Revisión de la aritmética de curvas hiperelípticas para la implementación de un criptoprocesador a usarse en un sistema HECC

Review of the arithmetic of hyperelliptic curves for the implementation of a cryptoprocessor to be used in a HECC system

Oscar Casas García

Resumen

Este artículo, producto del proyecto de investigación Diseño de un criptoprocesador basado en curvas hiperelípticas, presenta una revisión de la literatura orientada a la teoría de curvas hiperelípticas y de cómo los puntos de estas curvas se pueden utilizar para realizar aritmética de grupo sobre ellas. Se describen las curvas hiperelípticas sobre números reales; se presenta como se conforma un grupo abeliano adecuado para realizar cómputos con curvas hiperelípticas y la operación de grupo asociada; y finalmente se describen las curvas hiperelípticas género 2 de característica 2 y la optimización de la aritmética correspondiente para este tipo de curvas. La revisión va enfocada en la búsqueda de la aritmética más eficiente para la implementación de un sistema HECC en hardware; esto es, la que presente menor cantidad de operaciones y el campo finito base más pequeño.

Palabras clave: Curvas hiperelípticas, criptografía de curvas hiperelípticas, HECC, multiplicación escalar, algoritmo de cantor, criptografía.

Abstract

This article, a result of the research project Design of a cryptoprocessor based on hyperelliptic curves, presents a literature review focused on the theory of hyperelliptic curves and how the points of these curves

• Fecha de recepción del artículo: 25-09-2009 • Fecha de aceptación: 12-07-2010.

OSCAR CASAS GARCÍA. Ingeniero Electrónico de la Universidad de San Buenaventura Cali. Magíster en Ingeniería con énfasis en Ingeniería Electrónica de la Universidad del Valle. Docente del programa de Ingeniería Electrónica e investigador del grupo de investigación LEA de la Universidad de San Buenaventura Cali, Colombia. Correo electrónico: *ocgarcia@usbcali.edu.co*

can be used for group arithmetic on them. We describe the hyperelliptic curves over real numbers, present how a suitable abelian group for computations with hy-perelliptic curves is formed, and the associated group operation. Finally we describe hyperel-liptic curves genre 2 in characteristic 2 and the optimization of the corresponding arithmetic for such curves. The review is focused on finding the most efficient arithmetic for the imple-mentation of a HECC system in hardware, i.e. the one that presents the least amount of opera-tions and the smallest base finite field.

Keywords: Hyperelliptic curves, hyperelliptic curve cryptography, HECC, scalar multiplication, algorithm, cryptography.

Introducción

El uso generalizado de las redes informáticas, así como el aumento constante del número de usuarios de estos sistemas, han motivado la necesidad de mejorar la seguridad para el almacenamiento y transmisión de la información digital. Son muchas las aplicaciones donde se debe garantizar la privacidad, la integridad o la autenticación de la información almacenada o transmitida. Tales necesidades se han podido satisfacer mediante el uso de diferentes algoritmos criptográficos, los cuales son usados en los criptosistemas de clave privada o de clave pública.

La seguridad de un criptosistema de clave pública reside entonces en problemas matemáticos que se suponen computacionalmente difíciles de resolver; es decir, problemas para cuya solución no se conocen algoritmos eficientes. Por más de un cuarto de siglo, Rivest-Shamir-Adleman (RSA) ha sido el esquema criptográfico de clave pública dominante [Menezes, 1997]. Sin embargo, su seguridad se basa en la inhabilidad de factorizar eficientemente un número compuesto grande. Si se desarrolla un algoritmo de factorización eficiente, RSA ya no será seguro.

Como resultado, la tendencia ha sido cambiar el enfoque hacia otros sistemas criptográficos que permitan una seguridad mayor por cada bit de clave y que sean implementados alrededor del problema del logaritmo discreto. Uno de tales criptosistemas es la Criptografía de Curvas Elípticas (ECC – *Elliptic Curve Cryptography*), que permite alcanzar un nivel de seguridad comparable a RSA con un tamaño de clave más pequeño. Adicionalmente, se ha reportado que las implementaciones en hardware de ECC requieren un área significativamente menor y una mayor velocidad que su contraparte RSA [Gura, 2004].

Por otro lado, la comunidad académica científica presenta una generalización de la ECC, denominada Criptografía de Curvas Hiperelípticas (HECC – *Hyperelliptic Curve Cryptography*), la cual fue propuesta teóricamente en un principio por Neil Koblitz en Crypto 1988 [Koblitz, 1989]. Las curvas hiperelípticas son curvas especiales que presentan algoritmos que permiten ejecutar la aritmética de grupo más rápidamente [Menezes, 2004], lo cual hace que la implementación sea más eficiente [Lange, 2005]; sin embargo, recientemente estas curvas están empezando a recibir atención para ser usadas en la práctica y crear una alternativa a los criptosistemas ECC [Clancy, 2003].

Una de las principales razones para usar HECC en lugar de RSA, es la longitud de la clave requerida para un nivel de seguridad equivalente. Los criptosistemas basados en HECC pueden tener longitudes de clave más corta [Duquesne, 2006], debido a que es muy difícil realizar un ataque de tiempo subexponencial contra ellos, mientras que RSA puede romperse con un buen algoritmo de factorización [Avanzi, 2006]. Blake en [Blake, 1999] presenta una función que permite conocer el tamaño de clave ECC equivalente en RSA. Estos resultados también aplican directamente a la HECC, pues la longitud de la clave es la misma que para un sistema ECC, aunque el cuerpo de base es más pequeño en un sistema HECC.

Las implementaciones en hardware de HECC comenzaron en el 2001 con Wollinger en [Wollinger, 2001]. En la Tabla 1 se muestran los tiempos de ejecución de implementaciones en hardware de criptosistemas basados en curvas hiperelípticas.

En todos los sistemas HECC la operación principal es la multiplicación escalar de algún divisor en el Jacobiano de una curva hiperelíptica; por lo tanto, la eficiencia de esta operación de multiplicación escalar es de mucha importancia en la criptografía de curvas hiperelípticas.

Planteamiento de la temática

Para ciertas curvas elípticas, Montgomery (1987) desarrolló un método denominado escalera Montgomery, que permite una multiplicación escalar más rápida que los métodos usuales. Además, este método tiene la ventaja de ser inherentemente resistente a los ataques de canal adyacente, haciéndolo muy interesante para implementaciones de ECC en sistemas embebidos donde los recursos computacionales son mínimos.

Con el trabajo de Duquesne (2008), se generaliza el método de Multiplicación Escalar Montgomery para todas las curvas hiperelípticas género 2 de característica 2; y teniendo en cuenta que las curvas hiperelípticas permiten generalizar los criptosistemas de curvas elípticas en un campo base más pequeño, haciendo las operaciones básicas de campo finito más económicas desde el punto de vista de área usada por la lógica implementada, se podría obtener un procesador de multiplicación escalar para sistemas HECC competitivo ó con mejor desempeño que un sistema ECC que ofrezca la misma seguridad de clave.

