Conteo de puntos

en curvas hiperelípticas

Seminario Interuniversitario
de criptografía – CyTeD

Nicolas Thériault

Departamento de Matemática y Ciencia de la Computación,
Universidad de Santiago de Chile.

Red CyTeD de criptografía

- Brasil
  - Ricardo Dahab, Julio López
- Chile
  - Rodrigo Abarzúa, Alejandro Hevia, Edgardo Riquelme, Nicolas Thériault
- Colombia
  - John Baena, Daniel Cabarcas, Valérie Gauthier
- Cuba
  - Mijaíl Borges Quintana, Teresa Bernarda Pagés López, Luis Ramiro Piñeiro Díaz
- España
  - Gora Adj, Josep Maria Miret, Jordi Pujolàs, Francesc Sebé, Magda Valls
- México
  - Cuauhtemoc Mancillas, Guillermo Morales, Francisco Rodríguez-Henríquez
- Uruguay
  - Eduardo Canales, Claudio Qureshi, Alfredo Viola

Red CyTeD de criptografía

1. Mejora de la búsqueda de curvas hiperelípticas aleatorias seguras para aplicaciones criptográficas

2. Estudio de los principales candidatos para la criptografía resistente a computadores cuánticos:

    1. Criptografía pos-cuántica basada en isogenias
    2. Criptografía pos-cuántica basada en códigos
    3. Criptografía pos-cuántica basada en reticulados
    4. Criptografía pos-cuántica basada en polinomios multivariados

3. Estudio de la construcción de cifrados (simetricos) autenticados basados en permutaciones públicas

Algunos trabajos previos

- Doctorado en U. de Toronto, 2003

- Postdoc en U. de Duisburg-Essen (IEM), 2003-2004
- Postdoc en U. de Waterloo (CACR), 2004-2006
- Postdoc en Toronto (Instituto Fields), 2006

- Profesor en U. de Talca, 2007-2010
- Profesor en U. del Bío-Bío, 2011-2015
- Profesor en U. de Santiago de Chile, 2015-...

Algunos temas de trabajos

- Ataques de Weil
- Calculo de indice
- Side Channel Attacks y resistencia
  - representaciones uniformes de enteros
  - formulas explicitas uniformizadas
  - SVA
- Aritmetica de grupo
  - formulas explicitas e implementación
  - representaciones de enteros en base doble
  - bisecciones/halving
- Factorización de polinomio

¿Qué necesitamos de una curva algebraica?

- Aritmetica de grupo eficiente
  - algoritmo de Cantor
  - formulas explicitas

¿Qué necesitamos de una curva algebraica?

- Aritmetica de grupo eficiente
  - algoritmo de Cantor
  - formulas explicitas
- Resistente a ataques de raíz-cuadrada (Pollard Rho, etc.)
  - tamaño del cuerpo de definición

¿Qué necesitamos de una curva algebraica?

- Aritmetica de grupo eficiente
  - algoritmo de Cantor
  - formulas explicitas
- Resistente a ataques de raíz-cuadrada (Pollard Rho, etc.)
  - tamaño del cuerpo de definición
- Resistente al calculo de indices
  - genero a lo más 3
  - curva hiperelíptica

¿Qué necesitamos de una curva algebraica?

- Resistente a Pohlig-Helman
  - orden de grupo con un factor primo grande
  - requiere calcular el orden de grupo (conteo de puntos)

¿Qué necesitamos de una curva algebraica?

- Resistente a Pohlig-Helman
  - orden de grupo con un factor primo grande
  - requiere calcular el orden de grupo (conteo de puntos)
- No permite ataques especiales
  - ataques de sub-grupos (multiplicar por el cofactor)
  - cambios de curvas (verificar los divisores)
  - bajada de Weil, etc (tipo de cuerpo de definición)
  - ataque de Smith en genero 3 (ecuación de la curva)

¿Qué necesitamos de una curva algebraica?

