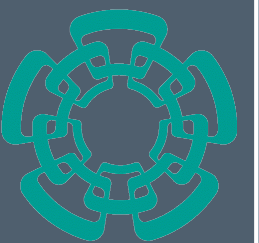
Análisis del algoritmo para cifrado de imágenes mediante curvas elípticas, implementado en un dispositivo programable FPGA

M. en C. José de Jesús Morales Romero



Contenido

- Introducción a la criptografía y redes neuronales artificiales.
- Redes Neuronales Artificiales.
- Criptografía y SCA sobre dispositivos programables.
- Criptografía con Curva Elíptica y su implementación en FPGA.
- Análisis del cifrado de imágenes utilizando RSA y ECC.

Contenido

- **Introducción a la criptografía y redes neuronales artificiales.**
- Redes Neuronales Artificiales.
- Criptografía y SCA sobre dispositivos programables.
- Criptografía con Curva Elíptica y su implementación en FPGA.
- Análisis del cifrado de imágenes utilizando RSA y ECC.

Introducción a la criptografía y redes neuronales artificiales

- Criptografía:
  - Criptografía de clave pública
    - Dos claves: una pública y una privada
      - RSA, ECC
  - Criptografía de clave privada
    - Una sola clave
      - AES, 3DES

Introducción a la criptografía y redes neuronales artificiales

› Redes Neuronales Artificiales en Criptografía

- Diferentes redes neuronales, entre ellas Redes Convolucionales, Perceptrón Multicapa, Redes Neuronales Pulsadas y Redes Neuronales Celulares.
- Se han usado como calculador de funciones Hash, como funciones no clonables o como cifrador de imágenes.
- Las CeNN son utilizadas para el procesamiento de imágenes y como oscilador caótico.

Introducción a la criptografía y redes neuronales artificiales

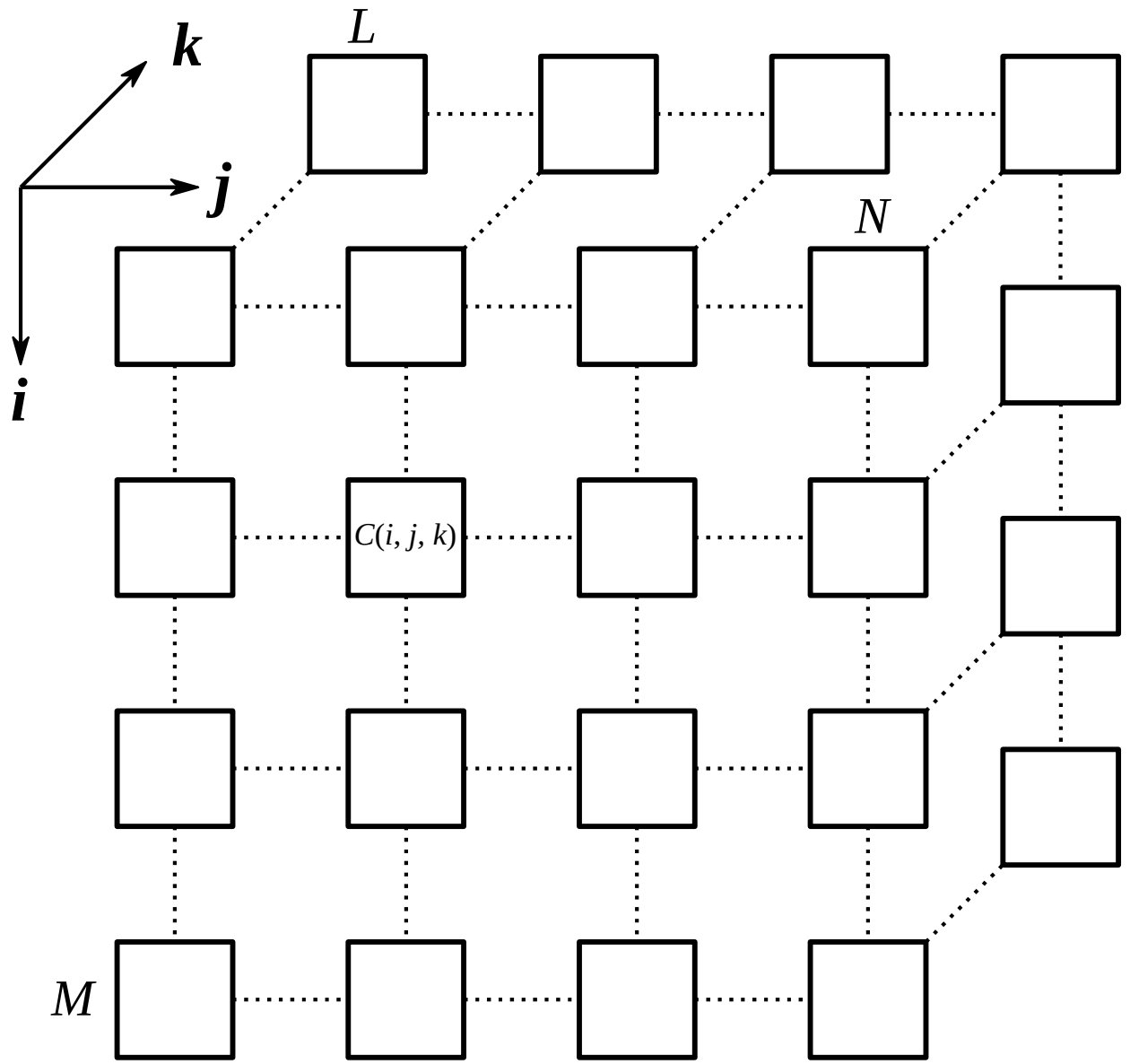
- Ataques a los sistemas criptográficos
- Profiled SCAs
  - Ataques de Canal Lateral (SCA) que utilizan redes neuronales artificiales.
- Non – Profiled SCAs
  - Ataques de Canal Lateral Pasivos
    - PAA: Power Analysis Attacks
    - TAA: Timing Analysis Attaccks
    - EAA: Electromagnetic Analysis Attacks

Contenido

- Introducción a la criptografía y redes neuronales artificiales.
- **Redes Neuronales Artificiales.**
- Criptografía y SCA sobre dispositivos programables.
- Criptografía con Curva Elíptica y su implementación en FPGA.
- Análisis del cifrado de imágenes utilizando RSA y ECC.

REDES NEURONALES ARTIFICIALES

› Red Neuronal Celular (CeNN)



Redes Neuronales Artificiales

BÚSQUEDA DE PLANTILLAS

- Métodos Analíticos
  - Aprendizaje local y aprendizaje global
- Métodos Heurísticos
  - PSO, GA, ABC y Simplex

Redes Neuronales Artificiales

ALGORITMO ABC

- **Abejas trabajadoras (EB)**: Abejas que son enviadas a las posibles soluciones y calculan que tan buena es la solución.
- **Abejas observadoras (OB)**: Son enviadas a las posibles soluciones, buscan en los alrededores una mejor solución y si encuentran una mejor, se actualiza la solución.
- **Abejas exploradoras (SB)**: Son enviadas a la búsqueda de nuevas soluciones dentro del área de solución.

Redes Neuronales Artificiales

ALGORITMO ABC

› Las abejas exploradoras utilizan:

$$x_i = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right)$$

Redes Neuronales Artificiales

ALGORITMO ABC

› Las abejas trabajadoras utilizan:

$$v_i = v_{i,j} + \phi_{i,j}\left(x_{i,j} - x_{k,j}\right)$$

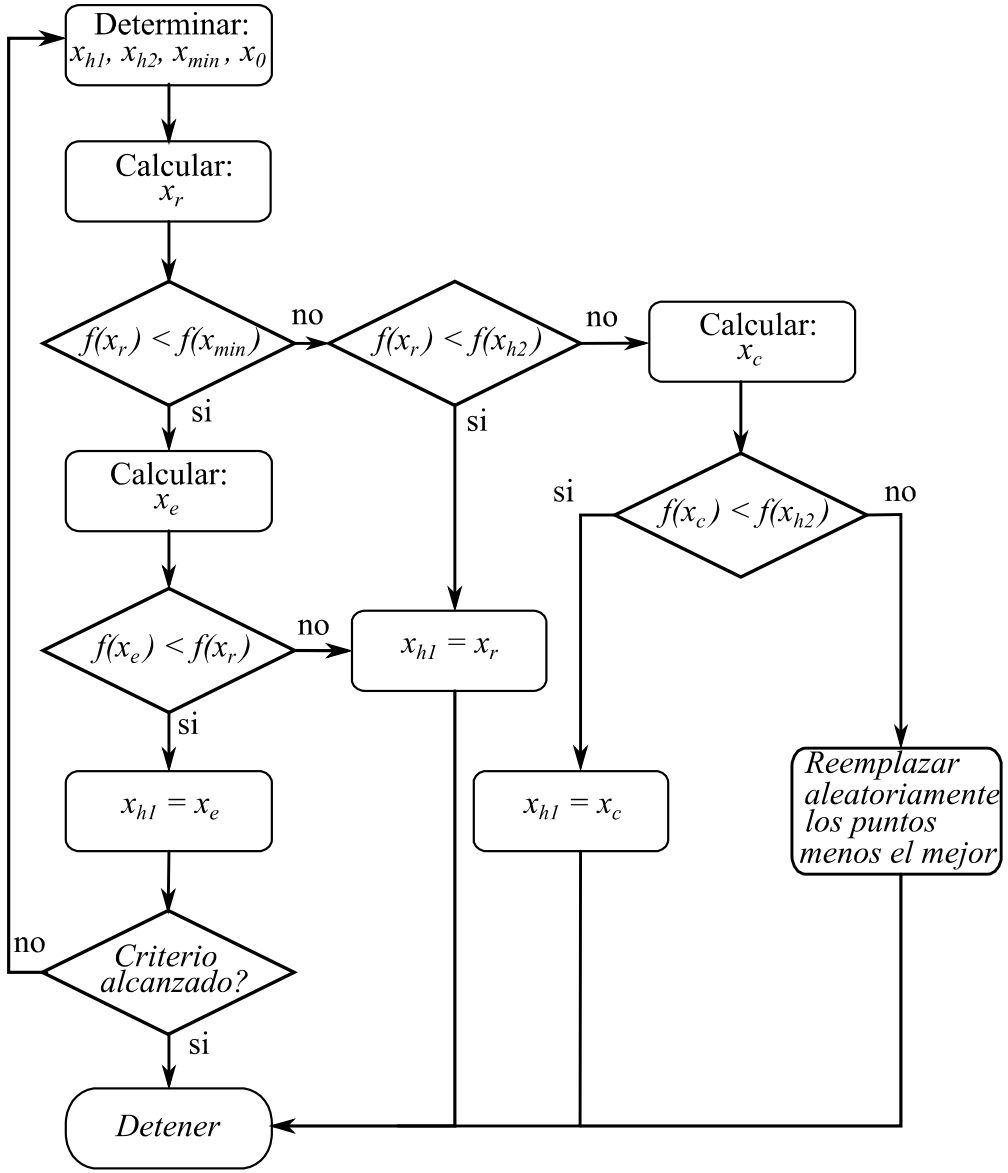
Redes Neuronales Artificiales

NELDER – MEAD

- También llamado Simplex:
  - Se definen $n + 1$ puntos.
  - Se definen tres operaciones:
    - $x_r = x_0 + \alpha(x_0 - x_{h1})$ (reflexión)
    - $x_e = x_0 + \gamma(x_r - x_0)$ (expansión)
    - $x_c = x_0 + \beta(x_{h1} - x_0)$ (contracción)

REDES NEURONALES ARTIFICIALES

› Nelder – Mead



Redes Neuronales Artificiales

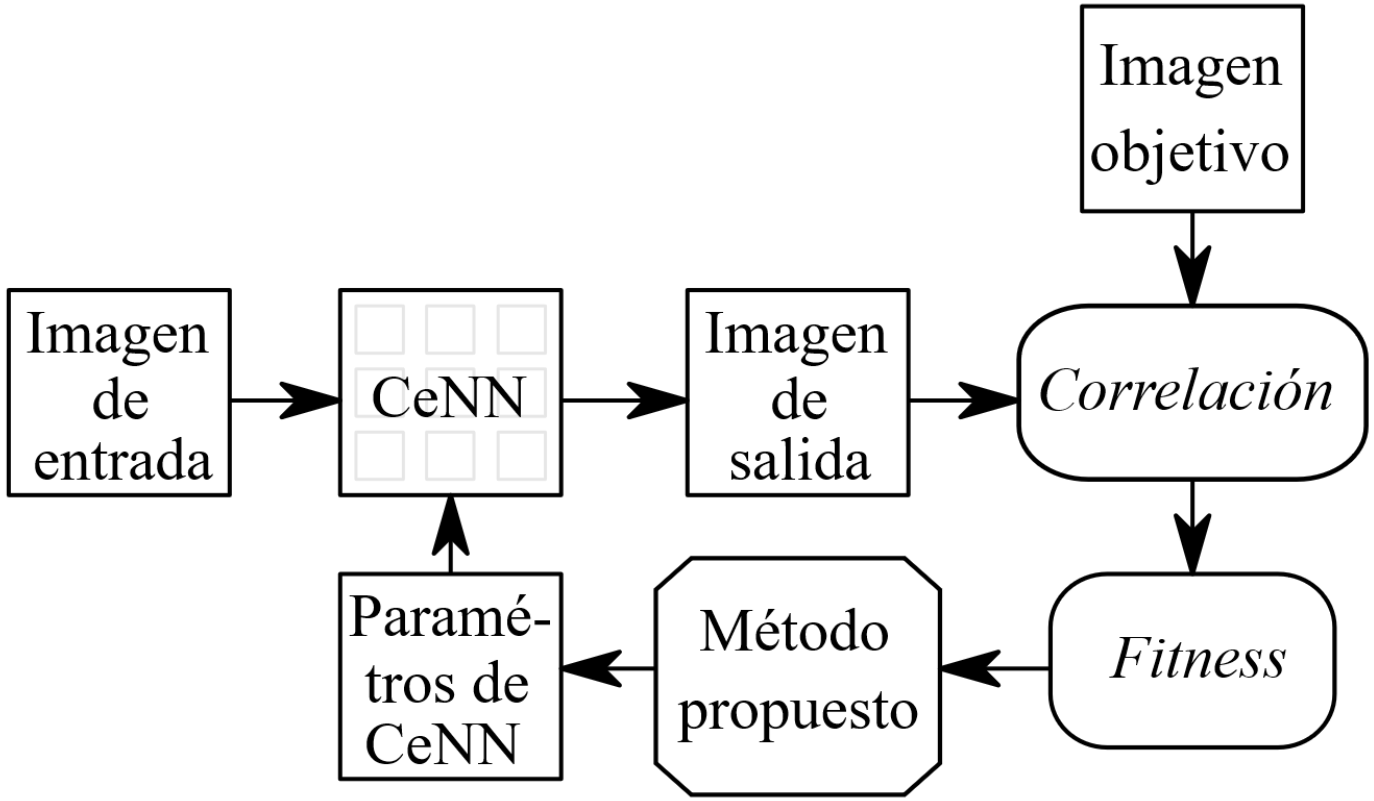
› ABC y NM Híbrido

Algoritmo Híbrido NM – ABC

```
Asignar los parámetros del ABC y NM
Inicializar el enjambre de abejas del ABC
Calcular la calidad inicial de las fuentes de alimento iniciales de EB
Mientras el criterio de parada no es alcanzado hacer:
    Enviar a las EB a las fuentes de alimento y calcular su calidad
    Aplicar la selección del mejor
    Mientras OBs hacer:
        Aplicar NM utilizando OBs
        Aplicar la selección del mejor
    Fin Mientras
    Enviar a las SBs en búsqueda de nuevas fuentes de alimento y calcular su calidad.
Esto es realizado en función de Lim
    Memorizar la mejor fuente de alimento hasta el momento
Fin mientras
```

REDES NEURONALES ARTIFICIALES

› Entrenamiento de la Red Neuronal



Redes Neuronales Artificiales

ABC Y NM HÍBRIDO: BÚSQUEDA DE CONTORNOS

Ciclos	Nelder-Mead		ABC		Propuesta	
	C	$fit(C)$	C	$fit(C)$	C	$fit(C)$
100	0.6602	0.1023	0.6860	0.0931	0.8500	0.0405
500	0.8625	0.0368	0.7644	0.0667	**0.9006**	**0.0261**

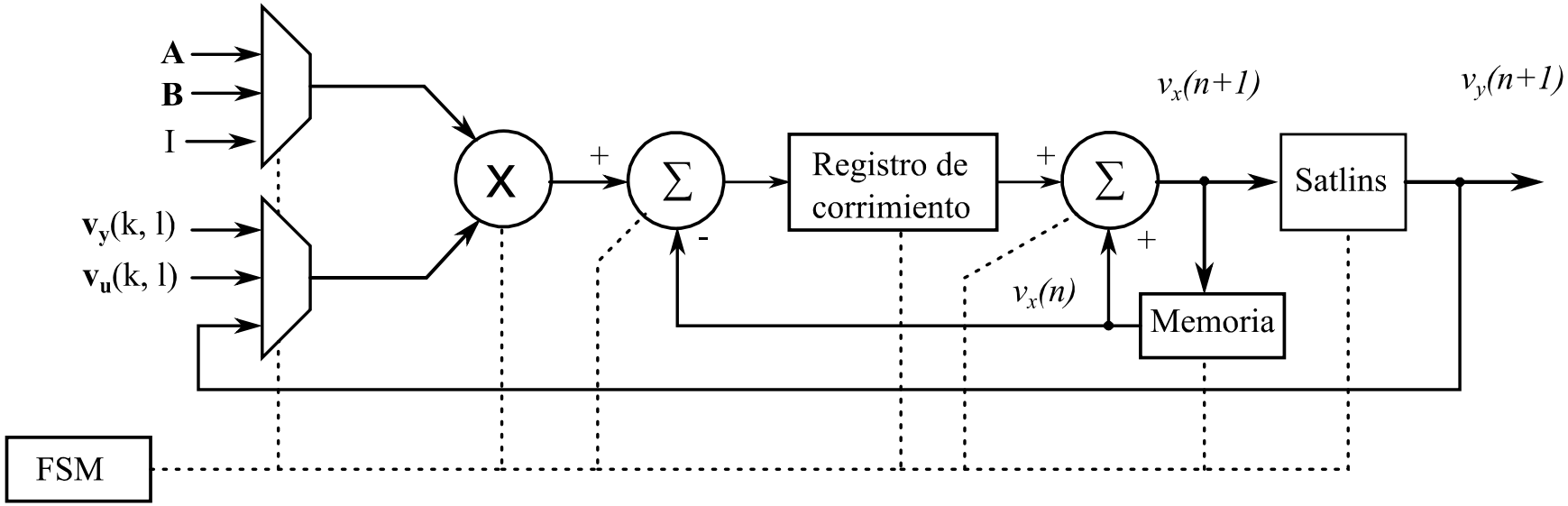
Redes Neuronales Artificiales

ABC Y NM HÍBRIDO: REMOVEDOR DE RUIDO

Ciclos	Nelder-Mead		ABC		Propuesta	
	C	$fit(C)$	C	$fit(C)$	C	$fit(C)$
100	0.8616	0.0371	0.8691	0.0350	0.8976	0.0269
500	0.8982	0.0268	0.8535	0.0309	**0.9000**	**0.0263**

