultado, esto provoca que el  $k' = k + d(\#E)$  resulte en  $k' = k$ , por ende, no se puede enmascarar el  $k$  en la contramedida.

Y la Contramedida de Ha et al. tampoco se pudo implementar debido a que por si sola supera el límite de memoria del Arduino UNO, por lo que esta no se logra ejecutar.

Dentro de los resultados obtenidos en relación a la energía, se obtuvo que Bloque Atómico es la contramedida más eficiente contra los SPA, ya que su consumo energético aumenta un 5.75 % en comparación al Left-to-Right. En relación a las contramedidas para DPA, la Tercera Contramedida de Coron es la más eficiente, con un aumento del 2.7 % en el consumo de energía. Por otro lado, en las contramedidas para Address-bit DPA, solo esta el Montgomery Powering Ladder con un aumento en el consumo de energía de un 48.77 % en comparación al algoritmo de la multiplicación escalar. Finalmente entre las contramedidas para ataques múltiples, la contramedida de Kim et al. es la más eficiente en consumo energético, ya que consume un 39.62 % más en comparación al Left-to-Right y protege contra distintos tipos de ataques, tales como el DPA, RPA, SVA, ZPA, doubling attacks y 2-torsion attacks.

En cambio, la contramedida Add-Only de Joye es la de mayor consume energético entre las que brindan protección contra los SPA, con un aumento en el consumo de un 113.76 %. Mientras que la Segunda contramedida de Coron: Blinding Point, es la que consume más energía entre

las contramedidas para los DPA, consumiendo un 145.92% más de energía. Y Finalmente la contramedida BRIP, con un 49.30 % más en el consumo energético, es la contramedida menos eficiente entre las contramedidas que protegen contra ataques múltiples.

# Bibliografía

- Abarzúa, R., & Thériault, N. (2012). Complete Atomic Blocks for Elliptic Curves in Jacobian Coordinates over Prime Fields. *LATINCRYPT 2012* (págs. 37-55). Berlin, Heidelberg: Springer.
- Akashita, T., & Takagi, T. (2003). Zero-value point attacks on elliptic curve cryptosystem. *ISC 2003* (págs. 218-233). Berlin, Heidelberg: Springer.
- Akishita, T., & Takagi, T. (2004). On the Optimal Parameter Choice for Elliptic Curve Cryptosystems Using Isogeny. *PKC 2004* (págs. 346-359). Berlin, Heidelberg: Springer.
- Amiel, F., Feix, B., Tunstall, M., Whelan, C., & Marnane, W. (2008). Distinguishing multiplications from squaring. *SAC 2008* (págs. 346-360). Berlin, Heidelberg: Springer.
- Amiel, F., Villegas, K., Feix, B., & Marcel, L. (2007). Passive and Active Combiner Attack. *FDTC 2007* (págs. 92-99). IEEE.
- Arduino. (2020). AnalogReadResolution. Recuperado de: <https://www.arduino.cc/reference/en/language/functions/zero-due-mkr-family/analogreadresolution/>
- Avanzi, R. (2005). Side Channel Attacks on Implementations of Curve-Based Cryptographic Primitives. Obtenido de *Cryptology ePrint Archive*: <https://eprint.iacr.org/2005/017.pdf>
- Bauer, A., Jaulmes, E., Prouff, E., Reinhard, J., & Wild, J. (2014). Horizontal collision correlation attack on elliptic curves – Extended Version – . *Cryptography and Communications 2014* (págs. 91-119). Springer.
- Bernstein, D., & Lange, T. (2007). Faster Addition and Doubling on Elliptic Curves. *ASIACRYPT 2007* (págs. 29-50). Berlin, Heidelberg: Springer.
- Bessalov, A., & Kovalchuk, L. (2019). Supersingular Twisted Edwards Curves Over Prime Fields. I. Supersingular Twisted Edwards Curves with  $j$ -Invariants Equal to Zero and 123. *Cybernetics and Systems Analysis*, 347-353.
- Billet, O., & Joye, M. (2003). The Jacobi model of an elliptic curve and side-channel analysis. *AAECC 2003* (págs. 34-42). Berlin, Heidelberg: Springer.
- Brier, É., & Joye, M. (2002). Weierstraß elliptic curves and side-channel attacks. *PKC 2002* (págs. 335-345). Berlin, Heidelberg: Springer.
- Brier, E., & Joye, M. (2003). Fast point multiplication on elliptic curves through isogenies. *AAECC 2003: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes* (págs. 43-50).