Para lograr este objetivo se deben desarrollar diferentes operaciones aritméticas a diferentes niveles de jerarquía, de manera que sean óptimas para su implementación en hardware; es decir, operaciones rápidas y usando la menor área posible. Los bloques operacionales que se deben implementar son:

– *Aritmética de campo finito*. En este nivel de jerarquía se consideran las operaciones aritméticas de elementos de un campo finito, que son la base aritmética para los niveles superiores. En este nivel se encuentran operaciones como suma, elevación al cuadrado, raíz cuadrada y multiplicación.

– *Aritmética de divisores*. Este nivel de jerarquía se refiere a las operaciones realizadas en el Jacobiano de la curva hiperelíptica entre los elementos de torsión del grupo, denominados divisores. Para el caso especial de esta implementación en hardware de la multiplicación escalar

Tabla 1
Implementaciones en hardware de multiplicación escalar en HECC

Implementación	Plataforma	Género de la curva	Campo	Multiplicación Escalar (ms)
[Wollinger, 2001]*	FPGA	4		21.4
[Clancy, 2003]*	Xilinx FPGA Virtex II	2		10
				14
				19
				26
				33
				40
[Nguyen 11, 2002]	SmartCard FameXE Coprocessor	2		30
[Elias, 2006]	Xilinx FPGA Virtex II	2		2

*Estos valores son estimados

usando la escalera Montgomery, no se trabaja directamente en el Jacobiano de la curva, sino en la superficie Kummer de la misma. Las operaciones que se deben implementar en este nivel son la suma y el doblado de divisores.

– *Multiplicación escalar*. En este nivel se implementa la operación de multiplicación escalar de un divisor en la superficie Kummer de una curva hiperelíptica de género 2 de característica 2, utilizando el método de multiplicación escalar Montgomery.

Entonces, la presente revisión se enfoca en la aritmética de curvas hiperelípticas, o de divisores, que permitiría optimizar una implementación en hardware de un procesador criptográfico encargado de realizar la multiplicación escalar en curvas hiperelípticas utilizando el método de Multiplicación Escalar Montgomery; el cual teóricamente según Duquesne (2004) Byramjee (2004) y Duquesne (2006), permitiría realizar esta operación en un tiempo menor que su equivalente para sistemas ECC y las implementaciones similares para sistemas HECC.

Recuperación bibliográfica

Curvas hiperelípticas

Las curvas hiperelípticas son una clase especial de curvas algebraicas (Frey, 2006) y una curva hiperelíptica de género g sobre el campo K está definida por la Ecuación 1.

$$C: y^2 + h(x)y = f(x) \qquad (1)$$

Donde $f(x)$ es un polinomio mónico de grado $(2g + 1)$, y $h(x)$ es un polinomio con grado máximo g. Además, $f(x)$ y $h(x)$ están definidas en $K[x]$ y no hay puntos singulares en la curva (Menezes, 2004).

El negativo de un punto $P(x, y)$ está dado por $-P = (x, -y - h(x))$. Los puntos fijados bajo esta *involución hiperelíptica* se denominan *puntos Weierstraβ*. Cabe destacar que las curvas elípticas se encuentran cubiertas bajo esta definición como curvas hiperelípticas de género $g = 1$.

En las Figuras 1, 3 y 5 se muestran ejemplos de algunas curvas hiperelípticas con los coeficientes de sus respectivos polinomios $f(x)$ y $h(x)$, siendo elementos del conjunto de números reales. Y los ejemplos de las Figuras 2, 4 y 6 presentan los coeficientes de $f(x)$ y $h(x)$ en un campo primo $GF(29)$. En todos estos ejemplos cada curva presenta género 2 y $h(x) = 0$.

En general, para aplicaciones criptográficas, los coeficientes de estos polinomios $f(x)$ y $h(x)$ son elementos de un campo finito; en particular, para diseños óptimos en hardware, los coeficientes de estos polinomios son elementos de un campo finito de característica 2; es decir, de la forma $GF(2^m)$.

Operación de grupo para curvas hiperelípticas

Para curvas hiperelípticas de género 1 (también conocidas como curvas elípticas) se pueden tomar los puntos que pertenecen a dicha curva y un punto al infinito (P_∞) como elementos para conformar un grupo abeliano (Hankerson, 2004). Sin embargo, para curvas hiperelípticas de género mayor a uno esto ya no es posible; es decir, los puntos de una curva hiperelíptica no conforman un grupo abeliano y por tanto no se puede realizar operaciones aritméticas con ellos.

Entonces, los elementos del grupo abeliano para curvas hiperelípticas se obtienen de sumas finitas de puntos que pertenecen a la curva y su operación de grupo se realiza a manera de suma de coeficientes.

Por ejemplo, si se asume que P, Q, y R son puntos sobre una curva hiperelíptica C, $[P + Q]$ y $[R + Q]$ podrían ser elementos del grupo abeliano asociado a la curva que estamos considerando, y la operación de grupo de estos elementos sería la Ecuación 2.

