Implementación de suma y doblado de un punto en una curva de Edwards en un campo binario

Implementation of addition and doubled of a point on a curve in a field Edwards binary

Fabián Velasquez[1], Javier Vargas[2], Sebastián Puente[3]

[1]M.Sc. Matemática Aplicada, [2]M.Sc. Administración Educativa, [3]M.Sc. Ciencias de la Información y las Comunicaciones
[1, 2, 3] Facultad de Ciencias Básicas e Ingeniería, Grupo de Investigación Macrypt, Universidad de los Llanos. Villavicencio, Colombia. Email: fvelasquez@unillanos.edu.co

Recibido 19/04/2016
Aceptado 11/04/2017

Cite this article as: F. Velasquez, J. Vargas, S. Puente, "Implementation of addition and doubled of a point on a curve in a field Edwards binary", Prospectiva, Vol 15, N° 2, 33-39, 2017.

RESUMEN

En este artículo se presentan los resultados del diseño y desarrollo de operaciones de la aritmética de curvas de Edwards en el campo de Galois GF (2251). Se implementaron las operaciones de suma y doblado de puntos en una curva de Edwards binaria basado en la aritmética de campo finito utilizando una base polinomial. La evaluación de las operaciones se realizó sobre un sistema multiprocesador MPSoC, utilizando las capacidades de multiprocesamiento. En las operaciones de la aritmética en curvas de Edward y campos finitos se utilizan algoritmos eficientes y adecuados para un sistema de multiprocesamiento MPSoC Propeller.

Palabras clave: MPSoC; Propeller; Aritmética de curvas de Edwards; Aritmética de campos finitos, Curva binaria de Edwards.

ABSTRACT

This article presents the results of the design and development of operations of arithmetic of curves of Edwards in the field of Galois GF (2251). Is implemented the operations of sum and bent of points in a curve of Edwards binary based on the arithmetic of field finite using a base polynomial. The operations were evaluated on a system multiprocessor MPSoC, using multiprocessing capabilities. Operations of arithmetic in finite fields and Edward curves are used in appropriate and efficient algorithms for a multiprocessing system MPSoC Propeller.

Key words: MPSoC; Propeller; Edwards curve arithmetic; Arithmetic of finite fields; Binary Edwards curve.

Doi: http://dx.doi.org/10.15665/rp.v15i2.716

1. INTRODUCCIÓN

En plena era de la información, las telecomunicaciones ocupan un lugar destacado en las tecnologías de vanguardia, haciendo de la información un producto vital y de un valor incalculable. Desde el ataque a las torres gemelas en el 2001, el concepto de la seguridad informática ocasionó una preocupación profunda en la mayor parte de los gobiernos del mundo, por tanto, se promovió de una forma vertiginosa la investigación en el área de la seguridad informática y específicamente la investigación sobre nuevos algoritmos criptográficos. Diversas ramas de las matemáticas como la teoría de números, la geometría algebraica, entre otras, han permitido el desarrollo de la criptografía moderna, como la aplicación de las curvas elípticas, y de Edward en los criptosistemas de clave pública.

Actualmente la criptografía de curvas elípticas (ECC) es ampliamente usada dentro de los esquemas de clave pública que permiten el intercambio de claves, la firma digital y otros servicios de seguridad informática. Gracias a la complejidad matemática y alto costo computacional que implica romper un criptosistema basado en estas curvas, que son la base de los esquemas criptográficos más robustos en vigencia. La criptografía de curvas elípticas forma parte de los estándares industriales gracias a la alta seguridad que ofrece. Estas curvas junto con los esquemas criptográficos de clave pública suponen un criptosistema altamente confiable.

La teoría de curvas elípticas desarrollada en el siglo pasado, se propone por primera vez en 1986 y 1987 para uso en la criptografía de clave pública. A partir del trabajo de Neil Koblitz [1] y Victor Miller [2] la criptografía basada en curvas algebraicas ha tenido un desarrollo muy vertiginoso hasta nuestros días, y hoy se puede hablar de criptografía basada en curvas de Huff, o criptografía utilizando curvas de Edward.