- Resistente a Pohlig-Helman
  - orden de grupo con un factor primo grande
  - requiere calcular el orden de grupo (conteo de puntos)
- No permite ataques especiales
  - ataques de sub-grupos (multiplicar por el cofactor)
  - cambios de curvas (verificar los divisores)
  - bajada de Weil, etc (tipo de cuerpo de definición)
  - ataque de Smith en genero 3 (ecuación de la curva)
- Resistente a los Side-Channel
  - SPA, ZPA, SVA, etc
  - algunas curvas tienen más riesgo que otras

Objetivo actual en el proyecto CyTeD

Línea 1: Nuevas curvas para Diffie-Hellmann

Objetivos específicos:

- Método eficiente de trisección de puntos en curvas elípticas
- Técnicas para obtener ℓ-secciones de puntos en curvas elípticas
- Diseño de métodos de bisección de divisores en curvas hiperelípticas de género g
- Mejoras en la eficiencia del algoritmo SEA para género 2
- Búsqueda de curvas seguras para aplicaciones criptográficas

Algoritmos de tipo Schoof

Para calcular el orden de grupo, se utiliza el polinomio característico del grupo de la curva:

$$\chi(T) = T^{2g} + a_1 T^{2g-1} + \ldots + a_{g-1} T^{g+1} + a_g T^g + a_{g-1} q T^{g-1} + \ldots a_1 q^{g-1} T + q^g$$

donde T es el Frobenius sobre $\mathbb{F}_q$ y los a_i satisfacen

$$|a_i| \leq \binom{2g}{i} q^{i/2} \quad .$$

Para todo elemento D del grupo (sobre $\overline{\mathbb{F}_q}$), tenemos

$$\chi(T) D = 0$$

Algoritmos de tipo Schoof

Para calcular el orden de grupo, se utiliza el polinomio característico del grupo de la curva:

$$\chi(T) = T^{2g} + a_1 T^{2g-1} + \ldots + a_{g-1} T^{g+1} + a_g T^g + a_{g-1} q T^{g-1} + \ldots a_1 q^{g-1} T + q^g$$

donde T es el Frobenius sobre $\mathbb{F}_q$ y los a_i satisfacen

$$|a_i| \leq \binom{2g}{i} q^{i/2} \quad .$$

Para todo elemento D del grupo (sobre $\overline{\mathbb{F}_q}$), tenemos

$$\chi(T)D = 0$$

Teorema (Hasse Weil)

Los $2g$ valores propios ζ_j satisfacen $|\zeta_j| = \sqrt{q}$ y van en pares $\zeta_j \cdot \zeta 2g - j = q$.

Algoritmos de tipo Schoof

Sea

$$\chi_\ell(T) = T^{2g} + b_1 T^{2g-1} + \ldots + b_{g-1} T^{g+1} + b_g T^g + b_{g-1} k T^{g-1} + \ldots b_1 k^{g-1} T + k^g$$

donde

$$b_i \equiv a_i \bmod \ell \quad \text{y} \quad k \equiv q \bmod \ell \ ,$$

Algoritmos de tipo Schoof

Sea

$$\chi_\ell(T) = T^{2g} + b_1 T^{2g-1} + \ldots + b_{g-1} T^{g+1} + b_g T^g + b_{g-1} k T^{g-1} + \ldots b_1 k^{g-1} T + k^g$$

donde

$$b_i \equiv a_i \bmod \ell \quad \text{y} \quad k \equiv q \bmod \ell \ ,$$

entonces para todo $D_\ell \in Jac(C)[\ell]$ (los divisores de ℓ-torsión),

$$\chi_\ell(T) D_\ell = 0$$

Algoritmos de tipo Schoof

Sea

$$\chi_\ell(T) = T^{2g} + b_1 T^{2g-1} + \ldots + b_{g-1} T^{g+1} + b_g T^g + b_{g-1} k T^{g-1} + \ldots b_1 k^{g-1} T + k^g$$

donde

$$b_i \equiv a_i \bmod \ell \quad \text{y} \quad k \equiv q \bmod \ell \ ,$$

entonces para todo $D_\ell \in Jac(C)[\ell]$ (los divisores de ℓ-torsión),

$$\chi_\ell(T) D_\ell = 0$$

y se puede deducir los coeficientes de $\chi(T)$ módulo ℓ utilizando los divisores de ℓ-torsión (los elementos de orden ℓ).