$$[P + Q] \oplus [R + Q] = [P + 2Q + R] \qquad (2)$$

Figura 1

Curva hiperelíptica en reales: $y^2 = x^5 + x^4 + 4x^3 + 4x^2 + 3x + 3$

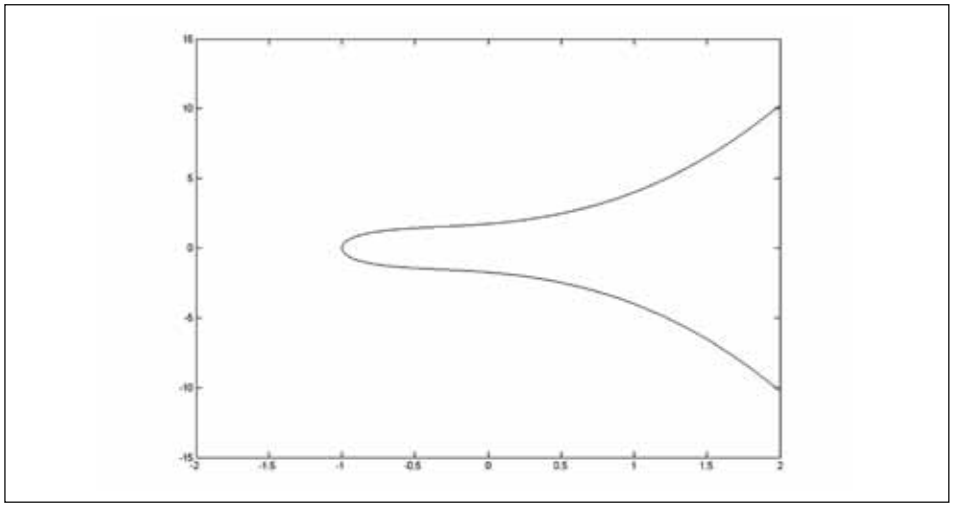


Figura 2

Curva hiperelíptica en $GF(29)$: $y^2 = x^5 + x^4 + 4x^3 + 4x^2 + 3x + 3$

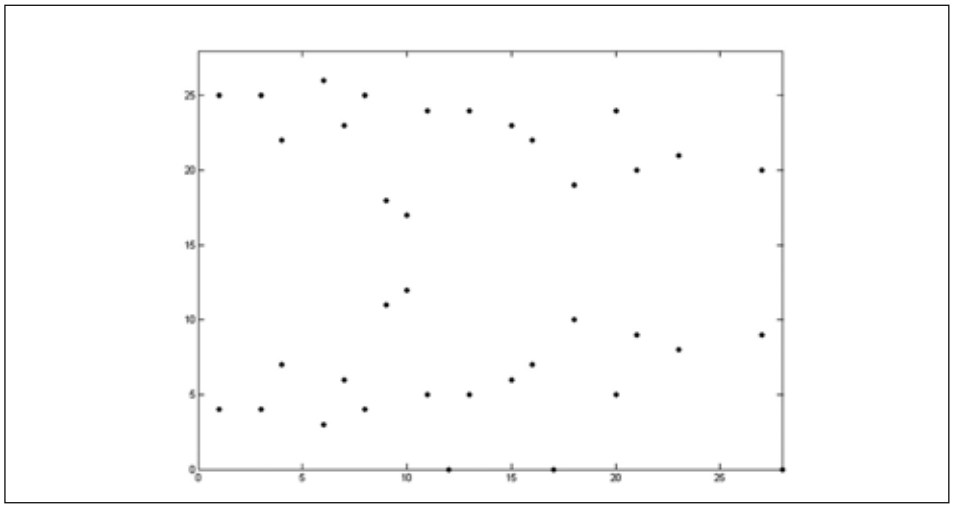


Figura 3

Curva hiperelíptica en reales: $y^2 = x^5 + x^4 - x^2 - x$

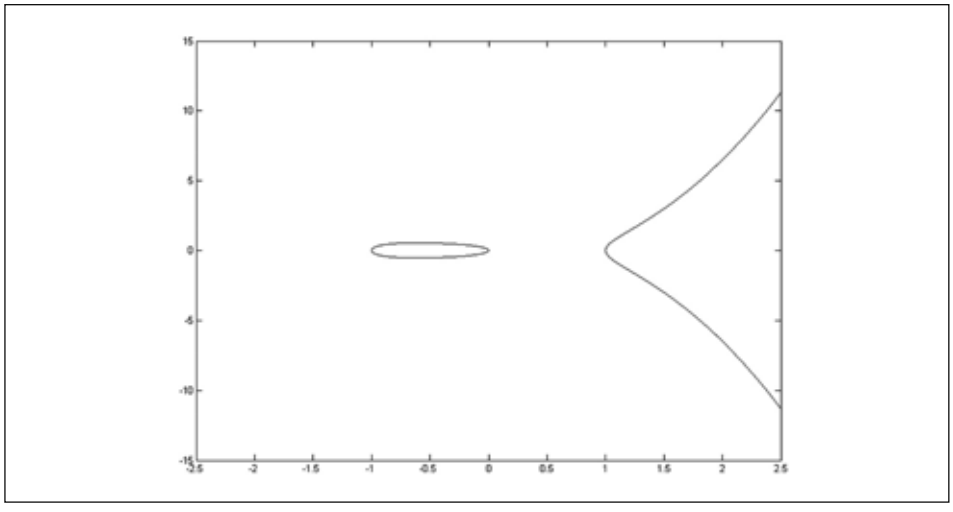


Figura 4

Curva hiperelíptica en $GF(29)$: $y^2 = x^5 + x^4 - x^2 - x$

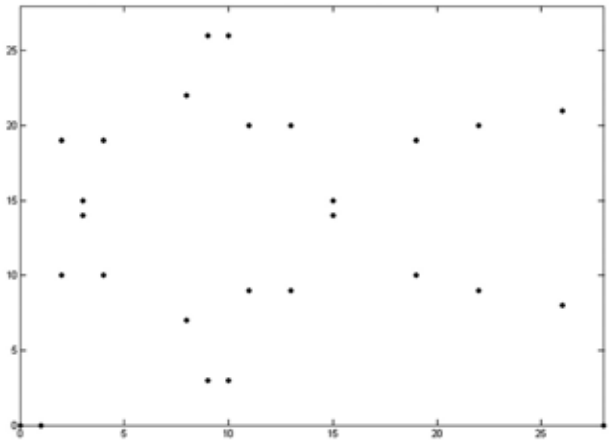


Figura 5

Curva hiperelíptica en reales: $y^2 = x^5 - 5x^3 + 4x$

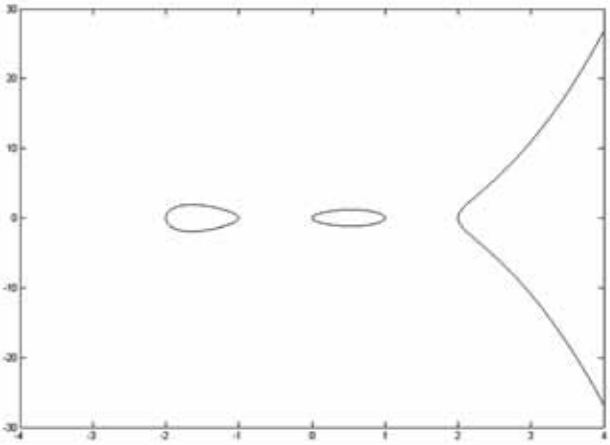
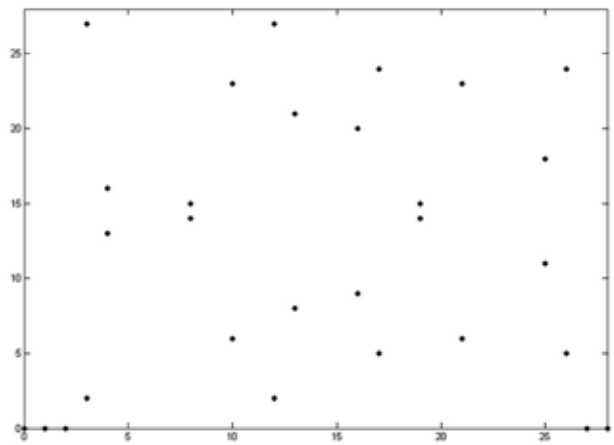


Figura 6

Curva hiperelíptica en $GF(29)$: $y^2 = x^5 - 5x^3 + 4x$



Lastimosamente, este grupo es infinito y las representaciones de elementos de grupo pueden llegar a ser muy grandes.

Para formar un grupo abeliano que se pueda utilizar, se toma el grupo cociente que resulta al realizar la división del grupo de sumas de puntos de la curva entre el subconjunto de esas sumas cuyos puntos subyacen en una función [Duquesne, 2006].