Las curvas de Edwards surgen como una forma normal de las curvas elípticas, estas curvas tienen la característica de que sus fórmulas para la adición son completas esto quiere decir que son válidas para todo punto de la curva a usar, también estas operaciones pueden hacerse de manera más eficiente en campos binarios, por tanto, son ideales para implementaciones en hardware. Los algoritmos con curvas de Edward están siendo implementados eficientemente en aplicaciones de comercio electrónico, smartcard, votación electrónica y telemática, gobierno digital, bitcoin, notaria digital, entre otras.

Las curvas de Edwards aparecen en el 2007 [3], como un tipo de curva elíptica que soluciona algunas de las debilidades de la criptografía de curvas elípticas, permitiendo un mejor rendimiento computacional en las operaciones de suma y doblado de puntos. La criptografía de curvas elípticas y de curvas de Edwards se puede implementar tanto en software como en hardware, con condiciones y rendimiento que dependen del campo finito. En las implementaciones en hardware es importante minimizar el uso de memoria y los tiempos de ejecución de los algoritmos en un procesador, multiprocesador MPSoC, o en una FPGA, por tanto, se busca el equilibrio entre el tiempo de ejecución y el consumo de memoria. La tecnología MPSoC de la empresa estadounidense Parallax, ha logrado cambiar los conceptos previos de microprocesadores y MPSoC, escalando posiciones en los ámbitos académico y comercial, debido a su bajo costo y las herramientas libres que se ponen a disposición de los desarrolladores. Este dispositivo es un multiprocesador simétrico que cuenta con 8 procesadores de 32 bits llamados COGS, los cuales pueden trabajar por separado o en conjunto dependiendo de la configuración y los requerimientos de la aplicación.

Desde el descubrimiento de las curvas de Edwards en 2007 [3] y de las curvas de Edwards binarias en el 2008 [4], se han presentado algunos resultados tanto en el terreno matemático como en el computacional, que mejoran las condiciones teóricas de esta clase de funciones y su implementación eficiente en diferentes plataformas.

En el año 2009 B. Balwin, R. Moloney, A. Byrne [5] presentaron un análisis hardware de las curvas de Edward trenzadas para el criptosistema de curvas elípticas. En el año 2010 Unal Kocabas, J. Fan y I. Verbauwhethe, presentaron una implementación en hardware de curvas de Edward Binarias [6].

En el año 2011 Hongfeng Wu, Ch. Tang y R. Feng presentaron una reformulación de las curvas de Edward binarias que facilitan una aritmética más rápida. La reformulación de las ecuaciones de aritmética en curvas de Edwards, permiten reducir el número de operaciones necesarias para su implementación [7].

En 2012 M. Li, A. Miri y D. Zhu desarrollaron un algoritmo eficiente para convertir las curvas elípticas ordinarias en curvas de Edward binaria [8]. En 2012 Ayantika Chatterjee y Indranil S. Gupta presentaron una implementación de un procesador reconfigurable en FPGA, para realizar multiplicaciones escalares en curvas de Edwards en el campo GF (2^{233}), con la particularidad de emplear la técnica de point-halving. La plataforma usada es una tarjeta de desarrollo Virtex 4 de Xilinx [9].

En 2013 Chiara Peretti, Paolo Gasltaldo, y Rodolfo Zunino presentaron una implementación de multiplicación escalar en curvas de Edwards y de Jacobi, representa-

das en un campo F_P y usando una plataforma embebida ARM donde optimizan las operaciones aritméticas propias de los campos primos utilizados [10]. En 2013 Andre Himmighofen, Steffen Reith y Berhnard Jungk presentaron una implementación en FPGA del protocolo de Schnorr para autenticación, usando curvas de Edwards representadas en campos primos [11].

En septiembre de 2013 Graham Enos, en su tesis de Doctorado en Matemática Aplicada, aborda la aplicación de criptografía de curvas de Edwards en algoritmos de autenticación e intercambio de claves basado en este tipo de curvas [12].

En 2014 Kwan Ho, Cholk Ok y Christophe Negre, presentaron una reformulación de las curvas de Edward binarias mejorando la eficiencia computacional de las operaciones de adición y doblado de puntos[13].