Redes Neuronales Artificiales

IMPLEMENTACIÓN EN FPGA



Redes Neuronales Artificiales

IMPLEMENTACIÓN EN FPGA

	Slice (LUTs) (63400)	Slice (Registers) (126800)	Slice (15850)	Block RAM (135)	DSPs (240)
CeNN	17601	8748	5660	32	64
Cell	215	114	111	0	1

Contenido

- Introducción a la criptografía y redes neuronales artificiales.
- Redes Neuronales Artificiales.
- **Criptografía y SCA sobre dispositivos programables.**
- Criptografía con Curva Elíptica y su implementación en FPGA.
- Análisis del cifrado de imágenes utilizando RSA y ECC.

Criptografía y SCA sobre dispositivos programables

RSA

- Desarrollado por Rivest, Shamir y Adleman.
- Para el cifrado y descifrado se utiliza la exponenciación modular:

$$c = m^e \bmod N$$

Criptografía y SCA sobre dispositivos programables

RSA

Algoritmo Square and Multiply Always, Left to Right

Entrada: m, d, N **con** $d = (d_{n-1}, \cdots, d_0)_2$

Salida: $S = m^d \bmod N$

1. $S[0] \leftarrow 1$
2. **Para** i **desde** $n-1$ **hasta 0 hacer:**
3. $\quad S[0] \leftarrow S^2 \bmod N$
4. $\quad S[1] \leftarrow S[0] \cdot m \bmod N$
5. $\quad S \leftarrow S[d_i] \bmod N$
6. **Fin Para**
7. **Devolver** (S)

Criptografía y SCA sobre dispositivos programables

ATAQUE N – 1

› Dos observaciones:

1. La primera observación es que $(N-1)^2 \equiv 1 \bmod N$, esto puede ser extendido al hecho de que $(N-1)^j \equiv 1 \bmod N$ para j como número par.

2. La segunda observación es similar a la anterior, $(N-1)^k \equiv (N-1) \bmod N$ para k como número impar.

Criptografía y SCA sobre dispositivos programables

CONTRAMEDIDA AL ATAQUE N – 1

Algoritmo modificado de SaMAL2R.

Entrada: m, d, N **con** $d = (d_{n-1}, \cdots, d_0)_2$

Salida: $S = m^d \bmod N$

1. $S \leftarrow 1$
2. $M \leftarrow m \cdot m \bmod N$
3. **Para** i **desde** $n-1$ **hasta 1 hacer:**
4. $\quad R[0] \leftarrow S^2 \bmod N$
5. $\quad R[1] \leftarrow R[0] \cdot M \bmod N$
6. $\quad S \leftarrow R[d_i] \bmod N$
7. **Fin Para**
8. **Si** $d_0 = 0$ **entonces**
9. $\quad$ **Devolver** S
10. **Si no**
11. $\quad$ **Devolver** $S \cdot m \bmod N$
12. **Fin Si**

Criptografía y SCA sobre dispositivos programables

MULTIPLICACIÓN MODULAR

Multiplicación Montgomery sin resta final.

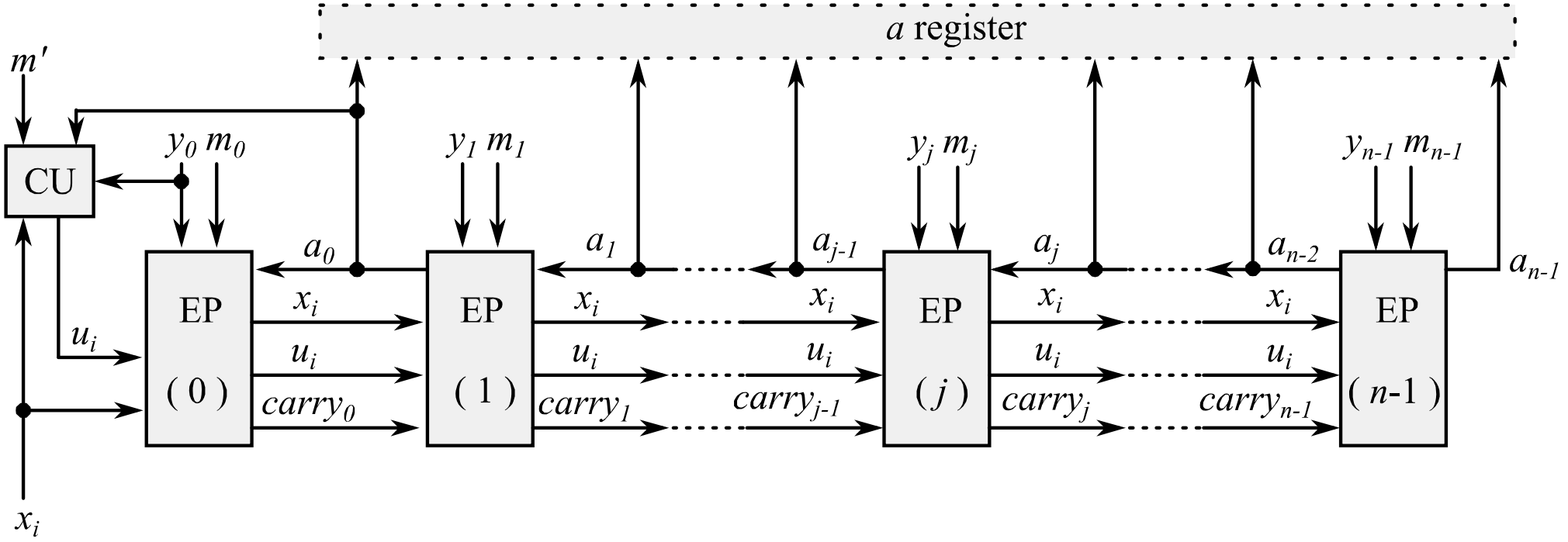
Entrada: $N = (N_{n-1}, \cdots, N_0)_b$, $x = (x_{n-1}, \cdots, x_0)_b$, $y = (y_{n-1}, \cdots, y_0)_b$, **con** $0 \leq x, y \leq 2N$, $2N \leq R = b^n$, **y con** $gcd(N, b) = 1$ **y** $N' = -N^{-1} \bmod b$

Salida: $A = xyR^{-1} \bmod N$

1. $A \leftarrow 0$ **» con** $A = (A_{n-1}, \cdots, A_0)_b$
2. **Para** i **desde 0 hasta** $(n-1)$ **hacer:**
3. $u_i \leftarrow (A_0 + x_i y_0) N' \bmod b$
4. $A \leftarrow (A + x_i y + u_i N)/b$
5. **Fin Para**
6. **Devolver** A

Criptografía y SCA sobre dispositivos programables

ARQUITECTURA SISTÓLICA PARA LA MULTIPLICACIÓN MODULAR



Criptografía y SCA sobre dispositivos programables

EXPONENCIACIÓN MODULAR PARA FPGA

Algoritmo modificado de SaMAL2R.

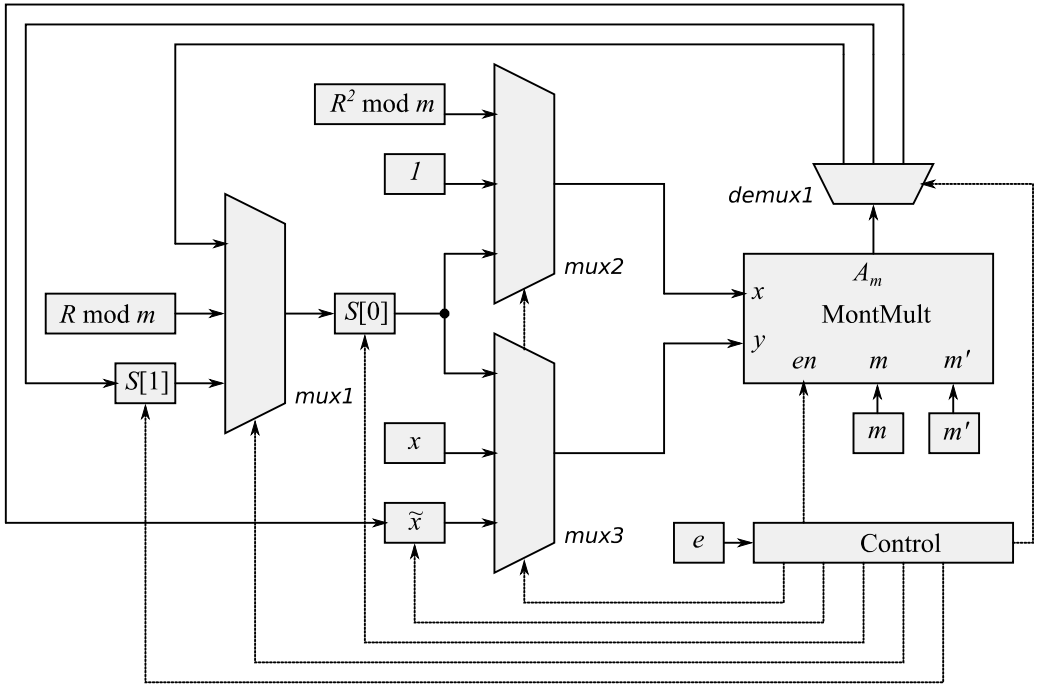
Entrada: $N = (N_{n-1}, \cdots, N_0)_b$, $m = (m_{n-1}, \cdots, m_0)_b$, $d = (d_t, \cdots, d_0)_2$ con $m < 2N < R = b^n$ y con $\gcd(N, b) = 1$ y $N' = -N^{-1} \bmod b$