Por ejemplo, los puntos $R_1 = (x_{R1}, y_{R1})$ y $-R_1 = (x_{R1}, -y_{R1})$ mostrados en la Figura 2.7 subyacen en la función dada por $s_2(x) = x_{R1}$, y por lo tanto $R_1 + (-R_1) = 0$. De igual forma, los seis puntos P_1, P_2, Q_1, Q_2, $-R_1$ y $-R_2$, mostrados en la Figura que subyacen en la función cúbica $s_1(x)$, suman 0 en el grupo cociente que se considera, y por tanto resulta la Ecuación 3.

$$[P_1 + P_2] \oplus [Q_1 + Q_2] \oplus [-R_1 - R_2] = 0 \quad (3)$$

De esta forma, se puede observar que cada elemento puede representarse por al menos dos puntos que no presentan ni la misma coordenada x, ni la misma coordenada y inversa.

Cualquier número $n > 1$ de puntos da lugar a una función de grado $n - 1$ y se presentan otros $\max\{5, 2(n-1)\} - n$ puntos de intersección. Cuando n es mayor a 2, el inverso de esta suma (obtenido al invertir los puntos con respecto al eje x) contiene menos puntos. Al repetir este proceso se reduce el número de puntos que contiene el elemento del grupo, al menos hasta dos puntos. Entonces la operación de grupo para dos elementos del grupo cociente se realiza en dos pasos: primero se realiza la suma formal, y luego se reduce.

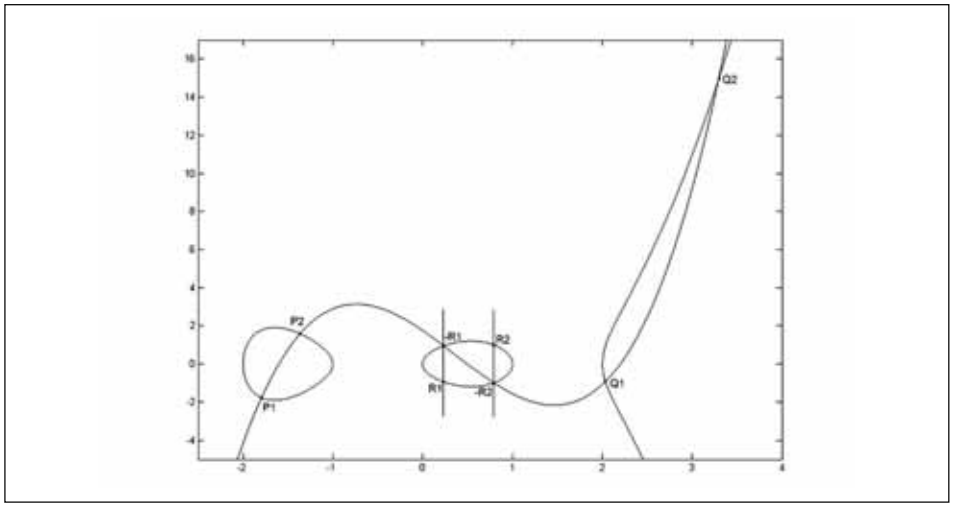
En el caso general, ambos elementos de grupo $[P_1 + P_2]$ y $[Q_1 + Q_2]$, constan de dos puntos dados, siendo los cuatro puntos todos diferentes. Una función $s(x)$ de grado 3 atraviesa todos los puntos teniendo dos puntos adicionales de intersección con C. Los nuevos dos puntos $-R_1$ y $-R_2$ se invierten con respecto al eje y y se convierten en el resultado de la operación de grupo, como se expresa en la Ecuación 4.

$$[P_1 + P_2] \oplus [Q_1 + Q_2] = [R_1 + R_2] \quad (4)$$

Para curvas hiperelípticas de género arbitrario se obtiene que cada elemento del grupo se encuentra representado como máximo por g puntos y que el paso de reducción se debe realizar en varias rondas para obtener un elemento con representación mínima de puntos.

Formalmente, el grupo que se acaba de describir se denomina *Grupo de Clase Divisor* Pic^0_C de C.

Figura 7
Operación de grupo en curvas hiperelípticas de género



A continuación se presenta con mayor detalle matemático esta operación de grupo y los entes matemáticos relacionados; sin embargo, se sugiere ver [Frey, 2006], [Menezes, 2004] y [Duquesne, 2006] para una lectura más profunda y rigurosa sobre estos conceptos.

Divisor

Como se explicó anteriormente, los puntos de una curva hiperelíptica de género mayor a uno no conforman un grupo abeliano; y por tanto, en este estado no se puede realizar aritmética con ellos. Sin embargo, sí es posible tener un grupo asociado a los puntos de una curva hiperelíptica C.

El *grupo divisor de C*, Div_C es un grupo abeliano, y un elemento $D \in \mathrm{Div}_C$ se denomina *divisor*. Un divisor es una suma formal finita de todos los puntos de la curva y un punto al infinito, como lo describe la Ecuación 5, siendo n_i un entero y casi siempre 0.

$$D = \Sigma\, n_i P_i \qquad (5)$$

El entero n_i asociado al punto P_i se denomina el orden de D en P_i, y se denota: $\mathrm{ord}_{pi}\, D = n_i$. El *grado* de cualquier divisor es la sumatoria de sus coeficientes n_i, donde $\deg(D) = \Sigma\, n_i$. Se resalta que todo divisor resultante de las intersecciones de una función racional $s(x)$ y la curva hiperelíptica es de grado cero y se le llama *divisor principal.*

El conjunto de todos los divisores de grado cero se denota $\mathrm{Div}^0{}_C$ y el conjunto de divisores principales Princ_C. Princ_C es un subconjunto de $\mathrm{Div}^0{}_C$ y este a su vez de es un subconjunto de Div_C. El conjunto de todos los divisores Div_C conforma un grupo aditivo bajo la regla de suma, definida en la Ecuación 6.

$$D_1 \oplus D_2 = \Sigma\, n_i P_i + \Sigma\, m_i P_i = \Sigma\,(n_i + m_i)\, P_i \qquad (6)$$

Grupo de clase divisor

El *Grupo de Clase Divisor* $\mathrm{Pic}^0{}_C$ de C es el grupo que resulta al dividir $\mathrm{Div}^0{}_C$ entre el grupo de divisores principales Princ_C. También se le conoce como el G*rupo Picard de C*. Entonces, dos divisores D_1 y D_2 se encuentran en la misma clase si existe una función $s(x)$ tal que $\mathrm{div}(x) = D_1 \ominus D_2$.

En contraste con el grupo de divisores Div_C, el grupo de clase divisor $\mathrm{Pic}^0{}_C$ tiene muchos elementos de torsión, y si el campo sobre el cual está definido es finito, entonces también es finito.

Jacobiano

El *Jacobiano* o *Variedad Jacobiana* de C, escrito como J_C, es un isomorfismo de $\mathrm{Pic}^0{}_C$ que le permite obtener la estructura de un grupo algebraico, proyectivo, irreducible y de variedad abeliana.