El grupo Macrypt de la Universidad de los Llanos ha tenido avances en el estudio y aplicación de la aritmética de curvas de Edward que han permitido implementar la operación multiplicación escalar en diferentes plataformas, en este caso el MpPSoC Propeller [14], el microcontrolador MSP430 [15] y FPGA [16]. El trabajo ha continuado optimizando la implementación en Propeller que se presenta en este artículo y se está desarrollando una implementación en FPGA. El presente artículo muestra la implementación de la multiplicación escalar en curvas de Edwards definidas en GF(2^{251}), usando como plataforma el sistema multiprocesador MPSoC Propeller.

2. METODOLOGÍA

La metodología utilizada se desarrolló en 3 fases, denominadas Fase 1, Curvas de Edwards, Fase 2, Aritmética de campos binarios y Fase 3, Aritmética curvas de Edwards binarias. Cada fase se analiza a continuación.

2.1 Curvas de Edwards

Los puntos sobre la curva elíptica en la forma de Weierstrass.

$$y^2+a_1\,xy+a_3\,y = x^3+a_2\,x^2+a_4\,x+a^6 \quad (1)$$

Incluyen no solo los puntos afines , también un punto extra al infinito que sirve como elemento neutro. Las fórmulas estándar para calcular una suma en curvas elípticas P_1+P_2 falla si P_1, P_2 o P_1+P_2 resulta ser el elemento neutro o punto al infinito, o si P_1 es igual a P_2. Cada una de estas posibilidades debería probarse por separado antes de generar algún criptosistema de curva elíptica. Un algoritmo de adición completo es producido por la combinación de varias fórmulas de adición incompletas.

Curvas de Edwards Binaria. Sea k un campo con $char(k) = 2$. Sean elementos de d_1, d_2 con $d_1 \neq 0$ y $d_2 \neq d_1^2 + d_1$, entonces las curvas de Edwards binarias con coeficientes d_1 y d_2 es la curva afín.

$$E_{B,d_1,d_2} = d_1(x+y) + d_2\,(x^2+y^2) = y+xy(x+y)+x^2\,y^2 \quad (2)$$

Esta curva es simétrica en (y_1, x_1) y (0,0) por tanto tiene la propiedad de que es un punto sobre la curva luego es un punto de la curva y el punto es el elemento neutro de la ley de adición (2).

2.2 Aritmética de campos binarios

Las operaciones básicas necesarias en el campo finito para la implementación de aritmética de curvas de Edwards son suma, producto, elevación al cuadrado e inverso multiplicativo. Estas operaciones constituyen la base para la operación multiplicación escalar en curvas elípticas; dependiendo de las características de la implementación y la plataforma empleada, puede implementarse la elevación al cuadrado y en general a cualquier potencia con el mismo multiplicador, mientras que, el inverso multiplicativo que se emplea para realizar divisiones, puede ser remplazado directamente por un divisor.

Los parámetros del campo finito se escogieron teniendo en cuenta las consideraciones de diseño, fortaleza y aplicación en curvas elípticas. El campo definido es ***GF (2^{251})***, en donde los polinomios que lo conforman pueden representarse como palabras binarias de ***251*** bits. En [4] se demuestran las consideraciones de seguridad tenidas en cuenta para construir la curva de Edwards en el campo finito mencionado.

Para el caso del campo finito escogido, se define el polinomio irreducible:

$$P(x) = x^{251} + x^7 + x^4 + x^2 + 1 \quad (3)$$

Las operaciones en el campo finito se realizan teniendo en cuenta la arquitectura de 32 bits del MPSoC Propeller, se representan todos los elementos del campo GF(2^{251}) con 8 longs, colocando los 5 bits de mayor peso del long 7 siempre en 0, realizando las operaciones sobre arreglos contenidos en la memoria RAM compartida.

2.2.1 Suma en GF (2^{251})

La operación adición se realiza operando mediante la función XOR bit a bit (bit-wise XOR), los dos sumandos definidos en campo finito se implementan a través de una función que realiza la XOR posicionalmente, para los 8 longs que conforman cada operando. Esto teniendo en cuenta la coincidencia entra la suma mó-

dulo 2 y la operación XOR. En el algoritmo 1 se presenta la adición en F_2^m.

Algoritmo 1. Adición en F_2^m.