Algoritmos de tipo Schoof

Sea

$$\chi_\ell(T) = T^{2g} + b_1 T^{2g-1} + \ldots + b_{g-1} T^{g+1} + b_g T^g + b_{g-1} k T^{g-1} + \ldots b_1 k^{g-1} T + k^g$$

donde

$$b_i \equiv a_i \bmod \ell \quad \text{y} \quad k \equiv q \bmod \ell \ ,$$

entonces para todo $D_\ell \in Jac(C)[\ell]$ (los divisores de ℓ-torsión),

$$\chi_\ell(T) D_\ell = 0$$

y se puede deducir los coeficientes de $\chi(T)$ módulo ℓ utilizando los divisores de ℓ-torsión (los elementos de orden ℓ).

Teorema (Schoof)

Se puede determinar completamente el polinomio característico combinando la información módulo $O(\log q)$ primos de tamaño $O(\log q)$.

¿Dónde está el desafio?

La complejidad es muy alta:

- Hay $\ell^{2g} - 1$ divisores de ℓ-torsión
  - pueden vivir en una extensión de cuerpo de grado $\ell^{2g} - 1$

¿Dónde está el desafio?

La complejidad es muy alta:

- Hay $\ell^{2g} - 1$ divisores de ℓ-torsión
  - pueden vivir en una extensión de cuerpo de grado $\ell^{2g} - 1$
- Requiere tomar potencias q (Frobenius)
- Hay ℓ^g combinaciones de coeficientes posibles

¿Dónde está el desafio?

La complejidad es muy alta:

- Hay $\ell^{2g} - 1$ divisores de ℓ-torsión
    - pueden vivir en una extensión de cuerpo de grado $\ell^{2g} - 1$
- Requiere tomar potencias q (Frobenius)
- Hay ℓ^g combinaciones de coeficientes posibles
- Da $O((\log q)^{3g})$ operaciones en $\mathbb{F}_q$ para cada ℓ (con aritmetica rápida)

¿Dónde está el desafio?

La complejidad es muy alta:

- Hay $\ell^{2g} - 1$ divisores de ℓ-torsión
  - pueden vivir en una extensión de cuerpo de grado $\ell^{2g} - 1$
- Requiere tomar potencias q (Frobenius)
- Hay ℓ^g combinaciones de coeficientes posibles
- Da $O((\log q)^{3g})$ operaciones en $\mathbb{F}_q$ para cada ℓ (con aritmetica rápida)
- A tamaños criptográficos, podemos tener que mirar unas 1000 curvas para encontrar una buena.

¿Dónde está el desafio?

Para genero 1 (curvas elípticas), $O((\log q)^{4+\epsilon})$, podría ser manejable

- hay mejoras disponibles (Elkies, Atkin), reduce de un factor de $\log q$ (un poco más)
- implementaciones muy eficientes

¿Dónde está el desafio?

Para genero 1 (curvas elípticas), $O((\log q)^{4+\epsilon})$, podría ser manejable

- hay mejoras disponibles (Elkies, Atkin), reduce de un factor de $\log q$ (un poco más)
- implementaciones muy eficientes

Para genero 2, $O((\log q)^{7+\epsilon})$, hay unos pocos resultados

- el record (Gaudry y Schost, 2012) utilizó unos $0.6 \cdot 10^6$ horas CPU.
- queremos hacerlo mejor

¿Dónde está el desafio?

