Familias de curvas elípticas adecuadas para Criptografía Basada en la Identidad

Josep M. Miret
Departament de Matemàtica
Universitat de Lleida
Email: miret@matematica.udl.cat

Daniel Sadornil
Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria
Email: sadornild@unican.es

Juan G. Tena
Departamento de Algebra, Análisis matemático y Geometría y Topología
Universidad de Valladolid
Email: tena@agt.uva.es

***Resumen*—La Criptografía Basada en la Identidad hace uso de curvas elípticas que satisfacen ciertas condiciones (*pairing-friendly curves*), en particular, el grado de inmersión de dichas curvas debe ser pequeño. En este trabajo se obtienen familias explícitas de curvas elípticas idóneas para este escenario. Dicha criptografía está basada en el cálculo de emparejamientos sobre curvas, cálculo factible gracias al algoritmo de Miller. Proponemos una versión más eficiente que la clásica de este algoritmo usando la representación de un número en forma no adyacente (*NAF*).**

***Palabras clave*—Algoritmo de Miller (*Miller's Algorithm*); Criptografía Basada en la Identidad (*Identity Based Encryption*); Curvas elípticas (*Elliptic curves*); emparejamientos (*pairings*); Forma No Adyacente (*NAF*); grado de inmersión (*embedding degree*).**

I. Introducción

Para evitar los problemas de la autentificación de las claves públicas (certificados y autoridades de certificación) que planteaba la Criptografía de Clave Pública clásica, Shamir en 1984 [12] propuso un nuevo paradigma: la Criptografía Basada en la Identidad, en la cual la clave pública de un usuario es su propio nombre o cualquier otro atributo ligado al mismo. Una de las formas en que es posible realizar la idea de Shamir es utilizando emparejamientos (pairings) sobre curvas elípticas ([5]) (otras aproximaciones basadas en el problema de la residuosidad cuadrática han sido desarrolladas por Cooks [1] o por Boneh [4]).

En lo que sigue, E representará una curva elíptica definida sobre un cuerpo finito con q elementos $\mathbb{F}_q$, $q = p^m$, p primo, $p \geq 5$, con ecuación en la forma canónica de Weierstrass $y^2 = x^3 + Ax + B$; $A, B \in \mathbb{F}_q$. Recordemos ([9]) que el conjunto $E(\mathbb{F}_q)$ de puntos de esta curva sobre $\mathbb{F}_q$ tiene cardinal $N = q + 1 - t$, con $|t| \leq 2\sqrt{q}$ y $E(\mathbb{F}_q)$ admite una estructura de grupo abeliano.

Los emparejamientos, Weil, Tate, etc, ([3]) sobre la curva E son aplicaciones bilineales con valores en un cuerpo extensión $\mathbb{F}_{q^k}$. El número k grado de inmersión, viene dado por la siguiente,

Definición 1: Sea ℓ un divisor de $N = \sharp E(\mathbb{F}_q)$ (habitualmente ℓ primo). Se denomina grado de inmersión de $E/\mathbb{F}_q$ respecto de ℓ al mínimo entero positivo k verificando las condiciones equivalentes:

i) $\ell \mid (q^k - 1)$.
ii) $\mathbb{F}_{q^k}^*$ contiene un subgrupo cíclico de orden ℓ.

Si ℓ es el mayor divisor primo de N, k se denomina simplemente grado de inmersión de $E/\mathbb{F}_q$.

Por razones computacionales, la Criptografía Basada en la Identidad requiere curvas con grado de inmersión pequeño. En particular, las denominadas curvas supersingulares (aquellas para las que $p|t$) son idóneas para este propósito ya que estas curvas tienen siempre $k \leq 6$ ([9]). Por contra, las curvas elípticas ordinarias (aquellas no supersingulares) con grado de inmersión pequeño son una minoría en el conjunto de todas las curvas posibles y su caracterización es complicada (ver [7]). Un cálculo efectivo de los emparejamientos se puede realizar usando el algoritmo de Miller ([10]).

En el presente artículo consideramos la familia de curvas elípticas con ecuación de Weierstrass $y^2 = x^3 + Ax$ cuyas propiedades han sido estudiadas ampliamente, en particular la caracterización de sus clases de isomorfía ([11]). Esta familia proporciona ejemplos de curvas tanto supersingulares como ordinarias. En la sección II se estudiará el grado de inmersión de curvas en esta familia, en el caso supersingular, su grado de inmersión puede ser obtenido fácilmente y para el caso ordinario proponemos un método para determinar las curvas con un grado de inmersión pequeño y prefijado.