Entrada: Polinomios Binarios a (z) y b (z) de grado máximo m-1

Salida: c (z) =a (z) + b (z).

1. para i desde 0 a t-1 haga
   1.1 C[i]←A[i]⊕B[i]
2. retorne (c)

2.2.2 División en GF (2^{251})

Teniendo en cuenta las características de la plataforma empleada, se decidió implementar directamente la división en GF (2^{251}), aplicando una modificación del algoritmo binario para el cálculo de inversos multiplicativos [17]. En este artículo se muestra que las operación suma y doblado de puntos en curvas de Edwards no requieren directamente la implementación de inversos multiplicativos, pero sí requieren una división. Adicionalmente, se debe tener en cuenta que el tiempo de ejecución de este algoritmo de división es equivalente al original de inversión, por lo cual al implementar la división directamente se ahorra una multiplicación, que tendría que implementarse en el caso de realizar la división b/a como ba^{-1}, primer calculando a^{-1} y luego multiplicando por b, el algoritmo 2, presenta la división en F_2^m.

Algoritmo 2. División en F_2^m.

Entrada: Dos polinomios binarios diferentes de cero de grado máximo m-1.

Salida: $\frac{b}{a} \bmod f = ba^{-1} \bmod f$

1. u←a, v←f
2. g_1←b, g_2←0
3. Mientras que u≠1 Λ v≠1
   3.1 Mientras que z divide a haga
   u← $\frac{u}{z}$
   Si z divide a g_1, entonces g_1← $\frac{g_1}{z}$
   Si no g_1← $\frac{g_{1+f}}{z}$
   3.2 Mientras que z divide v haga
   v← $\frac{v}{z}$
   Si z divide a g_2, g_2← $\frac{g_2}{z}$,
   Si no g_2← $\frac{g_{+f}}{z}$
   3.3 si grado (u) ≥ grado (v)
   Entonces: u←u + v, g_1←g_1+g_2
   Si no: v←u + v, g_2←g_1+g_2,
4. Si u=1 entonces retorne g_1
   Sino retorne g_2

2.2.3 Producto en GF (2^{251})

La operación multiplicación en el campo binario se utilizó el método comb [18], el cual implementa en forma combinada el producto convencional de polinomios, seguido de la reducción correspondiente en el campo finito, módulo el polinomio irreducible. Para la reducción se implementa un algoritmo recursivo, que de izquierda a derecha desplaza el polinomio irreducible y realiza una suma con el resultado, el cual es de un tamaño máximo 2m-2, correspondiente a 16 palabras de 32 bits. Teniendo en cuenta que el polinomio irreducible definido en (1) es fijo, se simplifica la operación reducción modular aprovechando que en cada iteración solamente se afectan 4 bits. En el algoritmo 3 se presenta el método Comb para realizar la multiplicación.

Algoritmo 3. Multiplicación por el método Comb.

Entrada: Polinomios Binarios a (z) y b (z) de grado máximo m-1

Salida: c (z) =a (z). b (z)

1. C←0
2. para k desde 0 a W-1 haga
2.1 para j desde 0 a t-q haga,
   Si la k-esimo bit de A[j] es 1 entonces adicione B a C { j}
2.2 Si k ≠ (W-1) entonces B←B.z.
2.3 Retorne (C)

2.3 Aritmética de curvas de Edwards binarias

Definida una curva de Edwards por la ecuación 4.

$$E_{B,D_1\ D_2}=d_1(x+y)+d_2(x^2+y^2)=xy+xy(x+y)\ x^2\ y^2 \quad (4)$$

Donde $d=d_1=d_2=x^{57}+x^{54}+x^{44}+1$

Entonces la curva se denomina curva binaria de Edwards [7].

El parámetro **d** cumple con una serie de condiciones [6] y se opera como una palabra de 251 bits, en un arreglo de 8 longs.