Salida: $S = m^d \bmod N$

1. $S \leftarrow R \bmod N$
2. $M[0] \leftarrow MultMont(m, R^2, N)$
3. $M[1] \leftarrow MultMont(M[0], M[0], N)$
4. **Para** i **desde** $(n-1)$ **hasta 1 hacer:**
5. $\quad R[0] \leftarrow MultMont(S, S, N)$
6. $\quad R[1] \leftarrow MultMont(R[0], M[1], N)$
7. $\quad S \leftarrow R[d_i] \bmod N$
8. **Fin Para**
9. $S \leftarrow MultMont(S, M[0], N)$
10. $S \leftarrow MultMont(S, 1, N)$
11. **Devolver** S

Criptografía y SCA sobre dispositivos programables

DIAGRAMA A BLOQUES PARA LA EXPONENCIACIÓN MODULAR PARA FPGA



Criptografía y SCA sobre dispositivos programables

RECURSOS UTILIZADOS EN EL FPGA PAR LA MULTIPLICACIÓN Y EXPONENCIACIÓN MODULAR

	Slice (LUTs) (63400)	Slice (Registers) (126800)	Slice (15850)	BRAM (135)	DSPs (240)
SaMAL2R	7,587	9,499	4,253	0	66
⇒ Multiplicación	6,736	5,337	3,429	0	66
SaMAL2R Modificado	9,803	10,568	4,485	0	66
⇒ Multiplicación	9,714	5,337	3,970	0	66

Criptografía y SCA sobre dispositivos programables

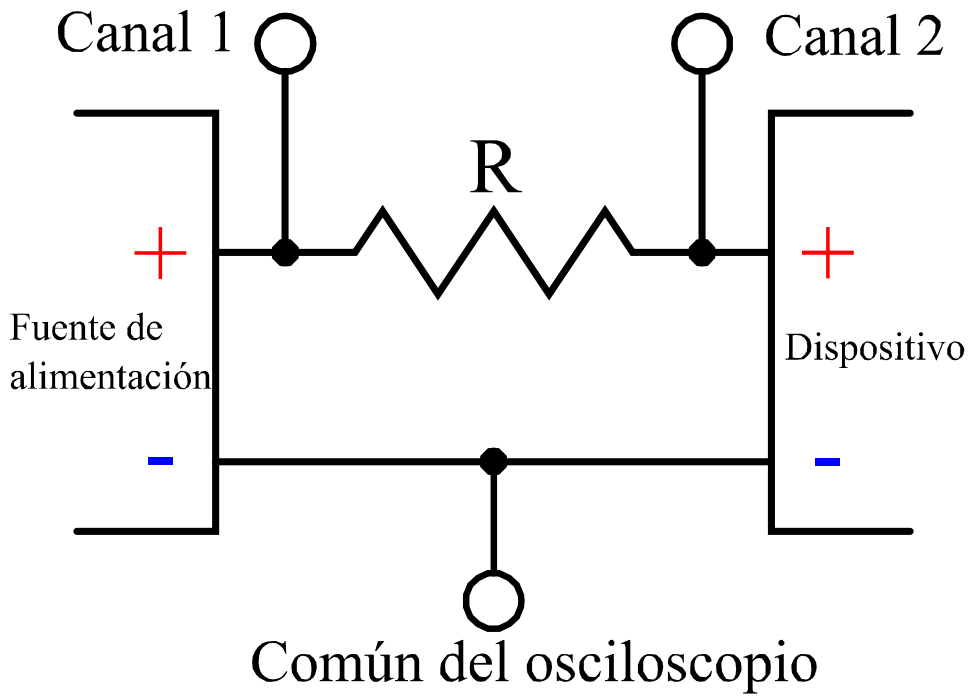
ESTADO DEL ARTE PARA LA EXPONENCIACIÓN MODULAR CON N = 1024

Trabajo	FPGA	LUT	FF	BRAM	Slices	DSP	Frec. (MHz)	Latencia (Ciclos)
Wangchen	Virtex-6	11,834	-	32/2[1]	3,470	18	-	822 por FMLM
Vankatesh	Virtex-7	128,814	75,839	132[2]	37,134	-	200	14,813
Somnath	Artix-7	19.77%	7.81%	7.81%	-	-	90.9	-
Propuesta	Artix-7	9,803 (15.46%)	10,568 (8.33%)	0 (0%)	4,485 (28.3%)	66 (27.5%)	100	399,113

[1] 32 bloques de 32 kb y 2 bloques de 18 kb. [2] 132 bloques del total. – No mencionado.

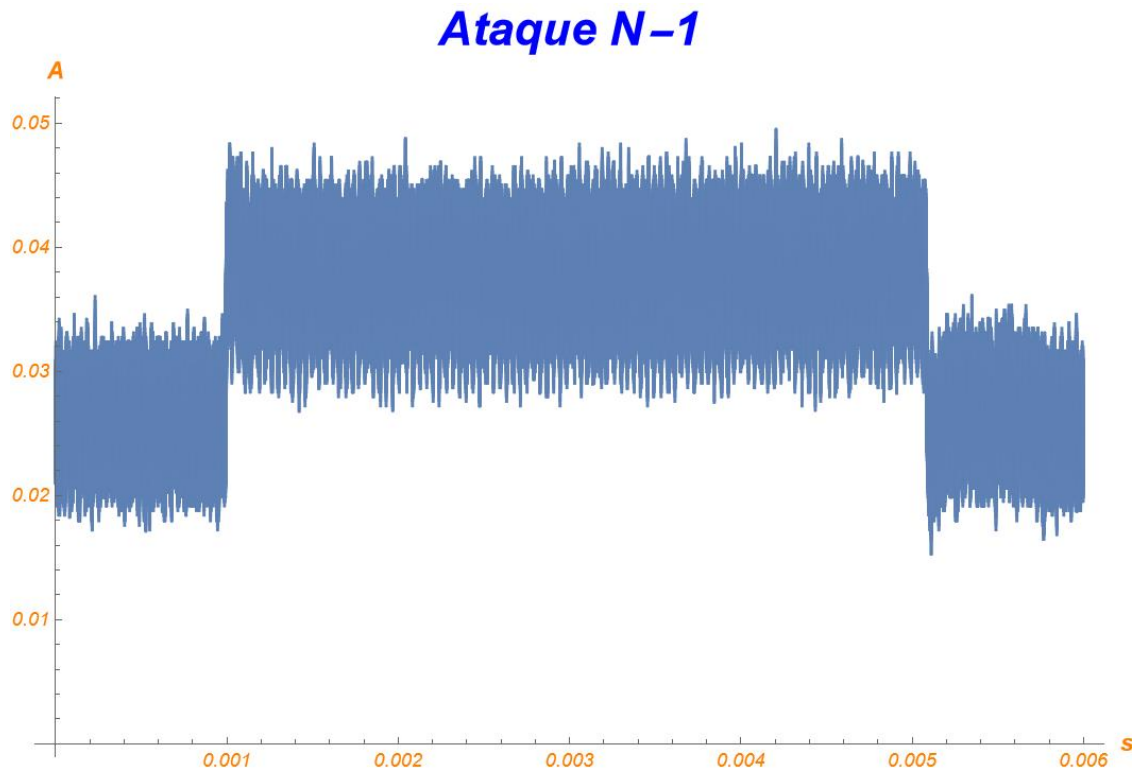
Criptografía y SCA sobre dispositivos programables