Para genero 1 (curvas elípticas), $O((\log q)^{4+\epsilon})$, podría ser manejable

- hay mejoras disponibles (Elkies, Atkin), reduce de un factor de $\log q$ (un poco más)
- implementaciones muy eficientes

Para genero 2, $O((\log q)^{7+\epsilon})$, hay unos pocos resultados

- el record (Gaudry y Schost, 2012) utilizó unos $0.6 \cdot 10^6$ horas CPU.
- queremos hacerlo mejor

Para genero 3, $O((\log q)^{10+\epsilon})$, no hay resultados prácticos.

- incluso, se propuso utilizar curvas de orden desconocido como primitivas (calcular el orden es parte del secreto).

Mejoras I

Dirección: Elkies y Atkin

En genero 1 (curvas elípticas):

- identificar primos donde los valores propios de χ_ℓ están en $\mathbb{F}_\ell$ (en ves de $\mathbb{F}_{\ell^2}$).
- trabaja en una extensión de cuerpo de grado $\ell - 1$ en vez de $\ell^2 - 1$
- funciona para la mitad de los primos ℓ
- elimina unos coeficientes (la mitad)
- calcula los "vectores propios" es más directo
- trabaja con una "buena base" de las ℓ-torsiones

Mejoras I

Dirección: Elkies y Atkin

En genero 1 (curvas elípticas):

- identificar primos donde los valores propios de χ_ℓ están en $\mathbb{F}_\ell$ (en ves de $\mathbb{F}_{\ell^2}$).
- trabaja en una extensión de cuerpo de grado $\ell - 1$ en vez de $\ell^2 - 1$
- funciona para la mitad de los primos ℓ
- elimina unos coeficientes (la mitad)
- calcula los "vectores propios" es más directo
- trabaja con una "buena base" de las ℓ-torsiones

En genero 2, hay unos avances, pero no es completo

- Hay avances, pero no está completo
- funciona para un proporción menor de los primos ℓ
- el calculo de los "vectores propios" es mucho más costoso (problema)
- pocas veces (1 de cada 24 primos ℓ) da una base completa de las ℓ-torsiones

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

En genero 2:

- utilizado por Gaudry y Schost
- para ir de módulo ℓ^s a ℓ^{s+1}, aumenta el costo por un factor de ℓ

- sube una potencia a la vez ("liftings")

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

En genero 2:

- utilizado por Gaudry y Schost
- para ir de módulo ℓ^s a ℓ^{s+1}, aumenta el costo por un factor de ℓ
- sube una potencia a la vez ("liftings")
- requiere calcular ℓ-secciones (pre-imagenes de la multiplicación por ℓ)

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

En genero 2:

- utilizado por Gaudry y Schost
- para ir de módulo ℓ^s a ℓ^{s+1}, aumenta el costo por un factor de ℓ
- utilizar generagores de las ℓ^s torsiones en vez de todas las ℓ^s torsiones
- sube una potencia a la vez (“liftings”)

- requiere calcular ℓ-secciones (pre-imagenes de la multiplicación por ℓ)

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

En genero 2:

- utilizado por Gaudry y Schost
- para ir de módulo ℓ^s a ℓ^{s+1}, aumenta el costo por un factor de ℓ
- utilizar generagores de las ℓ^s torsiones en vez de todas las ℓ^s torsiones
- sube una potencia a la vez ("liftings")
- ej: módulo 2^4 (extensión 120) es más barrato que módulo 2 (extensión 15) + módulo 7 (extensión 2400)
- requiere calcular ℓ-secciones (pre-imagenes de la multiplicación por ℓ)

Mejoras II

Dirrección: utilizar potencias de los primos *ell* más pequeños

En genero 1 (curvas elípticas):

- ayuda un poco (menos que Elkies y Atkin)
- poco estudiado

En genero 2:

- utilizado por Gaudry y Schost
- para ir de módulo ℓ^s a ℓ^{s+1}, aumenta el costo por un factor de ℓ
- utilizar generagores de las ℓ^s torsiones en vez de todas las ℓ^s torsiones
- sube una potencia a la vez ("liftings")
- ej: módulo 2^4 (extensión 120) es más barrato que módulo 2 (extensión 15) + módulo 7 (extensión 2400)
- requiere calcular ℓ-secciones (pre-imagenes de la multiplicación por ℓ)
- Ejemplo (común) de combinaciones de módulos en Gaudry y Schost:
  - 2^{17}, 3^6, 5^3, 7^2, y luego 11, 13, 17, 19, 23, 29 y 31
  - corresponde a 17, 9, 8, 6, 3, . . . , 5 bits de información (69 bits en total)

ℓ-secciones

Dado un punto/divisor D_ℓ, encontrar las pre-imagenes de la multiplicación por ℓ

$$\mathcal{D} = \{D_1 \in Jac(C) \mid [\ell]D_1 = D_\ell\} \quad .$$

Si tenemos los divisores de ℓ-torsión, i.e.

$$Jac(C)[\ell] = \{D \in Jac(C) \mid [\ell]D = 0\} \quad ,$$

entonces dado una primera ℓ-sección D_1, tendremos

$$\mathcal{D} = D_1 + Jac(C)[\ell] \quad .$$

ℓ-secciones

- Gaudry y Schost obtienen las ℓ-secciones escribiendo un sistema de ecuaciones (no lineales) y resolviéndolo a través de varias técnicas (bases de Groebner, resultantes, etc).
    - El sistema tiene ℓ^{2g} soluciones, y muchas veces deben descartar raíces falsas.
    - Reducir el sistema a ecuaciones en una variable es el costo dominante.

ℓ-secciones

- Gaudry y Schost obtienen las ℓ-secciones escribiendo un sistema de ecuaciones (no lineales) y resolviéndolo a través de varias técnicas (bases de Groebner, resultantes, etc).
  - El sistema tiene ℓ^{2g} soluciones, y muchas veces deben descartar raíces falsas.
  - Reducir el sistema a ecuaciones en una variable es el costo dominante.
- Para $\ell = 3$ en género 2 se puede re-trabajar el sistema para limpiar las raíces falsas y asociar las soluciones a los ceros de un polinomio de grado 81 (Riquelme y T., 2018).
  - Una vez conocido este polinomio, las trisecciones se pueden calcular mucho más rápido.
  - La factorización queda como costo principal.

ℓ-secciones

- Gaudry y Schost obtienen las ℓ-secciones escribiendo un sistema de ecuaciones (no lineales) y resolviéndolo a través de varias técnicas (bases de Groebner, resultantes, etc).
  - El sistema tiene ℓ^{2g} soluciones, y muchas veces deben descartar raíces falsas.
  - Reducir el sistema a ecuaciones en una variable es el costo dominante.
- Para $\ell = 3$ en género 2 se puede re-trabajar el sistema para limpiar las raíces falsas y asociar las soluciones a los ceros de un polinomio de grado 81 (Riquelme y T., 2018).
  - Una vez conocido este polinomio, las trisecciones se pueden calcular mucho más rápido.
  - La factorización queda como costo principal.
  - Todavía queríamos hacerlo mejor y hacerlo para otros ℓ...

bisecciones

Para las bisecciones, Gaudry y Schost pasan por las superficies de Kummer

- más limpio (no hay raíces falsas)
- requiere cuatro raíces cuadradas en vez de factorizar un polinomio de grado 16
- requieren que todas las 2-torsiones estén en $\mathbb{F}_q$ (limita la forma de la curva)

bisecciones

Para las bisecciones, Gaudry y Schost pasan por las superficies de Kummer

- más limpio (no hay raíces falsas)
- requiere cuatro raíces cuadradas en vez de factorizar un polinomio de grado 16
- requieren que todas las 2-torsiones estén en $\mathbb{F}_q$ (limita la forma de la curva)

Desarrollamos (Miret, Pujolàs y T., 2015) un método alternativo:

- también es limpio
- requiere cuatro raíces cuadradas y un poco de álgebra lineal
- todo queda en la curva

bisecciones

Para las bisecciones, Gaudry y Schost pasan por las superficies de Kummer

- más limpio (no hay raíces falsas)
- requiere cuatro raíces cuadradas en vez de factorizar un polinomio de grado 16
- requieren que todas las 2-torsiones estén en $\mathbb{F}_q$ (limita la forma de la curva)

Desarrollamos (Miret, Pujolàs y T., 2015) un método alternativo:

- también es limpio
- requiere cuatro raíces cuadradas y un poco de álgebra lineal
- todo queda en la curva
- no requiere las 2-torsiones en $\mathbb{F}_q$
- en algunas curvas, se puede trabajar con un solo generador de $Jac(C)[\ell^s]$ en vez de 4

Raíces ℓ-esimas

- Es más eficiente resolver $2g$ raíces ℓ-esimas (factorizar $2g$ polinomios de grado ℓ) que factorizar un solo polinomio de grado ℓ^{2g}.

Raíces ℓ-esimas

- Es más eficiente resolver $2g$ raíces ℓ-esimas (factorizar $2g$ polinomios de grado ℓ) que factorizar un solo polinomio de grado ℓ^{2g}.

- Como trabajamos en extensiones de cuerpo grandes, algunas técnicas de factorización especiales (en redacción) pueden reducir los costos (Avanzi y T., von zur Gathen y T.)

Raíces ℓ-esimas

- Es más eficiente resolver $2g$ raíces ℓ-esimas (factorizar $2g$ polinomios de grado ℓ) que factorizar un solo polinomio de grado ℓ^{2g}.
- Como trabajamos en extensiones de cuerpo grandes, algunas técnicas de factorización especiales (en redacción) pueden reducir los costos (Avanzi y T., von zur Gathen y T.)
- La manera de definir la extensión de cuerpo donde trabajamos es importante (trabajo en curso)

Raíces ℓ-esimas

- Es más eficiente resolver $2g$ raíces ℓ-esimas (factorizar $2g$ polinomios de grado ℓ) que factorizar un solo polinomio de grado ℓ^{2g}.
- Como trabajamos en extensiones de cuerpo grandes, algunas técnicas de factorización especiales (en redacción) pueden reducir los costos (Avanzi y T., von zur Gathen y T.)
- La manera de definir la extensión de cuerpo donde trabajamos es importante (trabajo en curso)
- Se obtiene una reducción de costos muy importante

ℓ-sectores

- Dado unos generadores $W_1, W_2, \ldots, W_{2g}$ del grupo de ℓ-torsión, podemos definir un polinomio $h_i(x, y)$ tales que

  para todo i, $h_i(-D) = {\omega_i}^{\ell}$ para algún $\omega_i \in \mathbb{F}_q \iff D$ admite ℓ-secciones en $\mathbb{F}_q$.

- Además, cada $2g$-tuple $(\omega_1, \omega_2, \ldots, \omega_{2g})$ está asociado a exactamente una ℓ-sección

ℓ-sectores

- Dado unos generadores $W_1, W_2, \ldots, W_{2g}$ del grupo de ℓ-torsión, podemos definir un polinomio $h_i(x, y)$ tales que

  $$\text{para todo } i,\ h_i(-D) = {\omega_i}^{\ell} \text{ para algún } \omega_i \in \mathbb{F}_q \iff D \text{ admite } \ell\text{-secciones en } \mathbb{F}_q\ .$$

- Además, cada $2g$-tuple $(\omega_1, \omega_2, \ldots, \omega_{2g})$ está asociado a exactamente una ℓ-sección

- Llamamos ℓ-sectores una combinación de ℓ-raíces asociadas a un divisor D_ℓ

- Por la reciprocidad de Weil, a cada ℓ-sección D_1 se asocia un polinomio $g(x, y)$ cuyo divisor principal es $\ell D_1 + [-1]D_\ell$
  - se puede calcular $g(x, y)$ con el algoritmo de Miller (pairings)
  - los coeficientes de $g(x, y)$ dependen solamente de las coordenadas de D_1
  - tenemos $g(W_i) = \omega_i h_i(D_\ell)$

ℓ-sectores

- Dado unos generadores $W_1, W_2, \ldots, W_{2g}$ del grupo de ℓ-torsión, podemos definir un polinomio $h_i(x,y)$ tales que

  para todo i, $h_i(-D) = {\omega_i}^{\ell}$ para algún $\omega_i \in \mathbb{F}_q \iff D$ admite ℓ-secciones en $\mathbb{F}_q$.