- Springer.
- Brown, D. (27 de January de 2010). SEC 2: Recommended Elliptic Curve Domain Parameters. Obtenido de <https://www.secg.org/sec2-v2.pdf>
- Caddy, T. (2011). Differential Power Analysis. En H. van Tilborg, & S. Jajodia, Encyclopedia of Cryptography and Security. Boston: Springer.
- Chevallier-Mames, B., Ciet, M., & Joye, M. (2004). Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity. IEEE Transactions on Computers (págs. 760-768). IEEE.
- Chen, T., Li, H., Wu, K., & Yu, F. (2009). Countermeasure of ECC against Side-Channel Attacks: Balanced Point Addition and Point Doubling Operation Procedure. APCIP 2009 (págs. 465-469). IEEE.
- Ciet, M., & Joye, M., (2003). (Virtually) Free randomization techniques for elliptic curve cryptography. ICICS 2003 (págs. 348-359). Springer-Verlag,
- Clavier, C., & Joye, M., (2001). Universal Exponentiation Algorithm - A First Step Towards PROvable SPA-resistance -. CHES 2001, LNCS 2162 (págs 300-308). Springer.
- Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., & Vercauteren, F. (2005). Handbook of Elliptic and Hyperelliptic Curve Cryptography. New York: Chapman and Hall/CRC.
- Connor, N. (2019). Qué es Joule (unidad J) – Unidad de energía – Definición. Recuperado de: <https://www.thermal-engineering.org/es/que-es-joule-unidad-j-unidad-de-energia-definicion/>
- Coron, J.-S. (1999). Resistance against differential power analysis for elliptic curve cryptosystems. CHES 1999: Cryptographic Hardware and Embedded Systems (págs. 292-302). Berlin, Heidelberg: Springer.
- Ebeid, N. (2007). Key randomization countermeasures to power analysis attacks on elliptic curve cryptosystems. Electrical and Computer Engineering. University of Waterloo.
- Fan, J., & Verbaauwhede, I. (2012). An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost. Cryptography and Security: From Theory to Applications (págs. 265-282). Berlin, Heidelberg: Springer.
- Fan, J., Guo, X., Mulder, E., Schaumont, P., Preneel, B., & Verbaauwhede, I. (2010). State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures. HOST 2010. Anaheim: IEEE.
- Feix, B., Roussellet, M., & Venelli, A. (2014). Side-Channel analysis on blinded regular scalar multiplications. INDOCRYPT 2014 (págs. 3-20). Springer.
- Fouque, P., Réal, D., Valette, F., & Drissi, M. (2008). The Carry Leakage on the Randomized Exponent Countermeasure. CHES 2008 (págs. 198-213). Springer.
- Fouque, P., & Valette, F. (2003). The doubling attack why upwards is better than downwards. CHES