ANALISIS DE POTENCIA SIMPLE



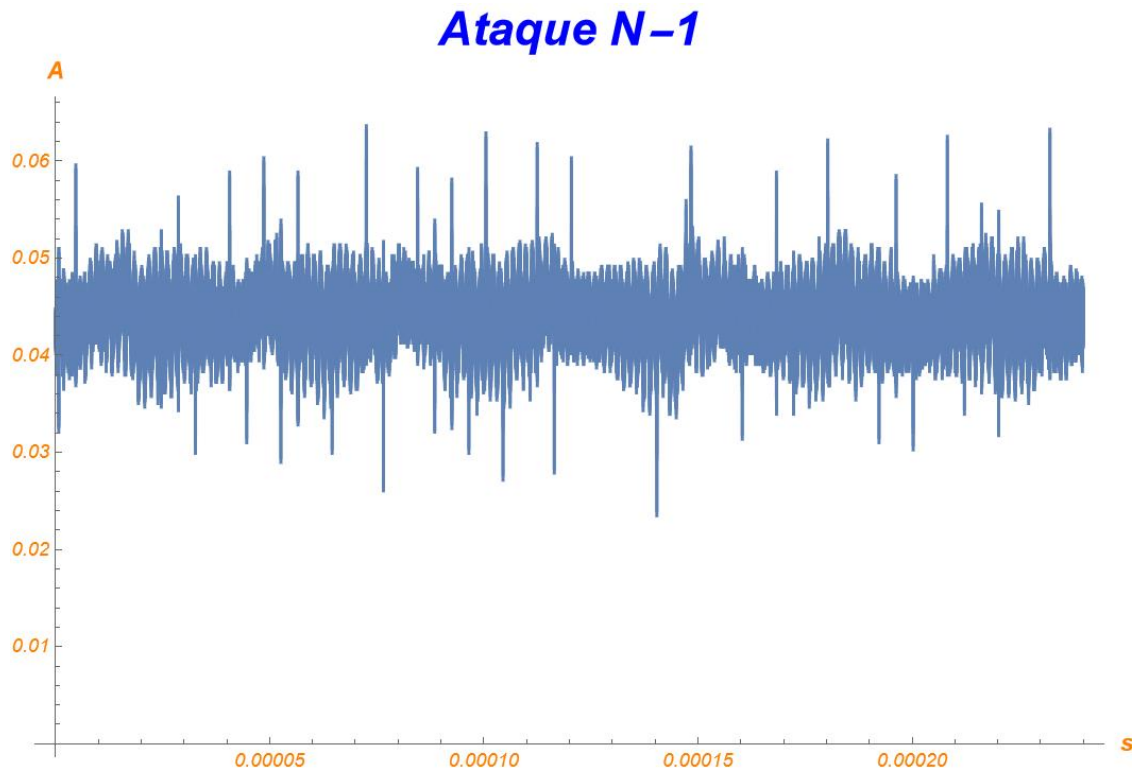
Criptografía y SCA sobre dispositivos programables

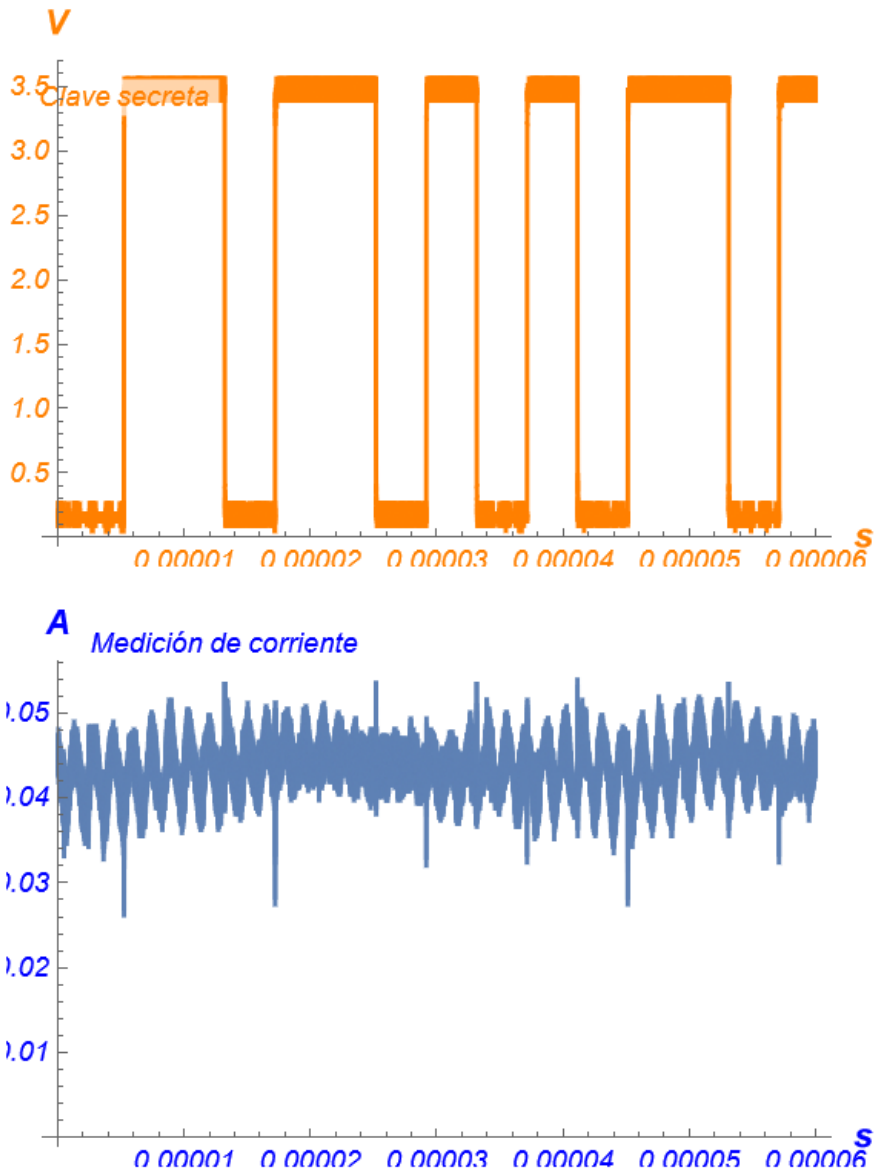
ANALISIS DE POTENCIA SIMPLE



Criptografía y SCA sobre dispositivos programables

ANÁLISIS DE POTENCIA SIMPLE



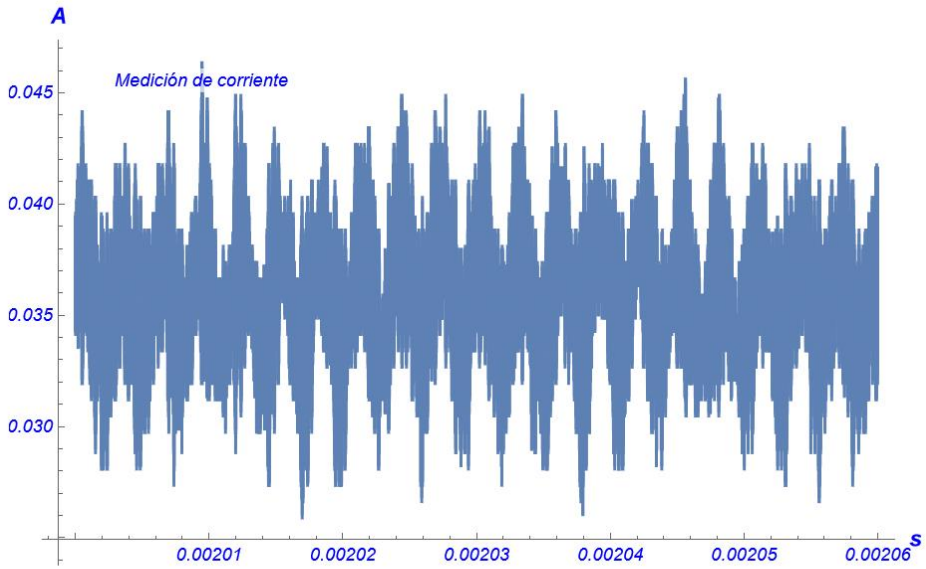


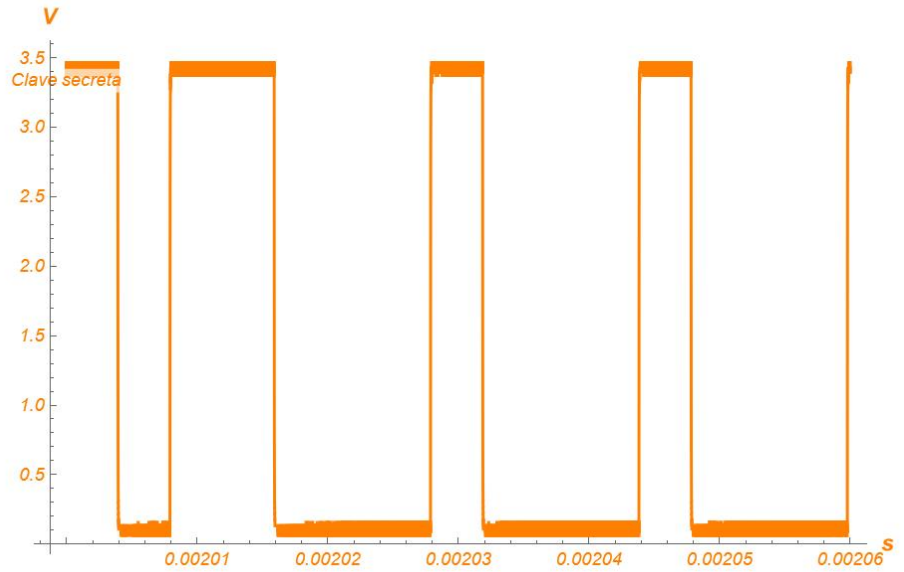
CRIPTOGRAFÍA Y SCA SOBRE DISPOSITIVOS PROGRAMABLES

› Análisis de potencia simple

CRIPTOGRAFÍA Y SCA SOBRE DISPOSITIVOS PROGRAMABLES

› Análisis de potencia simple vs la propuesta de tesis





Contenido

- Introducción a la criptografía y redes neuronales artificiales.
- Redes Neuronales Artificiales.
- Criptografía y SCA sobre dispositivos programables.
- **Criptografía con Curva Elíptica y su implementación en FPGA.**
- Análisis del cifrado de imágenes utilizando RSA y ECC.