En la sección III presentamos una variante del algoritmo de Miller basada en la expresión en forma no adyacente de un número natural. En la sección IV se presentan ejemplos numéricos y tiempos de ejecución de la implementación realizada del algoritmo de Miller y las variantes propuestas.

Finalmente, en la sección V se detallan las conclusiones obtenidas en la presente comunicación.

II. Curvas con grado de inmersión pequeño

En [11] se caracterizan todas las clases de isomorfía de curvas elípticas con ecuación de Weierstrass del tipo $y^2 = x^3 + Ax$.

Proposición 2: El número de clases de isomorfía de curvas elípticas de la familia mencionada sobre $\mathbb{F}_q$, $q = p^m$ viene dado por:

ISBN: 978-84-9717-323-0

i) Si $q \equiv 1 \bmod 4$ entonces existen cuatro clases con representantes

$$E_i : y^2 = x^3 + \omega^i x, \quad 0 \le i \le 3,$$

donde ω es un generador de $\mathbb{F}_q^*$. Para $p \equiv 3 \bmod 4$, (por tanto m par), éstas son supersingulares. E_1, E_3 tienen cardinal $q+1$ mientras que E_0 tiene cardinal $q+1\pm 2\sqrt{q}$ y el cardinal de E_2 es $q+1\mp 2\sqrt{q}$ (el signo corresponde a $m \equiv 2, 0 \bmod 4$).

Si $p \equiv 1 \bmod 4$ las cuatro curvas son ordinarias.

i) Si $q \equiv 3 \bmod 4$ ($p \equiv 3 \bmod 4$ y m impar) entonces existen dos clases con representantes

$$E'_1 : y^2 = x^3 + x, \qquad E'_{-1} : y^2 = x^3 - x.$$

Ambas curvas son supersingulares con cardinal $q+1$.

En el caso supersingular, el grado de inmersión k de las curvas anteriores se deduce fácilmente a partir de su cardinal (ver [9]). Explícitamente, dicho grado se muestra en la tabla I.

Tabla I
GRADO DE INMERSIÓN DE CURVAS SUPERSINGULARES

Curva	k
E_0, E_2	1
E_1, E_3	2
E'_1, E'_{-1}	2

En lo que sigue consideraremos únicamente las curvas ordinarias, es decir las curvas E_i sobre $\mathbb{F}_q$, $q \equiv 1 \bmod 4$. Para caracterizar su grado de inmersión respecto de ℓ será útil el resultado siguiente ([6]).

Lema 3: Una curva elíptica E tiene grado de inmersión k con respecto de ℓ si y sólo si $t \equiv 1 + \zeta_k \bmod \ell$, con ζ_k una raíz de orden k de la unidad módulo ℓ.

Por ejemplo, si $k = 1$ se tiene que $t \equiv 2 \bmod \ell$ y si $k = 2$ se tiene que $t \equiv 0 \bmod \ell$. Por otra parte, ℓ debe dividir tanto a $q+1-t$ como a $q^k - 1$. Fijado el grado de inmersión, para cada primo ℓ, se pueden obtener condiciones para entero q (primo o potencia de un primo).

En particular para $k = 1$ y 2 hemos demostrado que:

Teorema 4: Una de las cuatro curvas E_i tiene grado de inmersión 1 con respecto de ℓ si y sólo si

$$q = (x^2 + y^2)\ell^2 + 2x\ell + 1, \quad x, y \in \mathbb{Z},\ x \equiv y \bmod 2.$$

Teorema 5: Una de las cuatro curvas E_i tiene grado de inmersión 2 con respecto de ℓ si y sólo si

$$q = x^2\ell^2 + y\ell - 1,\ x \equiv y \bmod 2 \text{ y } y\ell - 1 \text{ es un cuadrado.}$$

Nótese que tal q debe ser primo o potencia de primo. La curva concreta puede obtenerse teniendo en cuenta la Proposición 3.5 de [11].

Ejemplos concretos para algunos valores de los parámetros x, y pueden verse en la tabla II.