Esto implica que J_C es un grupo en el cual su operación de grupo $\oplus$ está dada mediante la evaluación de funciones racionales (si se toman coordenadas afines) o polinomios (si se toman coordenadas proyectivas) con coeficientes dados por parejas $[u, v]$. Como resultado se pueden introducir coordenadas para elementos en $\mathrm{Pic}^0{}_C$ y llevar a cabo cómputos utilizando fórmulas algebraicas.

Representación Mumford

La representación Mumford hace evidente el isomorfismo entre J_C y $\mathrm{Pic}^0{}_C$. Esta representación se basa en que cada divisor no trivial se puede representar a través de un par único de polinomios $u(x)$ y $v(x)$ donde:

- $u(x)$ es mónico
- $\deg(v) = \deg(u) \leq g$
- $u \mid v^2 + v * h - f$

Sea $D = (\Sigma_{i=1}^{g} P_i) - g P_\infty$, siendo $P_i = (x_i, y_i)$, el divisor D se encuentra representado por las Ecuaciones 7 y 8.

$$u(x) = \prod_{i=1}^{g} (x - x_i) \qquad (7)$$

$$v(x) = \sum_{i=g}^{t} \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}\, y_i \qquad (8)$$

Entonces, el divisor D representado por $u(x)$ y $v(x)$ se denota $[u(x), v(x)]$, y para unificar notación, el elemento neutro se escribe como [1,0].

Algoritmo de Cantor

Se puede transferir la operación de grupo que se describió anteriormente como una secuencia de composición y reducción, a un algoritmo que opera sobre los polinomios representativos de Mumford, y utiliza solamente aritmética polinomial sobre el campo de definición. Este algoritmo fue descrito por Cantor (1987) para campos de característica impar y por Koblitz en [Koblitz, 1989] para campos arbitrarios.

El Algoritmo de Cantor, presentado en el Algoritmo 1, permite obtener la operación de grupo entre dos divisores en representación Mumford: $D = D_1 \oplus D_2$. El Algoritmo de Cantor es completamente general y sirve para cualquier campo y género. Se puede verificar que las fórmulas para la operación de grupo de curvas elípticas (curvas hiperelípticas de género 1) se pueden obtener como un caso especial del Algoritmo de Cantor realizando todos los pasos explícitos para $g = 1$.

Curvas hiperelípticas género 2 de característica 2

Las curvas hiperelípticas de característica 2 se encuentran definidas sobre un campo finito $GF(2^m)$, donde m es el grado del polinomio irreducible de reducción modular $r(X)$. También, para propósitos criptográficos, se requiere que m sea primo. Tomando en cuenta la definición de las curvas hiperelípticas dada por la Ecuación 1*;* entonces, los polinomios $f(x)$ y $h(x)$ son elementos del anillo $GF(2^m)[X]$, es decir, los coeficientes de estos polinomios pertenecen al campo $GF(2^m)$.

En las Figuras 8, 9 y 10 se muestran ejemplos de algunas curvas hiperelípticas género 2 de característica 2. En estos ejemplos cada curva presenta $h(x) = 0$ y se encuentra en el campo $GF(2^5)$ con polinomio de reducción modular $r(X) = X^5 + X^2 + 1$.

La cardinalidad

La cardinalidad de una curva hiperelíptica determina el número de elementos de torsión del grupo; es decir, el número de elementos del Jacobiano con los que se realiza aritmética sobre la curva. La cardinalidad $|\mathrm{Pic}^0_C|$ se encuentra dada por el teorema Hasse – Weil [Duquesne, 2006] descrito por la Ecuación 9, que indica los límites del número de elementos de torsión dependiendo sólo del campo finito $GF(2^m)$ y del género de la curva .

$$(\sqrt{2^m} - 1)^{2g} \leq |\mathrm{Pic}^0_C| \leq (\sqrt{2^m} + 1)^{2g} \quad (9)$$

Algoritmo 1. Algoritmo de Cantor

Entrada: Dos divisores $D_1 = [u_1, v_1]$ y $D_2 = [u_2, v_2]$ en la curva $C: y^2 + h(x)y = f(x)$:

Salida: El divisor reducido único $D = [u, v]$ tal que $D = D_1 \oplus D_2$

Paso 1. $d_1 \leftarrow \mathrm{gdc}(u_1, u_2)$

Paso 2. $d \leftarrow \mathrm{gdc}(d_1, v_2 + v_2 + h)$

Paso 3. $s_1 \leftarrow c_1 e_1, s_2 \leftarrow c_1 e_2, s_3 \leftarrow c_2$

Paso 4. $u \leftarrow \frac{u_1 u_2}{d^2}$, $v \leftarrow \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3 (v_1 v_2 + f)}{d} \bmod u$