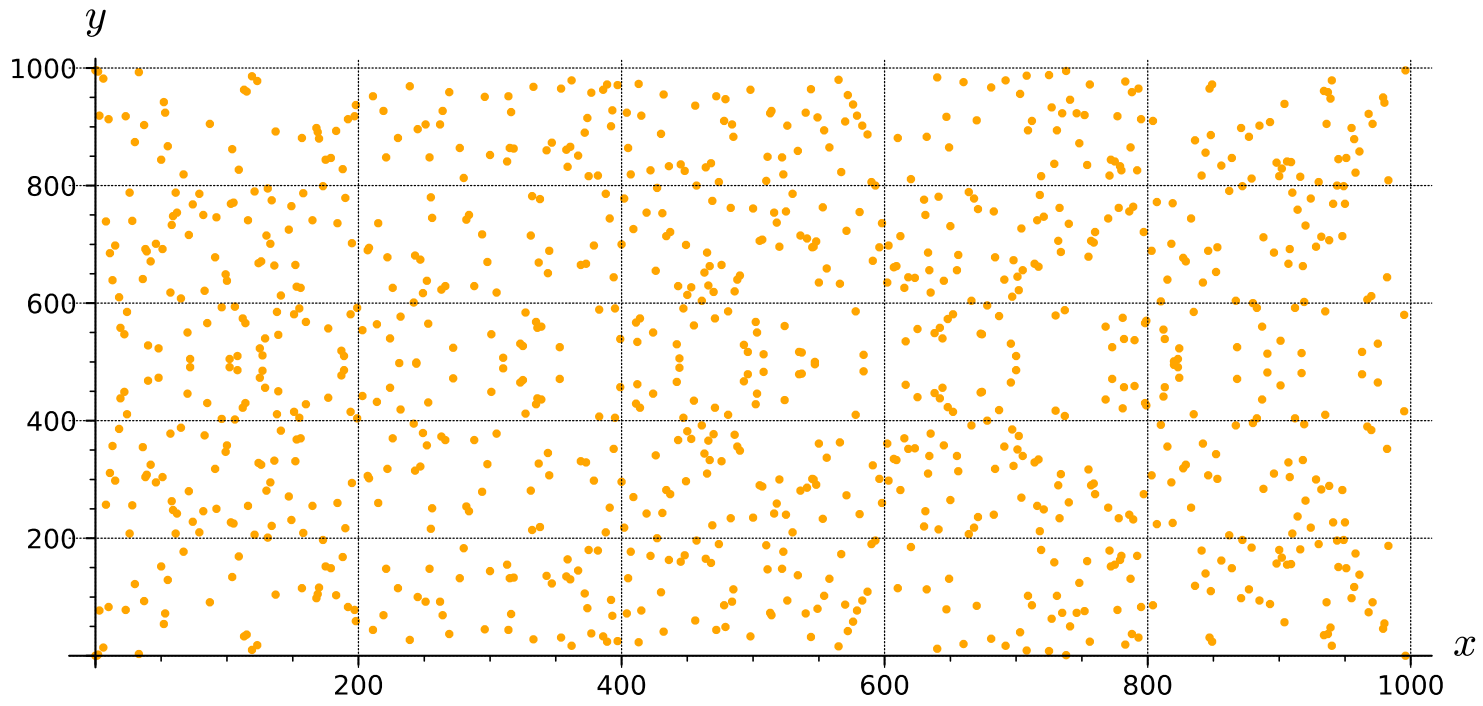
Criptografía con Curva Elíptica y su implementación en FPGA

ECUACIÓN DE WEIERSTRASS

$$E(\mathbb{F}_p): y^2 + xy = x^3 + ax^2 + b$$

Criptografía con Curva Elíptica y su implementación en FPGA

CURVA ELÍPTICA DEFINIDA EN $y^2 + y = x^3 + 996x$ SOBRE $\mathbb{Z}_{997}$



Criptografía con Curva Elíptica y su implementación en FPGA

CIFRADO DE DATOS UTILIZANDO ECC

$$Q = kP$$

Algoritmo doblar y sumar de izquierda a derecha.

Entrada: Un punto $P \in E(\mathbb{F}_p)$,$k = (k_{n-1}, \cdots, k_0)_2 \in \mathbb{N}$

Salida: $Q = kP$

Hacer $R = P$

Hacer $Q = \begin{cases} O & k_0 = 0 \\ P & k_0 = 1 \end{cases}$

Para $i = 1$ hasta t hacer:

  Hacer $R = 2R$

  Si $n_i = 1$ hacer $Q = Q + R$

Fin para

Regresar Q

Criptografía con Curva Elíptica y su implementación en FPGA

SCA SOBRE ECC

› Los ataques son similares a los realizados en RSA.

Criptografía con Curva Elíptica y su implementación en FPGA

PROTECCIÓN DE LOS SISTEMAS ECC

Multiplicación de puntos de Montgomery (para curvas elípticas sobre $\mathbb{F}_{2^m}$).

Entrada: $k = (k_{t-1}, \cdots, k_1, k_0)_2$ **con** $k_{t-1} = 1$, $P = (x, y) \in E(\mathbb{F}_{2^m})$

Salida: $Q = kP$

1. $X_1 \leftarrow x$, $Z_1 \leftarrow 1$, $X_2 \leftarrow x^4 + b$, $Z_2 \leftarrow x^2$ **»{Calcula** $(P, 2P)$**}**
2. **Para** i **desde** $t - 2$ **hasta 0 hacer:**
3. **Si** $k_i = 1$ **entonces:**
4. $T \leftarrow Z_1$, $Z_1 \leftarrow (X_1 Z_2 + X_2 Z_1)^2$, $X_1 \leftarrow x Z_1 + X_1 X_2 T Z_2$.
5. $T \leftarrow X_2$, $X_2 \leftarrow X_2^4 + b Z_2^4$, $Z_2 \leftarrow T^2 Z_2^2$.
6. **Sino:**
7. $T \leftarrow Z_2$, $Z_2 \leftarrow (X_1 Z_2 + X_2 Z_1)^2$, $X_2 \leftarrow x Z_2 + X_1 X_2 T Z_1$.
8. $T \leftarrow X_1$, $X_1 \leftarrow X_1^4 + b Z_1^4$, $Z_1 \leftarrow T^2 Z_1^2$.
9. $x_3 \leftarrow X_1 / Z_1$
10. $y_3 \leftarrow (x + X_1/Z_1)[(X_1 + x Z_1)(X_2 + x Z_2) + (x^2 + y)(Z_1 Z_2)](x Z_1 Z_2)^{-1} + y$

Criptografía con Curva Elíptica y su implementación en FPGA

IMPLEMENTACIÓN DE LA MULTIPLICACIÓN ESCALAR

Multiplicación en campo en $\mathbb{F}_{2^m}$, Derecha a Izquierda, Rotar y Sumar.

Entrada: Polinomio $a(z)$ y $b(z)$ de grado al menos $m-1$

Salida: $c(z) = a(z) \cdot b(z) \bmod f(z)$

1. **Si** $a_0 = 1$ **entonces** $c \leftarrow b$ **sino** $c \leftarrow 0$
2. **Para** $i = 1$ **hasta** $m-1$ **hacer:**
3. $\quad b \leftarrow b \cdot z \bmod f(z)$
4. $\quad$ **Si** $a_i = 1$ **entonces** $c \leftarrow c + b$
5. **Regresar** (c)

Multiplicador MSB (Most significant bit first) para $\mathbb{F}_{2^m}$.

Entrada: $a = (a_{m-1}, \cdots, a_1, a_0)$, $b = (b_{m-1}, \cdots, b_1, b_0)$, **y un polinomio de reducción** $f(x) = z^m + r(z)$.

Salida: $a \cdot b$

1. **Hacer** $c \leftarrow 0$
2. **Para** i **desde** $m-1$ **hasta** 0 **hacer:**
3. $\quad c \leftarrow (c \ll 1) \oplus (c_{m-1} \& r)$
4. $\quad c \leftarrow c \oplus (b_i \& a)$

$$a(z) = a_{m-1}z^{m-1} + \cdots + a_2 z^2 + a_1 z + a_0$$

$$a(z)^2 = a_{m-1}z^{2m-2} + 0 + \cdots + a_2 z^4 + 0 + a_1 z^2 + 0 + a_0$$

Implementación en FPGA del cuadrado

Criptografía con Curva Elíptica y su implementación en FPGA

IMPLEMENTACIÓN DE LA MULTIPLICACIÓN ESCALAR

Reducción rápida modulo $f(z) = z^{163} + z^7 + z^6 + z^3 + 1$.

Entrada: Polinomio $a(x)$ con 325 bits.

Salida: $r(x)$ con 163 bits

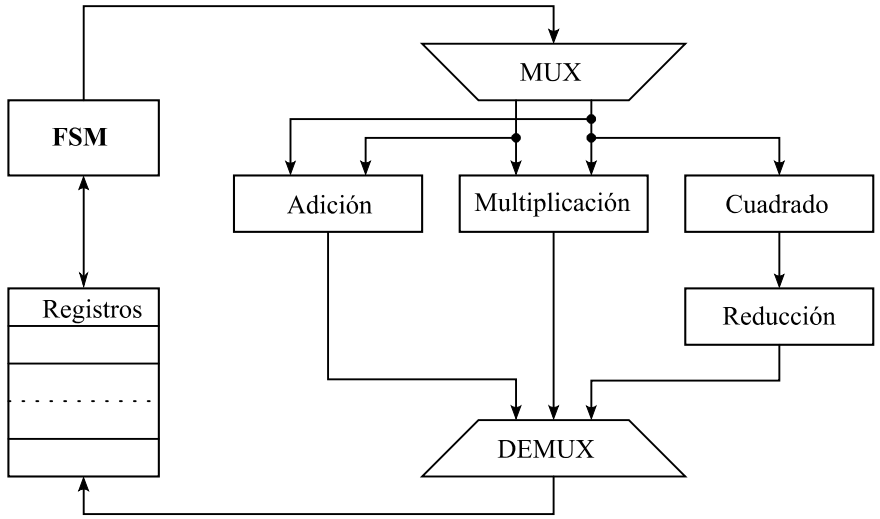
$M \leftarrow a[i] \oplus a[i+163] \oplus a[i+319]$.