- 2003 (págs. 269-280). Berlin, Heidelberg: Springer.
- Gandolfi, K., Mourtel, C., & Olivier, F. (2001). Electromagnetic Analysis: Concrete Results. CHES 2001: Cryptographic Hardware and Embedded Systems (págs. 251-261). Berlin, Heidelberg: Springer.
- GeeksforGeeks. (s.f.). Horner's Method for Polynomial Evaluation. Recuperado de <https://www.geeksforgeeks.org/horners-method-polynomial-evaluation/>
- Giraud, C., & Verneuil, V. (2010). Atomicity improvement for elliptic curve scalar multiplication. CARDIS 2010 (págs. 80-101). Berlin, Heidelberg: Springer.
- Giry, D., & Quisquater, J. (2011). Bluekrypt cryptographic key length recommendation. Recuperado de: <https://www.keylength.com>
- Goubin, L. (2002). A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. PKC 2003 (págs. 199-211). Berlin, Heidelberg: Springer.
- Goundar, R., Joye, M., Miyaji, A., Rivain, M., & Venelli, A. (2011). Scalar multiplication on Weierstraß elliptic curves from Co-Z arithmetic. Journal of Cryptographic Engineering (págs. 161-176). Berlin, Heidelberg: Springer.
- Ha, J., Park, J., Moon, S., & Yen, S. (2007). Provably Secure Countermeasure Resistant to Several Types of Power Attack for ECC. WISA 2007 (págs. 333-344). Berlin: Springer.
- Hanley, N., Kim, H., & Tunstall, M. (2015). Exploiting collisions in addition chain-based exponentiation algorithms. CT-RSA 2015 (págs. 431-448). Berlin, Heidelberg: Springer.
- Hanley, N., Tunstall, M., & Marnane, W. (2011). Using templates to distinguish multiplications from squaring operations. International Journal of Information Security, 255-266.
- Itoh, K., Izu, T., & Masahiko, T. (2003). A practical countermeasure against address-bit differential power analysis. CHES 2003 (págs. 382-396). Berlin: Springer.
- Itoh, K., Izu, T., & Takenaka, M. (2002). Address-Bit Differential Power Analysis of Cryptographic Schemes OK-ECDH and OK-ECDSA. CHES 2002 (págs. 129-143). Berlin, Heidelberg: Springer.
- Izu, T., & Takagi, T. (2002). A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks. PKC 2002 (págs. 280-296). Berlin, Heidelberg: Springer.
- Izu, T., & Takagi, T. (2003). Exceptional Procedure Attack on Elliptic Curve Cryptosystems. PKC 2003 (págs. 224-239). Berlin, Heidelberg: Springer.
- Izumi, M., Ikegami, J., Sakiyama, K., & Ohta, K. (2010). Improved countermeasure against Address-bit DPA for ECC scalar multiplication. 2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010). Dresden: IEEE.
- Jing, Q., Vasilakos, A., Wan, J., Lu, J., & Qiu, D. (2014). Security of the Internet of Things: perspectives and challenges. Wireless Netw 20. Springer.
- Joye, M. (2007). Highly Regular Right-to-Left Algorithms for Scalar Multiplication. CHES 2007

- (págs. 135-147). Berlin, Heidelberg: Springer.
- Joye, M., Olivier, F. (2011). Side-Channel Analysis. En H. va Tilborg, & S. Jajodia, Encyclopedia of Cryptography and Security. Boston: Springer.
- Joye, M., & Yen, S-M. (2002). The Montgomery Powering Ladder. CHES 2002 (págs. 291-302). Berlin, Heidelberg: Springer.
- Joye, M., & Quisquater, J. (2001). Hessian Elliptic Curves and Side-Channel Attack. CHES 2001 (págs. 402-410). Springer.
- Kim, C., Ha, J., Moon, S., Yen, S., Lien, W., & Kim, S. (2005). IACR Cryptology ePrint. Obtenido de An Improved and Efficient Countermeasure: <https://eprint.iacr.org/2005/022.pdf>
- Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation. Math. Comp. 48, 177, 203-209.
- Kocher, P. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. CRYPTO 1996: Advances in Cryptology (págs. 104-113). Berlin, Heidelberg: Springer.
- Kocher, P., Jaffe, J., & Jun, B. (1999). Differential Power Analysis. CRYPTO 1999: Advances in Cryptology (págs. 388-397). Berlin, Heidelberg: Springer.
- Le, D.-P., Tan, C., & Tunstall, M. (2015). Randomizing the Montgomery powering ladder. WISTP 2015 (págs. 169-184). berlin, Heidelberg: Springer.
- Longa, P., & Miri, A. (2008). Fast and flexible elliptic curve point arithmetic over prime fields. IEEE Transactions on Computers 57, 3, 289-302.
- López, J., & Dahab, R. (1999). Fast multiplication on elliptic curves over  $GF(2^m)$  without precomputation. CHES 1999 (págs. 316-327). Berlin, Heidelberg: Springer.
- Mamiya, H., Miyaji, A., & Morimoto, H. (2004). Efficient Countermeasures against RPA, DPA, and SPA. CHES 2004 (págs. 343-356). Berlin: Springer.
- McCann, D., Eder, K., & Oswald, E. (2015). Characterising and Comparing the Energy Consumption of Side Channel Attack Countermeasures and Lightweight Cryptography on Embedded Devices. 2015 International Workshop on Secure Internet of Things (SIoT). Vienna, Austria: IEEE.
- Menezes, A., Vanstone, S., & Hankerson, D. (2004). Guide to Elliptic Curve Cryptography. Springer Professional Computing (2004). Springer.
- Miller, V. (1985). Use of elliptic curves in cryptography. CRYPTO 1985: Advances in Cryptology (págs. 417-426). Berlin, Heidelberg: Springer.
- Montgomery, P. (1987). Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation 48, 177, 243-264.
- Muller, F. & Valette, F. (2006). High-order attacks against the exponent splitting protection. PKC 2006 (págs. 315-329). Springer.