Paso 5. Repetir

Paso 6. $u' \leftarrow \frac{f - vh - v^2}{u}$, $v \leftarrow (-h - v)$

Paso 7. $u' \leftarrow u, v' \leftarrow v$

Paso 8. Hasta deg $u \leq g$

Paso 9. Hacer u mónico

Paso 10. Retornar $[u, v]$

Figura 8

Curva hiperelíptica en $GF(2^5)$: $y^2 = x^5 + x^4 + 4x^3 + 4x^2 + 3x + 3$

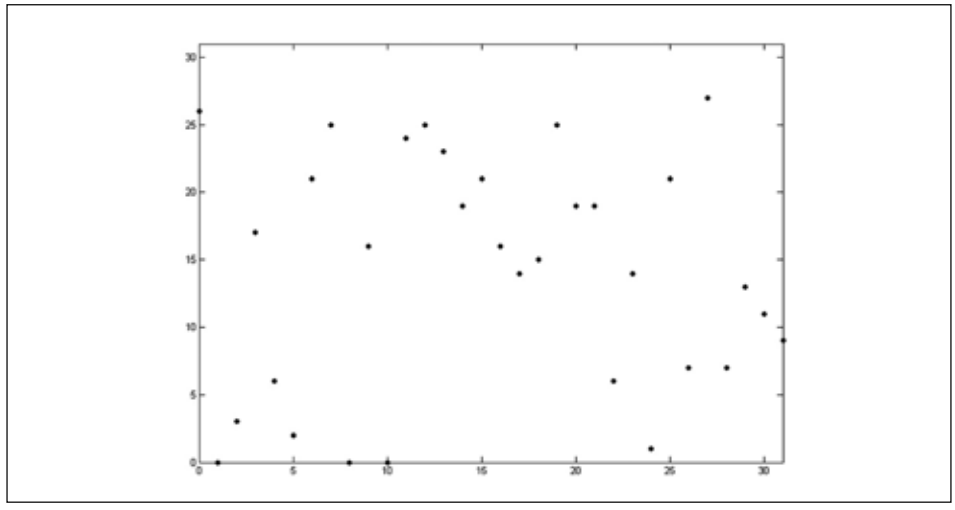


Figura 9

Curva hiperelíptica en $GF(2^5)$: $y^2 = x^5 + x^4 - x^2 - x$

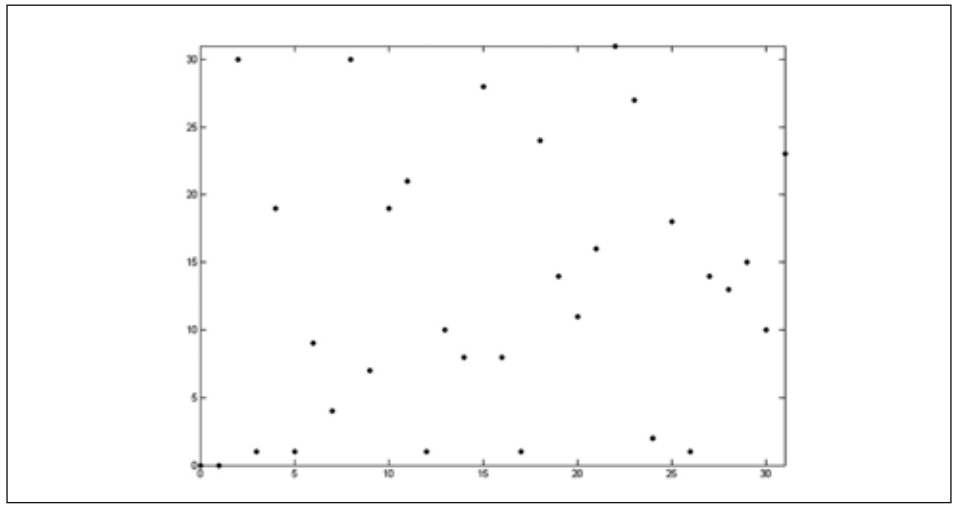
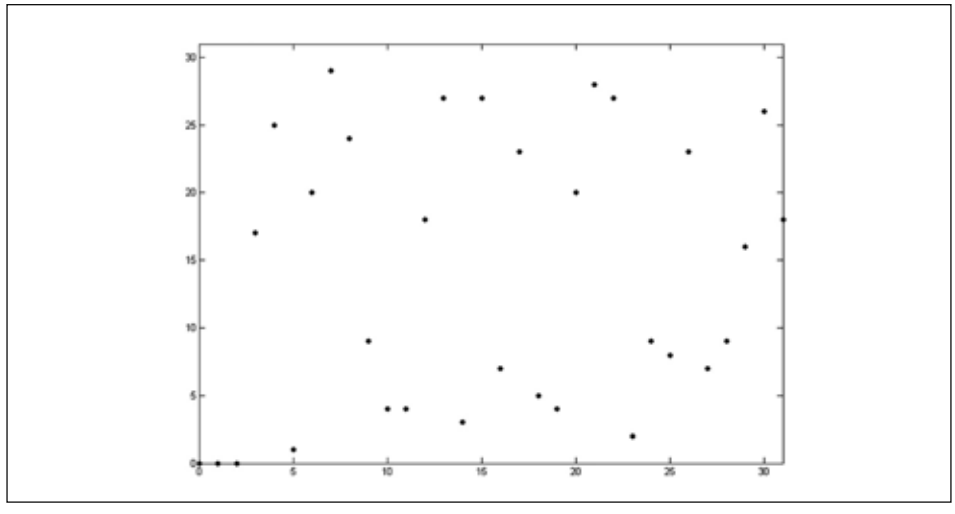


Figura 10

Curva hiperelíptica en $GF(2^5)$: $y^2 = x^5 - 5x^3 + 4x$



De manera que si queremos una cardinalidad de aproximadamente 2^{160}; es decir, 2^{160} diferentes elementos para realizar criptografía, con curvas de género 1 se necesitará un campo $GF(2^{160})$, lo que representa datos de 160 bits, mientras que con curvas de género 2 sólo se necesitará un campo $GF(2^{80})$, que representaría datos de tan sólo 80 bits. En la Tabla 2 se muestran los campos finitos para una cardinalidad de aproximadamente 2^{160} y diferentes géneros de curva.

Como se puede observar en la Tabla 2, a medida que se utiliza una curva con género mayor, el campo es menor y por tanto los bits necesarios para datos son menores. Debido a esta propiedad los sistemas criptográficos de curvas hiperelípticas presentan ventajas de implementación en ambientes de pocos recursos.

Tabla 2

Campos finitos para una cardinalidad de ~2^{160}

Género de la curva	Campo Finito de Característica
$g = 1$	$GF(2^{160})$
$g = 2$	$GF(2^{80})$
$g = 3$	$GF(2^{53})$
$g = 4$	$GF(2^{40})$

A medida que aumenta el género de la curva, el tamaño del campo base disminuye; sin embargo, no es aconsejable utilizar curvas con género mayor o igual a 5, debido a que ya se han propuesto algoritmos que resuelven el Problema del Logaritmo Discreto para curvas de género mayor a 5 (Gaudry, 2000), (Thériault, 2003) y (Hartley, 2000).

Aritmética para curvas hiperelípticas género 2 de característica 2

El algoritmo descrito por Cantor (Algoritmo 1) para sumar divisores en Pic^0_C, que sirve para cualquier género, es computacionalmente muy lento, principalmente porque utiliza algoritmos gcd y aritmética polinomial, y se necesitaría mucha memoria para su implementación en ambientes restringidos de hardware como *smart cards*.

Hartley (2000) obtiene fórmulas explícitas para llevar a cabo aritmética rápida de curvas hiperelípticas de género 2. Con el propósito de mejorar el desempeño del algoritmo de Cantor, [Lange, 2002 (1); Lange, 2002 (2); Lange, 2002 (3) y Lange, 2002 (4)] se consideran la aritmética de curvas hiperelípticas de género 2 utilizando coordenadas afines, proyectivas y coordenadas con peso, obteniéndose con ello aritmética libre de inversiones en estas últimas coordenadas.

Pelzl (2003 y 2004) considera los criptosistemas para HECC de género 2 con parámetros fijos para ambientes embebidos: como PDA y dispositivos de comunicación móvil y se obtiene una mejora en la aritmética para algunas curvas no supersingulares definidas sobre campos de característica 2.

Byramjee (2004) clasifica las curvas hiperelípticas de género 2 en tres diferentes tipos, optimiza su aritmética acorde con dicha clasificación, estudia la aritmética y seguridad de su Jacobiano, se concluye que las curvas descritas por la Ecuación 10 son las mejores para uso criptográfico y se recomienda este tipo de curvas para futuros estándares.

$$y^2 + xy = x^5 + f_3x^3 + x^2 + f_0 \qquad (10)$$

En la Tabla 3 se presentan de manera resumida el número de operaciones de campo de multiplicación M e inversión multiplicativa I en $GF(2^m)$, las cuales son necesarias para llevar a cabo una suma y un doblado de divisores, con fórmulas explícitas del algoritmo de Cantor optimizadas para curvas género 2 de característica 2 en diferentes coordenadas. Se supone que las sumas de campo S son despreciables en comparación con las multiplicaciones e inversiones, y por tanto no se consideran.