$W \leftarrow a[i] \oplus a[i+157] \oplus a[i+160]$.

1. **Para** $i = 0$ **hasta** 162 **hacer:**
2. **Si** $0 \leq i \leq 1$ **entonces:** $r[i] \leftarrow M \oplus a[i+320] \oplus a[i+323]$
3. **Si** $i == 2$ **entonces:** $r[i] \leftarrow M \oplus a[i+320]$
4. **Si** $3 \leq i \leq 5$ **entonces:** $r[i] \leftarrow M \oplus a[i+160] \oplus a[i+316] \oplus a[i+317]$
5. **Si** $i == 6$ **entonces:** $r[i] \leftarrow W \oplus a[i+163] \oplus a[i+313] \oplus a[i+314] \oplus a[i+316]$
6. **Si** $7 \leq i \leq 10$ **entonces:** $r[i] \leftarrow W \oplus a[i+156] \oplus a[i+163] \oplus a[i+312] \oplus a[i+314]$
7. **Si** $11 \leq i \leq 12$ **entonces:** $r[i] \leftarrow W \oplus a[i+156] \oplus a[i+163] \oplus a[i+312]$
8. **Si** $13 \leq i \leq 161$ **entonces:** $r[i] \leftarrow W \oplus a[i+156] \oplus a[i+163]$
9. **Si** $i = 162$ **entonces:** $r[i] \leftarrow W \oplus a[i+156]$

Criptografía con Curva Elíptica y su implementación en FPGA

DIAGRAMA A BLOQUES DE LA MULTIPLICACIÓN ESCALAR



Criptografía con Curva Elíptica y su implementación en FPGA

PARÁMETROS DE LA CURVA ELÍPTICA B – 163

B-163: $m = 163, f(z) = z^{163} + z^7 + z^6 + z^3 + 1, a = 1, h = 2$
$S = 0 \times 85E25BFE\ 5C86226C\ DB12016F\ 7553F9D0\ E693A268$
$b = 0 \times 00000002\ 0A601907\ B8C953CA\ 1481EB10\ 512F7874\ 4A3205FD$
$n = 0 \times 00000004\ 00000000\ 00000000\ 000292FE\ 77E70C12\ A4234C33$
$x = 0 \times 00000003\ F0EBA162\ 86A2D57E\ A0991168\ D4994637\ E8343E36$
$y = 0 \times 00000000\ D51FBC6C\ 71A0094F\ A2CDD545\ B11C5C0C\ 797324F1$

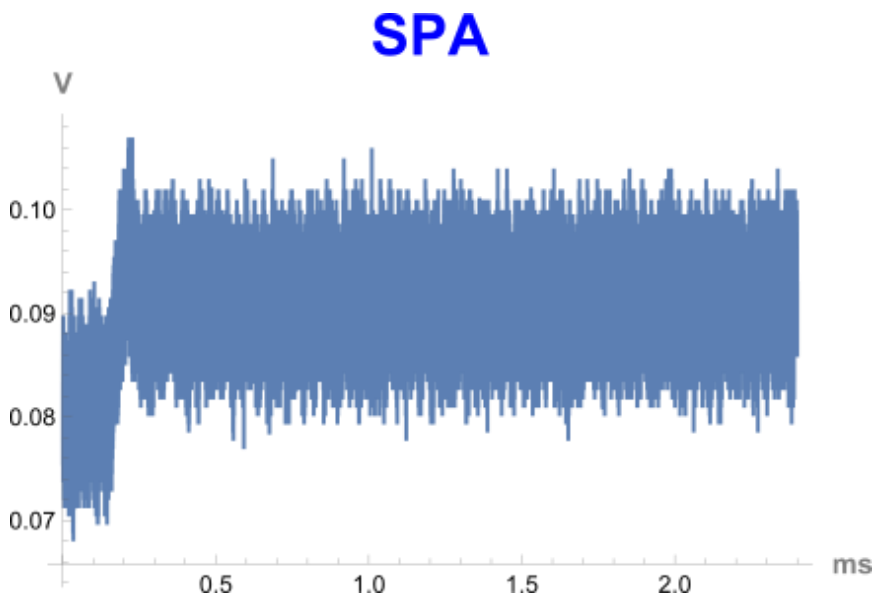
Criptografía con Curva Elíptica y su implementación en FPGA

RECURSOS UTILIZADOS PARA LA IMPLEMENTACIÓN DE LA ECC B – 163

	Slice (LUTs) (63400)	Slice (Registers) (126800	Slice (15850)	BRAM (135)	DSPs (240)
ECC B-163	2,525	2,446	954	0	0
⇒ Multiplicador MSB	778	362	335	0	0

Criptografía con Curva Elíptica y su implementación en FPGA

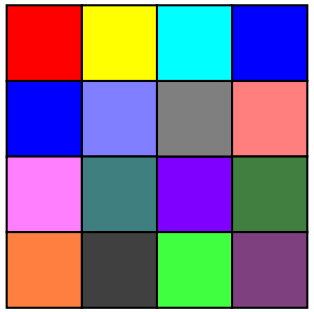
ATAQUES

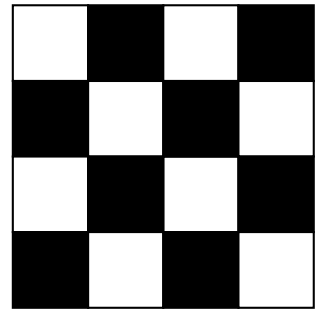


Contenido

- Introducción a la criptografía y redes neuronales artificiales.
- Redes Neuronales Artificiales.
- Criptografía y SCA sobre dispositivos programables.
- Criptografía con Curva Elíptica y su implementación en FPGA.
- **Análisis del cifrado de imágenes utilizando RSA y ECC.**

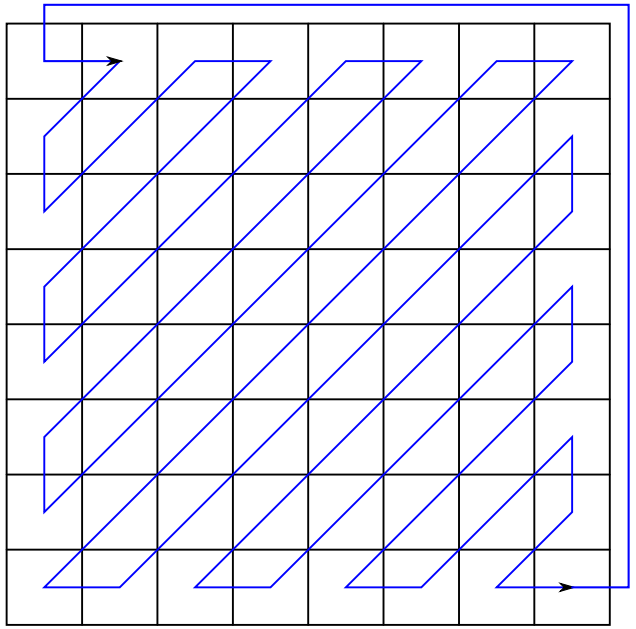
Análisis del cifrado de imágenes utilizando RSA y ECC





Análisis del cifrado de imágenes utilizando RSA y ECC

CODIFICACIÓN DE IMÁGENES PARA CIFRADO



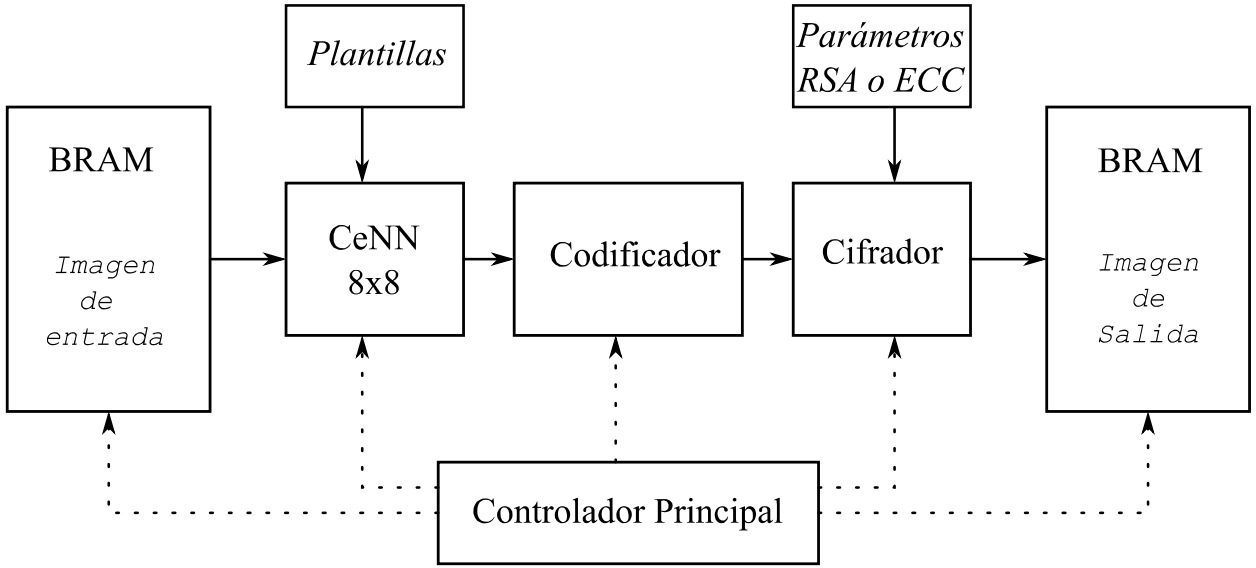
Análisis del cifrado de imágenes utilizando RSA y ECC

CODIFICACIÓN DE IMÁGENES PARA CIFRADO

- Se toman los valores finales de la CeNN.
  - Esta representación numérica está conformada por 18 bits, donde el bit más significativo representa el signo, de tal manera que, un bit 1 representa un valor negativo y un 0 representa un valor positivo, estos valores corresponden a la representación de los pixeles blancos y negros.