El punto aleatorio P=(x, y) obtenido de la curva binaria de Edwards tiene las siguientes coordenadas:

x=54CFD57EB2629F4B53313A79DA9E9BD59255D17D720FD706A4BDBA2F219F302h

y=78611FC3B5E535CD6067D7C2BAB2AF668E68F06F4D422427DA66F407B161D7C8h

2.3.1 Suma de puntos

En las curvas de Edwards binarias, E_{B,d_1,d_2}, la ley de adición se representa en la correspondiente curva elíptica en la forma de Weierstrass. Si el denominador $d_1 + (x_1 + x_1^2)(x_2 + y_2)$ y $d_1 + (y_1 + y_1^2)(x_2 + y_2)$ son no cero, entonces la suma (x_3, y_3) es un punto sobre E_{B,d_1,d_2}, por ejemplo, $d_1(x_3 + y_3) + d_2(x_3^2 + y_3^2) = x_3y_3 + x_3y_3(x_3 + y_3) + (x_3^2y_3^2)$.

En el presente trabajo se seleccionó $d_1 = d_2 = d$, por tanto, la suma de los puntos $P = (x_1, y_1)$ y $Q = (x_2, y_2)$ sobre E_{B,d_1,d_2} es el punto $R = (x_3, y_3)$ con las coordenadas:

$$x_3 = \frac{d((x_1+x_2)+(x_1+y_1)(x_2+y_2))+(x_1+x_1^2)(x_2(y_1+y_2+1)+y_1y_2)}{d+(x_1+x_1^2)(x_2+y_2)} \quad (6)$$

$$y_3 = \frac{d((y_1+y_2)+(x_1+y_1)(x_2+y_2))+(y_1+y_1^2)(y_2(x_1+x_2+1)+x_1x_2)}{d+(y_1+y_1^2)(x_2+y_2)} \quad (7)$$

Se implementan las operaciones directamente para el numerador y denominador, por último, se realiza directamente la división; la elevación al cuadrado se realiza con la función multiplicación. El costo computacional es de 13 multiplicaciones, 12 sumas y dos divisiones; esta operación se implementa usando paralelamente los 8 núcleos del dispositivo MPSoC.

2.3.2 Doblado de puntos

Existe un algoritmo de la operación suma, en la cual un operando se suma con sí mismo, para esto se calcula un punto Q igual a 2P. Esta operación se conoce como doblado y tiene menor complejidad computacional que la suma, como se muestra a continuación:

$$x_3 = 1 + \frac{d(1+x_1^2+y_1^2)+y_1^2+y_1^4}{d+x_1^2+y_1^2+x_1^4+y_1^4} \quad (8)$$

$$y_3 = 1 + \frac{d(1+x_1^2+y_1^2)+x_1^2+x_1^4}{d+x_1^2+y_1^2+x_1^4+y_1^4} \quad (9)$$

Estas fórmulas fueron adaptadas de tal forma que se minimizara el número de productos y la expresión del denominador es recursiva para el cálculo de la otra coordenada del punto. El costo computacional es de 8 sumas, 5 productos y dos divisiones; esta operación se implementa usando paralelamente los 8 núcleos del dispositivo MPSoC.

2.3.3 Multiplicación escalar

Una de las aplicaciones de la suma y doblado de puntos es obtener la operación multiplicación escalar muy útil en la criptografía con curvas elípticas y con curvas de Edward. Con las operaciones en el campo binario de suma y duplicación de puntos, se llega a la operación de multiplicación escalar $\boldsymbol{kP = Q}$, a partir del punto aleatorio P y un escalar k que pertenece al campo binario. Para la multiplicación escalar se implementa el algoritmo de Montgomery Ladder [19], el cual se muestra en algoritmo 4.

Algoritmo 4. Método de Montgomery Ladder.

Sea un punto P=(x,y), con k escalar, 0<k<M,
$k = (k_{m-1}, k_{m-2}, \ldots\ldots, k_0)_2, k_{m-1}=1 \;\; P \in E\,(F_q)$
Obtiene Q= [k]P
$P_1 \leftarrow P, \;\; P\_2 \leftarrow 2P_2$
Para i de m — 2 a 0 haga
Si $k_i = 1$ *entonces*
$P_1 \leftarrow P_1 + P_2, P_2 \leftarrow 2P_2$
Si no entonces
$P_2 \leftarrow P_1 + P_2, P_1 \leftarrow 2P_1$
fin si
fin para
Retorne (P_1)

El algoritmo 4, permite realizar la multiplicación escalar, toma el escalar k, lo evalúa bit a bit, considerando siempre el bit de mayor peso como 1 y a partir del resultado de esta evaluación, realiza una suma de puntos y un doblado, variando los parámetros en cada caso dependiendo del valor del bit respectivo.