Lange (2005) presenta fórmulas explícitas para operaciones de grupo de curvas de género 2 completamente generales, pero para obtener un mínimo de operaciones tratan campos de característica par e impar por separado, utilizan tres sistemas de coordenadas apropiados para diferentes ambientes como *software* o *smart cards*.

Tabla 3

Operaciones de campo para curvas género 2 en $GF(2^m)$

Coordenadas	Curvas Generales =2 en $GF(2^m)$	Curvas Tipo I		Curvas Tipo II
AFINES				
Suma	25M+I	25M+I		24M+I
Doblado	27M+I	26M+I		18M+I
PROYECTIVAS		(Ia)	(Ib)	
Suma	45M	45M	44M	42M
Doblado	45M	44M	41M	31M
PROYECTIVAS MODIFICADAS				
Suma	45M	45M	44M	42M
Doblado	43M	42M	40M	31M
PROYECTIVAS CON PESO				
Suma	42M	42M	41M	40M
Doblado	46M	45M	42M	27M

Gaudry (2007) se propone utilizar fórmulas provenientes de Funciones Theta para la aritmética en el Jacobiano de curvas de género 2 y deriva fórmulas rápidas para la multiplicación escalar en la superficie Kummer asociada a la curva, utilizando la Escalera Montgomery.

Finalmente, Duquesne (2004 y 2008), utilizando la superficie Kummer, generaliza la escalera Montgomery para multiplicación escalar en el Jacobiano de curvas de género 2. Obtiene un algoritmo que es competitivo comparado con los métodos usuales de multiplicación escalar y presenta propiedades adicionales como la resistencia a ataques de canal adyacente, además de proveer en muchos casos una aceleración en el cálculo de la multiplicación escalar. Este nuevo algoritmo presenta aplicaciones muy importantes en criptografía de curvas hiperelípticas y particularmente en sistemas criptográficos basados en curvas hiperelípticas para sistemas embebidos.

Conclusiones

El análisis de la literatura y los desarrollos en la optimización de la aritmética de curvas hiperelípticas permiten llegar a las siguientes conclusiones:

- El desarrollo de ecuaciones explícitas para la operación de grupo de curvas hiperelípticas (algoritmo de Cantor) es un campo de investigación abierto.
- Cuanto mayor es el género de la curva hiperelíptica, el campo finito base a utilizar para la implementación de la aritmética de campo es menor. Esto es beneficioso para implementaciones en hardware, puesto que los bloques de operaciones de campo finito ocuparían menos espacio. Sin embargo, la aritmética explícita requiere una mayor cantidad de operaciones.
- Se deben utilizar curvas hiperelípticas de género menor a 5 en sistemas HECC, pues ya se han propuesto algoritmos que resuelven el Problema del Logaritmo Discreto para curvas de género mayor a 5, lo que compromete la seguridad de los sistemas.
- Para la implementación de sistemas HECC en hardware se recomienda la utilización de curvas hiperelípticas género 2 de característica 2 con ecuación característica:

$$y^2 + xy = x^5 + f_3x^3 + x^2 + f_0$$

- Para realizar la operación de multiplicación escalar utilizando curvas hiperelípticas género 2 de característica 2, se recomienda utilizar coordenadas con peso para la representación de los divisores, puesto que se se obtienen ecuaciones explicitas con la mínima cantidad de operaciones de campo finito y es una aritmética libre de inversiones: 40 multiplicaciones de campo finito para una suma y 27 multiplicaciones de campo finito para un doblado.

- Es posible, utilizando la superficie Kummer, generalizar la escalera Montgomery para multiplicación escalar en el Jacobiano de curvas de género 2 y obtener un algoritmo que es competitivo comparado con los métodos usuales de multiplicación escalar y que presenta propiedades adicionales como la resistencia a ataques de canal adyacente, además de proveer en muchos casos una aceleración en el cálculo de la multiplicación escalar. Este método se recomienda explorarlo para implementaciones en hardware.

Bibliografía

- ABDALLA, M.; BELLARE, M., & ROGAWAY, P. (1999). *DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem.* IEEE P1363a Submission.

- ANSI X9.62-1999. (1999). *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).* ANSI.

- ANSI X9.63-2001. (2001). *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography.* ANSI.

- AVANZI, R. (2006). *Generic Algorithms for Computing Discrete Logarithms*. En: R. Avanzi, C. Doche, T. Lange, K. Nguyen, & F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography* (págs. 475-209). Boca Raton: Chapman & Hall / CRC.

- AVANZI, R. M. & LANGE, T. (2006). *Introduction to Public-Key Cryptography*. En: H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, y otros, Handbook of Elliptic And Hiperelliptic Curve Cryprography (págs. 1-15). Boca Raton: Chapman &Hall/CRC.

- BLAKE, I., SEROUSSI, G., & SMART, N. (1999). *Elliptic Curves in Cryptography.* Cambridge: University Press.

- BYRAMJEE, B. & DUQUESNE, S. (2004). *Classification of genus 2 curves over F(2^n) and optimization of their arithmetic*. *Cryptology ePrint Archive* (107).

- CANTOR, D. G. (1987). *Computing in the Jacobian of a Hyperelliptic Curve. Mathematics of Computation, 48* (177), 95-101.

- CERTICOM CORPORATION (2000). *The Elliptic Curve Cryptosystem: Remarks on the Security of the Elliptic Curve Cryptosystem*. A Certicom Whitepaper.

- CLANCY, T. C. (2003). *Analisys of FPGA-Based Hyperelliptic Curve Cryptosystems.* Master Thesis, University of Illinois.

- DIFFIE, W. & HELLMAN, M. E. (1976). *New directions in cryptography*. IEEE Transactions on Information Theory, 644- 654.

- DIFFIE, W.; VAN OORSCHOT, P. C., & WIENER, M. J. (1992). *Authentication and Authenticated Key Exchanges*. Designs, Codes and Cryptography, 2 (2), 107-125.

- DOCHE, C. (2006). *Finite Field Arithmetic*. En H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, y otros, *Handbook Of Elliptic And Hiperelliptic Curve Cryprography* (págs. 201-237). Boca Raton: Chapman&Hall/CRC.

- DOCHE, C., & LANGE, T. (2006). *Arithmetic of Elliptic Curves*. En R. Avanzi, C. Doche, T. Lange, K. Nguyen, & F. Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography* (págs. 267-302). Boca Raton: Chapman & Hall / CRC.

- DUQUESNE, S. (2004). *Montgomery Scalar Multiplication for Genus 2 Curves*. Lecture Notes in Computer Science , 153–168.

- _____. (2007). *Traces of the Group Law on the Kummer Surface of a Curve of Genus 2 in Characteristic 2*. Preprint .

– _____. (2008). *Montgomery Ladder for All Genus 2 Curves in Characteristic 2*. WAIFI 2008, 174-188.