- Murdica, C. (2014). Physical security of elliptic curve cryptography. Telecom ParisTech. Recuperado de <https://pastel.archives-ouvertes.fr/tel-01179584/document>
- Murdica, C., Guilley, S., Danger, J., Hoogvorst, P., & Naccache, D. (2012). Same Values Power Analysis Using Special Points on Elliptic Curves. COSADE 2012 (págs. 183-198). Berlin, Heidelberg: Springer.
- NIST. (2013). FIPS 186-4 Digital Signature Standard (DSS). Recuperado de <https://csrc.nist.gov/publications/detail/fips/186/4/final>
- Popp, T., Mangard, S., & Oswald, E. (2007). Design & Test of Computers (págs 535-543). IEEE.
- Okeya, K., & Sakurai, K. (2000). Power analysis breaks elliptic curve cryptosystems even secure against the timing. INDOCRYPT 2000 (págs. 178-190). Berlin: Springer.
- Ryan, M. (2013). GitHub repository. Recuperado de <https://github.com/iSECPartners/nano-ecc>.
- Strobel, D., Oswald, D., Richter, B., Schellenberg, F., Paar, C. (2014). Microcontrollers as (In)Security Devices for Pervasive Computing Applications. Proceedings of the IEEE, vol 102. IEEE.
- Walter, C. (2004). Simple Power Analysis of Unified Code for ECC Double and Add. CHES 2004 (págs 191-204). Springer.
- Wenger, E., & Großschadl, J. (2012). An 8-bit AVR-Based Elliptic Curve Cryptographic RISC Processor for the Internet of Things. 45th Annual IEEE/ACM International Symposium on Microarchitecture Workshops 2012 (págs. 39-46). Vancouver.
- Yen, S-M., & Joye, M. (2000). IEEE Transactions on Computers Vol. 49 (págs 967-970). IEEE.
- Yen, S-M., Lien, W., Moon, S., & Ha, J. (2005). Power Analysis by Exploiting Chosen Message and Internal Collisions – Vulnerability of Checking Mechanism for RSA-Decryption. Mycrypt2005 (págs. 183-195). Berlin: Springer.
- Yen, S-M., Kim, S., Lim, S., & Moon, S. (2001). A Countermeasure against One Physical Cryptanalysis May Benefit Another Attack. ICISC 2001 (págs. 269-294). Berlin, Heidelberg: Springer.
- Yen, S-M., Ko, L.-C., Moon, S., & Ha, J. (2006). Relative doubling attack against Montgomery ladder. ICISC 2005 (págs. 117-128). Berlin, Heidelberg: Springer.
- Zhang, N., Pei, Q., & Xiao, G. (2007). Elliptic curve scalar multiplication with x-coordinate. Wuhan University Journal of Natural Sciences, 163-166.