Al final de la evaluación de k, se obtiene el punto producto; se usa el paralelismo del dispositivo MPSoC Propeller, para implementar en cada ciclo de evaluación de *k* la suma y la duplicación de puntos, en núcleos o cog separados.

3. RESULTADOS Y DISCUSIÓN

Se implementó la multiplicación escalar en 88.8 ms, haciendo uso del paralelismo del MPSoC Propeller, optimizando las operaciones suma y doblado de puntos mediante una librería de multiprocesamiento simétrico desarrollada para tal fin; cada iteración del algoritmo Montgomery Ladder se implementó en forma secuencial, realizando primero la suma y luego el doblado correspondiente. Para la verificación del proceso se utilizó la interfaz de la aplicación Parallax Serial Terminal, con comunicación serial a 115200 bps.

Se realizó la evaluación del tiempo de ejecución de cada una de las funciones implementadas, las cuales fueron desarrolladas en el lenguaje Assembly del MPSoC Propeller con el fin de lograr la mayor eficiencia posible; para determinar el tiempo de ejecución, se implementó un temporizador especial dedicado basado en un contador de 32 bits, el cual se activa al iniciar la

ejecución de una función específica y se detiene al finalizar la misma y con el valor de conteo se establece el tiempo que tardó la ejecución. En el caso de la aritmética en curvas de Edwards, los tiempos de ejecución de la función adición de puntos fue de 0.064 milisegundos, de la duplicación de puntos 0.025 milisegundos y para la multiplicación escalar fue de 88.8 milisegundos. Estos resultados se presentan en la tabla 1.

Tabla 1. Resultados aritmética de curvas de Edwards.
Table 1. Results arithmetic curves Edwards.

OPERACIÓN ARITMÉTICA	Duplicación de puntos	Multiplicación escalar	Adición de puntos
TIEMPO (ms)	0.025	88.8	0.064

La tabla 2 compara los parámetros como tiempo de ejecución, expresado en términos de ciclos de reloj, frecuencia de operación y rendimiento de multiplicación por escalar.

Tabla 2. Tabla comparativa de resultados en aritmética de curvas Edwards en GF (2^{251}).
Table 2. Comparative table of results in arithmetic Edwards curves in GF (2^{251}).

Tecnología	**Campo finito**	**Tiempo multiplicación escalar (ms)**	**Frecuencia reloj (MHz)**	**Ciclos de reloj por multiplicación escalar**	**Rendimiento. Multiplicaciones escalares por segundo**
MPSoC Propeller	GF (2^{251})	88.8	80	7.11×10^6	11.24
Multiprocesador	GF (2^{251})	287	80	23×10^6	3.47
Procesador Core 2 Quad	GF (2^{251})	3.33	2400	3.14×10^5	30000

En la aritmética de curvas de Edward se resalta los tiempos de procesamiento logrados por causa del multiprocesamiento en los núcleos, mejorando los resultados de los procesamientos en microcontroladores de un solo núcleo y con un rendimiento que puede ser usado en diversas aplicaciones como: sistemas de cifrado con autenticación, criptografía basada en identidad, entre otras.

Los algoritmos para las operaciones multiplicación e inverso en el campo finito requieren estudios futuros que permitan desarrollar e implementar métodos que mejoren el tiempo de procesamiento en los dispositivos MPSoC Propeller

4. CONCLUSIONES

Se puede optimizar las operaciones en curvas de Edwards, aprovechando la división de funciones entre los cog del MPSoC Propeller, sin presentarse problemas de dependencias entre recursos, esto es teniendo en cuenta las condiciones de los algoritmos implementados, pero la complejidad del manejo del paralelismo del MPSoC Propeller es bastante alta.

Se mejoraron resultados previamente obtenidos para la multiplicación escalar en curvas binarias de Edwards, usando como plataforma el MPSoC Propeller, de tal forma que este dispositivo puede ser usado en aplicaciones que requieran seguridad basada en el criptosistema de curvas elípticas, con tiempos razonables y bajo costo. Las aplicaciones directas más interesantes son redes inalámbricas de sensores, tema actual de investigación y múltiples usos.