– DUQUESNE, S., & LANGE, T. (2006). *Arithmetic of Hyperelliptic Curves*. En H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, y otros, *Handbook of Elliptic and Hiperelliptic Curve Cryprography* (págs. 303-353). Boca Raton: Chapman &Hall/CRC.

– DUQUESNE, S., & LANGE, T. (2006). *Pairing-Based Criptography*. En R. Avanzi, C. Doche, T. Lange, K. Nguyen, & F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography* (págs. 573-590). Boca Raton: Chapman & Hall / CRC.

– ELGAMAL, T. (1985). *A public key cryptosystem and a signature scheme based on discrete logarithms. Proceedings of CRYPTO 84 on Advances in cryptology*, 10 - 18.

– ELIAS, G.; MIRI, A. & YEAP, T.H. (2006). *High-Performance, FPGA-Based Hyperelliptic Curve Cryptosystem*. Ottawa: University of Ottawa.

– ERNST, M.; KLUPSCH, S.; HAUCK, O. & HUSS, S. A. (2001). *Rapid Prototyping for Hardware Accelerated Elliptic Curve Public-Key Cryptosystems. Proc. 12th IEEE Workshop on Rapid System Prototyping (RSP01)*.

– FIPS 186-2. (2000). *Digital Signature Standard (DSS)*.

– FLYNN, E. V. & SMART, N. P. (1997). *Canonical Heights on the Jacobians of Curves of Genus 2 and the Infinite Descent. Acta Arithmetica* (LXXIX), 333-352.

– FREY, G., & LANGE, T. (2006). *Background on Curves and Jacobians*. En: H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, y otros, *Handbook Of Elliptic And Hiperelliptic Curve Cryprography* (págs. 45-85). Boca Raton: Chapman &Hall/CRC.

– FREY, G., & RÜCK, H. G. (1994). *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Mathematics of Computation*, *62* (206), 865 - 874.

– GAUDRY, P. (2000). *An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. Advances in Cryptology - EUROCRYPT 2000, 1807*, 19-34.

– _____. (2007). *Fast genus 2 arithmetic based on Theta functions*. *Journal of Mathematical Cryptology , 1* (3), 243-266.

– GOLDWASSER, S.; MICALI, S. & RIVEST, R. (1988). *A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing*, 17 (2), 281-308.

– GURA, N.; PATEL, A.; WANDER, A.; ERBELE, H. & SHANTZ S. C. (2004). *Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. Cryptographic Hardware and Embedded Systems - CHES 2004*. 925-943.

– HANKERSON, D.; MENEZES, A. & VANSTONE, S. (2004). *Elliptic Curve Arihtmetic*. En: D. Hankerson, A. Menezes, & S. Vanstone. *Guide to Elliptic Curve Criptography* (págs. 75-152). New York: Springer.

– _____. (2004). *Cryptographic Protocols*. En *D. Hankerson, A. Menezes, & S. Vanstone. Guide to Elliptic Curve Cryptography* (págs. 153-204). New York: Springer.

– IEEE 1363-2000. (2000). *Standard Specifications for Public Key Cryptography*.

– IEEE 1363a-2004. (2004). *Standard Specifications for Public Key Cryptography - Amendment 1: Additional Techniques*.

– ISO/IEC 15946-2. (2002). *Cryptographic Techniques Based on Elliptic Curves* - Part 2: Digital signatures.

– ISO/IEC 15946-3. (2002). *Cryptographic Techniques Based on Elliptic Curves* - Part 3: Key establishment.

– KOBLITZ, N. (1989). *Hyperelliptic Cryptosystems. Journal of Cryptology*, 1 (3), 139-150.

– LANGE, T. (2002). *Efficient Arithmetic on Hyperelliptic Curves*. *Cryptology ePrint Archive* (107).

– _____. (2002). *Efficient Arithmetic on genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae*. *Cryptology ePrint Archive* (121).

– _____. (2002). Inversion-*Free Arithmetic on Genus 2 Hyperelliptic Curves*. *Cryptology ePrint Archive* (147).

– _____. (2002). *Weighted Coordinates on Genus 2 Hyperelliptic Curves*. *Cryptology ePrint Archive* (153).

- _____. (2005). *Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. Applicable Algebra in Engineering, Communication and Computing*, 15 (5), 295-328.
- LÓPEZ, J. & DAHAB, R. (1999). *Fast Multiplication on Elliptic Curves over GF(2^m) without Precomputation. Lecture Notes In Computer Science* (1717), 316 - 327.
- MENEZES, A.; OORSCHOT, P.V. & VANSTONE, S. (1997). *Handbook of Applied Cryptography*. Boca Raton: CRC Press.
- MENEZES, A. J.; WU, Y.-H. & ZUCCHERATO, R. J. (2004). *An Elementary Introduction to Hyperelliptic Curves*. En: *N. Koblitz, Algebraic Aspects of Cryptography* (págs. 155-178). Springer.
- MONTGOMERY, P. L. (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization. Mathematics of Computation*, 48 (177), 243-264.
- NAZAR, S.; RODRÍGUEZ, F. & PÉREZ, A. (2004*). A parallel architecture for fast computation of Elliptic Curve Scalar Multiplication over GF(2^191). Computer Science section, Electrical Engineering Departament*. México: Cinvestav.
- NGUYEN, K. (2002). *Curve Based Cryptography - The State of the Art in Smart Card Enviroments. Cryptology Competence Center, Business Unit Identification. Philips Semiconductors GmbH.*
- NGUYEN, K.; & WEIGL, A. (2006). *Fast Arithmetic in Hardware*. En: H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, y otros. *Handbook Of Elliptic And Hiperelliptic Curve Cryprography* (págs. 617-646). Boca Raton: Chapman&Hall/CRC.
- ORLANDO, G. & PAAR, C. (2000). *A high performance reconfigurable elliptic curve for GF(2m), Workshop on Cryptographic Hardware and Embedded Systems CHES 2000. Springer-Verlag, Lecture Notes in Computer Science.*
- PELZL, J.; WOLLINGER, T. & PAAR, C. (2003). *High Performance Arithmetic for Hyperelliptic Curve Cryptosystems of Genus Two. Cryptology ePrint Archive* (212).
- PELZL, J.; WOLLINGER, T. & PAAR, C. (2004). *High Performance Arithmetic for Special Hyperelliptic Curve Cryptosystems. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, 513.
- THÉRIAULT, N. (2003). *Index calculus attack for hyperelliptic curves of small genus. Advances in Cryptology* - ASIACRYPT 2003 (2894), 75-92.
- TRUJILLO V.; VELASCO J. & LÓPEZ J. (2005). *Design of an Elliptic curve Cryptoprocessor ove*r GF(2^163), IX Workshop Iberchip, Brasil.
- WOLLINGER, T. (2001). *Computer Architectures for Cryptosystems Based on Hyperelliptic Curves*. Worchester, New York: Master Thesis, Worchester Polytechnic Institute.