Análisis del cifrado de imágenes utilizando RSA y ECC

CONTROLADOR FINAL



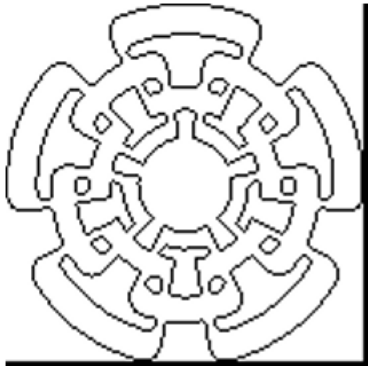
Análisis del cifrado de imágenes utilizando RSA y ECC

USANDO RSA

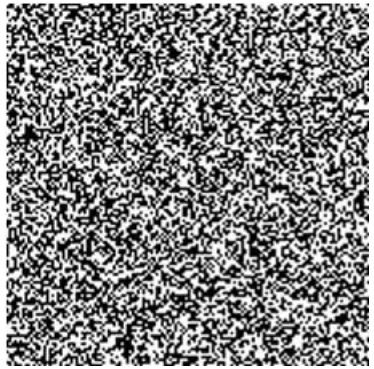
	Slice (LUTs) (63400)	Slice (Registers) (126800)	Slice (15850)	Block RAM (135)	DSP (240)
CeNN	17601	8748	5660	32	64
⇒ Cell	215	114	111	0	1
Codificador	0	64	35	0	0
SaMAL2R Modificado	10803	10557	4548	0	66
⇒ Multiplicación	10429	5353	3904	0	66

Análisis del cifrado de imágenes utilizando RSA y ECC

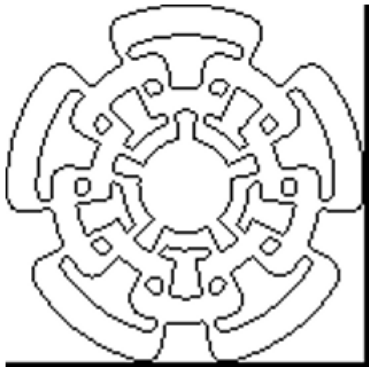
USANDO RSA



Original



Cifrada



Descifrada

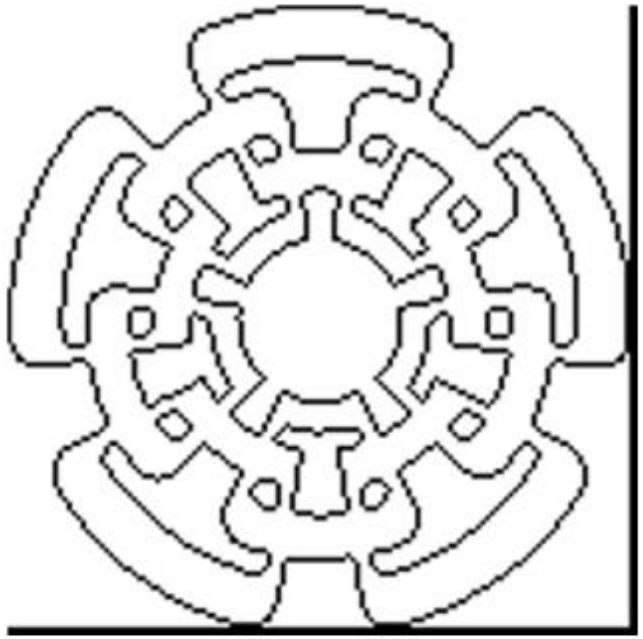
Análisis del cifrado de imágenes utilizando RSA y ECC

USANDO ECC

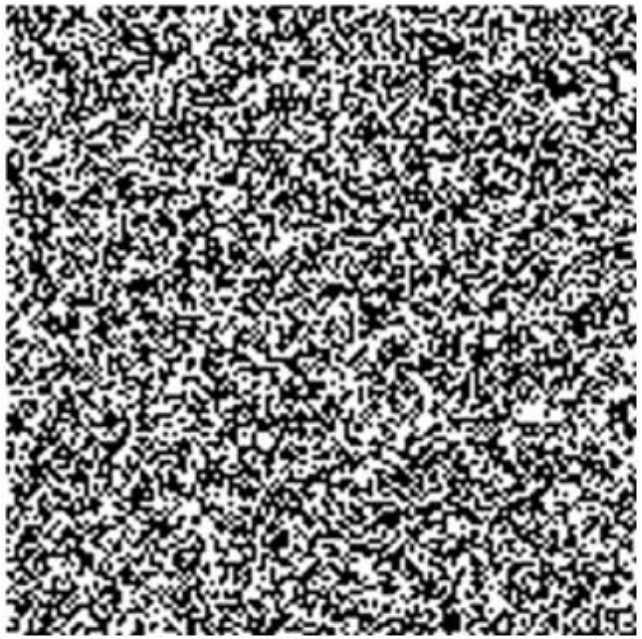
	Slice (LUTs) (63400)	Slice (Registers) (126800)	Slice (15850)	Block RAM (135)	DSP (240)
CeNN	17608	8750	5520	32	64
⇒ Cell	215	114	111	0	1
ECC B-163	2500	2393	967	0	0
⇒ Multiplicación	777	362	359	0	0

Análisis del cifrado de imágenes utilizando RSA y ECC

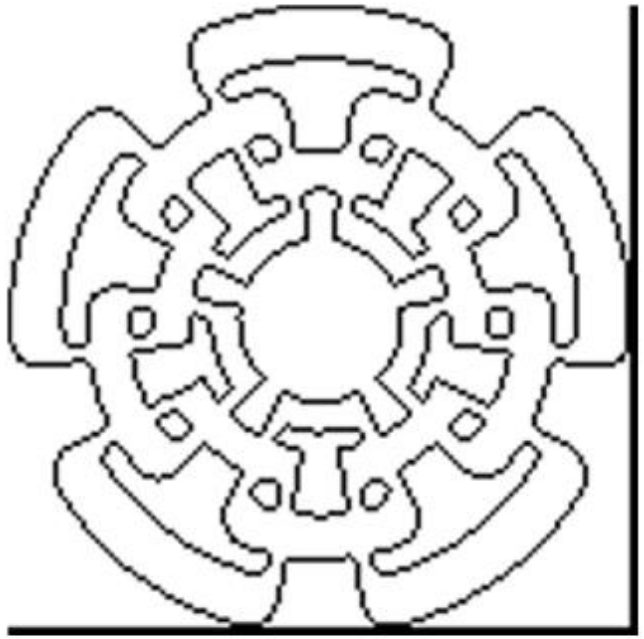
USANDO ECC



(a) Imagen original



(b) Imagen cifrada



(c) Imagen descifrada

Análisis del cifrado de imágenes utilizando RSA y ECC

COMPARATIVA

Implementación		Slices (15850)	BRAM (135)	DSP (240)	Tiempo (ns)
RSA – 1024 bits	CeNN	5741	32	64	-
	Cifrador	4437	0	66	3,991,130
	Total	10888	32	130	-
ECC – 163 bits	CeNN	5520	32	64	-
	Cifrador	967	0	0	70,742,000
	Total	7958	32	64	-

Conclusiones

- El cifrar imágenes, conlleva mucho tiempo de procesamiento.
- El uso de la CeNN reduce la cantidad de datos a cifrar, aumentando la velocidad de cifrado.
- Se propuso una técnica para cifrar imágenes, especialmente para dispositivos con pocos recursos.
- El uso de dispositivos programables, especialmente los dedicados a IoT, son indispensables actualmente, por ello es importante protegerlos.
- El análisis y propuestas realizadas en esta Tesis ayuda a proteger y tomar una mejor decisión al momento de utilizar un método de cifrado en específico.

Trabajo a futuro

- Optimizar las redes neuronales artificiales implementadas en el FPGA.
- Optimizar los sistemas de cifrado y descifrado de datos.
- Utilizar la CeNN como oscilador caótico para tener un sistema de generación de números aleatorios.
- Realizar diferentes ataques, como los basados en inteligencia artificial, sobre la propuesta.
- Mejorar el sistema para poder ser implementado en tarjetas inteligentes.

Gracias por su atención