El lenguaje del MPSoC Propeller hace eficiente la aritmética en las funciones que usan multiprocesamiento y acceso a memoria, y el uso eficiente de las funciones específicas en lenguaje Assembly para las operaciones de aritmética en el campo de Galois.

Como trabajo futuro, se propone la implementación de algoritmos en curvas de Edwards binarias en campos finitos de característica tres y la aritmética en base normal o gaussiana.

AGRADECIMIENTOS

Los autores agradecen al Instituto de Investigaciones de los Llanos Orientales (IIOC) por habernos financiado en proyectos relacionados con el área de la criptografía son los proyectos: sistema de voto telemático y criptografía basada en identidad que son la base para el desarrollo de los resultados que se publican en este artículo.

REFERENCIAS

[1] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, 48, 203–209, 1987.

[2] V. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptography-Crypto '85, Springer-Verlag New York, LNCS 218, 1986, pp. 417-426.

[3] D. Bernstein, T. Lange, "Inverted Edwards Coordinates", *Appl. Algebr. Algebr. Algorithms Error-Correcting Codes*, 4851, 20–27, 2007.

[4] S. Ionica, A. Joux, *Another approach to pairing computation in Edwards coordinates*, Prog. Cryptology-INDOCRYPT 2008, 400–413, 2008.

[5] B. Baldwin, R. Moloney, A. Byrne, G. Mcguire, "A Hardware Analysis of Twisted Edwards Curves for an Elliptic Curve Cryptosystem", *Discrete Math.*, 1–14, 2009.

[6] U. Kocabas, J. Fan, I. Verbauwhede, "Implementation of binary edwards curves for very-constrained devices", *2010 21st IEEE Int. Conf. Appl. Syst. Archit. Process.*, pp. 185–191, 2010.

[7] H. Wu, C. Tang, R. Feng, "A New Model of Binary Elliptic Curves", *International Journal of Rock Mechanics and Mining Science*, 42(4), 481–507, 2012.

[8] M. Li, A. Miri, D. Zhu, "Fast Algorithm for Converting Ordinary Elliptic Curves into Binary Edward Form", *International Journal of Digital Content Technology and its Applications*, 6(1), 405–412, 2012.

[9] A. Chatterjee, I. Sen Gupta, FPGA Implementation of Extended Reconfigurable Binary Edwards Curve based Processor, *Work. Comput. Comun.*, pp. 211–215, 2012.

[10] C. Peretti, P. Gastaldo, M. Stramezzi, R. Zunino, "Embedded implementation of Edwards curve- and extended Jacobi quartic curve-based cryptosystems", *8th Int. Conf. internet Technol. Secur. Transations*, pp. 394–400, 2013.

[11] A. Himmighofen, B. Jungk, S. Reith, On a FPGA-based Method for Authentication using Edwards Curves, 2013.

[12] G. Enos, Binary Edwards Curves in Elliptic Curve Cryptography, *A dissertation submitted to the faculty of The University of North Carolina at Charlot*. 2013.

[13] K. Ho, C. O. Lee, C. Negre, Binary Edwards Curves Revisited, *Ser. Lect. Notes Comput. Sci.*, 8885, 393–408, 2014.

[14] L. Martinez, Protótipo Coprocesador para Curvas de Edward en Campos de Galois basado em MPSoC Propeller, Universidad de los LLanos, 2011.

[15] O. Contreras, Desarrollo y evaluación de funciones para aritmética de curvas de Edward en el microcontrolador MSP430, Tesis de Pregrado. Universidad de los Llanos, 2012.

[16] C. Ortiz, M. Baquero, Prototipo de criptoprocesador para aritmética de curvas de Edwards en campos de Galois, Tesis de Pregrado, Universidad de los Llanos, 2012.

[17] H. Menezes, Guide to elliptic cryptography, Springer Verlag, pp 59, 2004.

[18] H. Menezes, Guide to elliptic cryptography, Springer Verlag, pp.49-50, 2004.

[19] P. Montgomery, "Speeding the Pollard and elliptic curve methods for factorizations", *Mathematics of Computation*, 48, 243-264, 1987.