Tabla II
CURVAS ELÍPTICAS CON GRADO DE INMERSIÓN 1 Ó 2

ℓ	x, y	q	Curva	k
73	0,2	$4\ell^2 + 1 = 21317$	E_0	1
41	2,2	$8\ell^2 + 4\ell + 1 = 13613$	E_2	1
79	1,1	$2\ell^2 + 2\ell + 1 = 12641$	$E_1\ (E_3)$	1
101	1,1	$\ell^2 + \ell + 1 = 10301$	E_0	2
101	3,1	$9\ell^2 + \ell - 1 = 91909$	E_2	2
1013	2,2	$4\ell^2 + 2\ell - 1 = 4106701$	$E_1\ (E_3)$	2

III. ALGORITMO DE MILLER CON FORMAS NO ADYACENTES

Como se ha dicho en la introducción, un emparejamiento de orden ℓ, e_ℓ, para una curva elíptica E sobre el cuerpo finito $\mathbb{F}_q$ (usualmente ℓ primo y divisor del cardinal de la curva) es una aplicación bilineal que asigna a un par de puntos P, Q de la curva una raíz de orden ℓ de la unidad. Supondremos que estas raíces se encuentran inmersas en $\mathbb{F}_{q^k}$ siendo k el grado de inmersión de la curva (ver definición 1). En el caso particular del emparejamiento de Weil, los puntos P, Q son ambos de ℓ-torsión (designaremos por $E(\mathbb{F}_{q^k})[\ell]$, el subgrupo de puntos de ℓ-torsión de E). Nótese que un punto de ℓ-torsión no necesariamente está definido en el cuerpo base pero si en $\mathbb{F}_{q^k}$ ([2]).

Para los puntos P, Q, el valor del emparejamiento e_ℓ se calcula como el cociente

$$e_\ell(P, Q) = \frac{f_{\ell,P}(Q + R) f_{\ell,Q}(S)}{f_{\ell,P}(R) f_{\ell,Q}(P + S)}, \tag{1}$$

donde R, S son puntos auxiliares de la curva y las funciones $f_{\ell,P}, f_{\ell,Q}$ son cociente de dos polinomios en dos variables (para más detalles ver [9]). La dificultad de este cálculo reside en la construcción de estas funciones. El algoritmo de Miller ([10]) permite realizar dicha construcción de forma eficiente. Las funciones $f_{\ell,T}$ para un punto cualquiera T, se obtienen de forma recursiva a partir de las siguientes identidades:

$$\begin{aligned} f_{0,T} &= f_{1,T} = 1 \\ f_{m+n,T} &= f_{m,T} f_{n,T} g_{mT,nT} \end{aligned} \tag{2}$$

donde $g_{U,V} = \frac{L_{U,V}}{L_{(U+V),-(U+V)}}$ y $L_{U,V} = 0$ es la ecuación de la recta que pasa por los dos puntos U y V, o la recta tangente si $U = V$. Nótese que $L_{(U+V,-(U+V)}$ es la recta vertical que pasa por el punto $U+V$ y por tanto sólo depende de su abscisa.

Por tanto, dado un entero ℓ y un punto T de orden ℓ, el cálculo de la función $f_{\ell,T}$ se realiza de la siguiente forma:

Algoritmo 6:
Input: $T \in E(\mathbb{F}_{q^k})[\ell]$,
$\ell = (\ell_{r-1}, \ell_{r-2}, \ldots, \ell_0)_2$ (representación binaria de ℓ)
Output: $f_{\ell,T}$
$f \leftarrow 1$, $W \leftarrow P$.
for i from $r-2$ to 0 **do**
$f \leftarrow f^2 \frac{L}{L'}$
(L recta tangente en W, L' recta vertical por $2W$)

$W \leftarrow 2W$
if $\ell_i = 1$ **then**
$f \leftarrow f\frac{L}{L'}$
(L recta que pasa por T y W, L' recta vertical por $T+W$)
$W \leftarrow T+W$
end if
end for
RETURN f

El algoritmo anterior se basa en una estrategia de cuadrados repetidos (doblado y suma) en la que se hace uso de la expresión binaria del entero ℓ. Es claro que el número de iteraciones en el algoritmo anterior es $log_2(\ell)$ y el número de *operaciones* (cálculo de rectas) a realizar depende del peso de Hamming de ℓ. Por tanto, el algoritmo será más rápido cuanto menor sea tal peso.

Los algoritmos que se basan en este tipo de estrategia pueden adaptarse para usar, en lugar de la representación binaria, cualquier otra cadena de adición-sustracción. En particular, es posible minimizar el número de bits no nulos en la representación binaria de un número usando el siguiente concepto en el que se permiten restas (hay que tener en cuenta que la resta de puntos de una curva elíptica se puede hacer con el mismo coste que una suma).

Definición 7 ([8]): Dado un entero n, se define la forma no adyacente de n ($NAF(n)$) como una representación dada por $n = \sum_{i=0}^{r-1} k_i 2^i$ donde $k_i \in \{0, \pm 1\}$, $k_{r-1} \neq 0$ y para todo $1 \leq i \leq r-1$, $k_{i-1}k_i = 0$.

A partir de las formas recurrentes dadas en (2), teniendo en cuenta que $f_{0,T} = f_{1,T} f_{-1,T} g_{T,-T}$, es posible determinar los pasos a añadir en el Algoritmo 6 cuando se cambia la representación binaria del entero ℓ por su representación en forma no adyacente. Más concretamente, habría que añadir las órdenes siguientes:

if $\ell_i = -1$ **then**
$f \leftarrow f\frac{L}{L_1 L'}$
(L_1 recta vertical por T, L recta que pasa por $-T$ y W, L' recta vertical por $W-T$)
$W \leftarrow W-T$
end if

Sin embargo, existe otra forma de obtener la función $f_{\ell,T}$ a partir de la representación NAF de ℓ. Para ello, se sustituirían las instrucciones anteriores por las siguientes:

if $\ell_i = -1$ **then**
$f \leftarrow f\frac{L'}{L}$
(L' recta vertical por W y L la recta que pasa por los puntos T y $-W$)
$W \leftarrow W-T$
end if

Téngase en cuenta que en este caso cuando $\ell_i = -1$, se calculan dos rectas mientras que en la versión anterior son necesarias tres rectas. Aunque la forma natural de trasladar el algoritmo clásico de Miller a su versión usando la representación NAF es la propuesta primera, la búsqueda de una solución computacionalmente más eficiente nos ha llevado a esta segunda forma.

Tenemos entonces, tres formas diferentes para el cálculo de la función $f_{\ell,T}$: método binario usando el algoritmo original de Miller y dos versiones NAF I y NAF II utilizando la representación en forma no adyacente. En realidad, la función obtenida en cada caso es diferente, sin embargo, al calcular el emparejamiento con la fórmula (1), el resultado obtenido con cualquiera de las tres funciones es el mismo, ya que dicho valor $e_\ell(P,Q)$ es independiente del camino seguido ([3]).

IV. Ejemplos numéricos

Se han implementado las diferentes versiones mostradas del algoritmo de Miller para el cálculo de la función $f_{\ell,T}$ usando Maple v.13 en un ordenador con un procesador Intel Core 2 Duo de 2.13 GHz y 2 GB de memoria RAM.

En la tabla III se muestran los tiempos de ejecución (en media) de los tres algoritmos: es decir, usando la representación binaria de ℓ en el primer caso, y en los otros dos tomando la forma no adyacente de ℓ.

Se han calculado previamente curvas elípticas con grado de inmersión 1 respecto a algún primo ℓ de tamaño 20, 40, 60 y 100 bits (con sus correspondientes puntos de orden ℓ) para las dos primeras familias de curvas elípticas ordinarias mostradas en la Tabla II.

El primo ℓ se ha escogido de dos formas distintas. En el primer caso no se ha considerado ninguna restricción sobre él mientras que en el segundo caso, se han escogido primos ℓ cuya representación en forma no adyacente tiene peso de Hamming pequeño. Para cada tipo se han tomado 500 de estos primos para cada longitud (100 cuando ℓ tiene 100 bits).

Tabla III
Tiempos de ejecución del algoritmo de Miller (ms)

Curva/primo	Nº de bits ℓ	Binario	NAF I	NAF II
E_0, $p = 4\ell^2 + 1$	20	4.1	3.8	3.8
	40	12.1	10.6	10.6
	60	23.5	21.9	20.7
	100	61.4	56.3	54.1
E_0, $p = 4\ell^2 + 1$ $w_{NAF}(\ell) \leq 7$	20	3.7	3.4	3.2
	40	10.4	8.7	8.7
	60	21.8	17.6	17
	100	56.3	44.9	43.6
E_2, $p = 8\ell^2 + 4\ell + 1$	20	6.3	6	5.9
	40	11.2	10.1	9.8
	60	33.8	30.7	28.2
	100	65.6	58.8	56.4
E_2, $p = 8\ell^2 + 4\ell + 1$ $w_{NAF}(\ell) \leq 8$	20	5.5	5.1	4
	40	13.8	12.3	11.2
	60	29.4	23.8	22.5
	100	80.9	60.2	59.9

Tal como era de esperar, en ambos casos las versiones basadas en formas no adyacentes son más eficientes y, en general, la versión NAF II es mucho más rápida.

El mayor descenso en tiempo de ejecución se encontraría cuando ℓ es un primo de Mersenne, 2^p-1, cuya representación binaria es $111\ldots111$, mientras que su expresión en forma no adyacente es $100\ldots000$-1.

Finalmente, hemos considerado la curva elíptica E_0 sobre $\mathbb{F}_q$, con $q = 4\ell^2 + 1$ siendo $\ell = 2^{258} + 2^{242} - 1$. Este primo ℓ es un ejemplo de los denominados primos de Solinas, recomendados para su uso en criptografía elíptica. Las expresiones binaria y NAF de este primo ℓ son:

$$1000000000000000\overbrace{111\ldots11}^{242bits}$$
$$1000000000000000\underbrace{1000\ldots0\text{-}1}_{243bits}$$

Los tiempos de ejecución de las tres versiones del algoritmo de Miller para este ejemplo son 468 ms para el caso binario, 234 ms para NAF I y 218 ms para NAF II.

V. Conclusión

Partiendo de las curvas con el denominado j-invariante igual a 1728, o lo que es lo mismo las curvas con ecuación $y^2 = x^3 + Ax$, hemos obtenido condiciones que permiten asegurar que dichas curvas son buenas para su empleo en Criptografía Basada en la Identidad. En particular, hemos proporcionado ejemplos concretos de curvas elípticas con grado de inmersión 1 y 2.

La herramienta principal utilizada en Criptografía Basada en la Identidad son los emparejamientos. Nos hemos ocupado también de la implementación del algoritmo de Miller que es el algoritmo básico en el cálculo de los emparejamientos, tanto en su versión original como en variantes propuestas en el trabajo basadas en la forma no adyacente del primo ℓ. De esta forma, se observa que las variantes propuestas, permiten un ahorro significativo en los tiempos de ejecución del algoritmo de Miller.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad con los proyectos MTM2010-21580-C02-01/02 y MTM2010-16051.

Referencias

[1] C. Cocks, "An identity based encryption scheme based on quadratic residues", *Proceedings of the 8th IMA Int. Conf. on Cryptology and Coding*, Springer LNCS 2260, pp. 360–363, 2001.

[2] R. Balasubramanian, N. Koblitz "The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm",*Journal of Cryptology* vol 11 (2), pp. 141–145, 1998.

[3] I. Blake, G. Seroussi, N. Smart, "Advances in Elliptic Curve Cryptography". London Mathematical Society, LNS 371. Cambdridge University Press, 2005.

[4] D. Boneh, C. Gentry, M. Hamburg. "Space-Efficient Identity Based Encryption without Pairings" *FOCS'07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 647-657, 2007.

[5] D. Boneh, M. Franklin "Identity based Encryption from the Weil Pairing" *Advances in Cryptology - CRYPTO 2001* , Springer LNCS 2139, pp. 213–229, 2001.

[6] F. Brezing, A. Weng, "Elliptic curves suitable for pairings based cryptography", *Designs, Codes and Cryptography* 37, pp. 133–141, 2005.

[7] D. Freeman, M. Scott, E. Teske, "A taxonomy of pairing-friendly elliptic curves", *Journal of Cryptology*, vol 23(2), pp. 224–280, 2010.

[8] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer, 2004.

[9] A. Menezes, "Elliptic Curves Public Key Cryptography" , Kluwer, 1993.

[10] V. Miller "The Weil Pairing, and its Efficient Calculation" *Journal of Cryptology* , vol. 17, pp. 235–261, 2004.

[11] C. Munuera, J. Tena, "An algorithm to compute the number of points on elliptic curves of j-invariant 0 or 1728 over a finite field", *Rendiconti del Circolo Matematico di Palermo*, Serie 2, tomo XLII, pp. 106–116, 1993.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes", *Advances in cryptology - CRYPTO 1984*, Springer LNCS196, pp. 47–53, 1985.