- Además, cada $2g$-tuple $(\omega_1, \omega_2, \ldots, \omega_{2g})$ está asociado a exactamente una ℓ-sección

- Llamamos ℓ-sectores una combinación de ℓ-raíces asociadas a un divisor D_ℓ

- Por la reciprocidad de Weil, a cada ℓ-sección D_1 se asocia un polinomio $g(x,y)$ cuyo divisor principal es $\ell D_1 + [-1]D_\ell$
  - se puede calcular $g(x,y)$ con el algoritmo de Miller (pairings)
  - los coeficientes de $g(x,y)$ dependen solamente de las coordenadas de D_1
  - tenemos $g(W_i) = \omega_i h_i(D_\ell)$

- Da un sistema de $2g$ ecuaciones en $2g$ variables.

ℓ-sectores, estrategia I

Linearización:

- Asociar nuevas variables a potencias y productos de las variables originales
- Da un sistema lineal con $\ell^{2g-1}+1$ variables

ℓ-sectores, estrategia I

Linearización:

- Asociar nuevas variables a potencias y productos de las variables originales
- Da un sistema lineal con $\ell^{2g-1}+1$ variables
- Se obtienen más ecuaciones con los $W_t = [t_1]W_1 + [t_2]W_2 + \ldots + [t_{2g}]W_{2g}$
- Se define un ω_t consistente por la reciprocidad de Weil (sin calcular raíces)

ℓ-sectores, estrategia I

Linearización:

- Asociar nuevas variables a potencias y productos de las variables originales
- Da un sistema lineal con $\ell^{2g-1}+1$ variables
- Se obtienen más ecuaciones con los $W_t = [t_1]W_1 + [t_2]W_2 + \ldots + [t_{2g}]W_{2g}$
- Se define un ω_t consistente por la reciprocidad de Weil (sin calcular raíces)
- Resolver el sistema tiene complejidad $O(\ell^{6g-3})$:
  - No depende del s en las ℓ^s torsiones
  - Para curvas elípticas, complejidad $O(\ell^3)$ (razonable)
  - Para genero 2, complejidad $O(\ell^9)$ (sirve solo para ℓ muy pequeño)
  - Para genero 3, demasiado alto

ℓ-sectores, estrategia II

Solución directa:

- Utilizar bases de Groebner
- El sistema tiene solución única

ℓ-sectores, estrategia II

Solución directa:

- Utilizar bases de Groebner
- El sistema tiene solución única
- Se pueden dar pesos a las variables:
  - $w(u_{1,g-1}) = 2$, $w(u_{1,g-2}) = 4$, ..., $w(u_{1,0}) = 2g$
  - $w(v_{1,g-1}) = 3$, $w(v_{1,g-2}) = 5$, ..., $w(v_{1,0}) = 2g+1$

ℓ-sectores, estrategia II

Solución directa:

- Utilizar bases de Groebner
- El sistema tiene solución única
- Se pueden dar pesos a las variables:
  - $w(u_{1,g-1}) = 2,\ w(u_{1,g-2}) = 4,\ \ldots,\ w(u_{1,0}) = 2g$
  - $w(v_{1,g-1}) = 3,\ w(v_{1,g-2}) = 5,\ \ldots,\ w(v_{1,0}) = 2g + 1$
- ¿Complejidad?

Otro objetivo de interés especial

Línea 2.iii: Criptografía pos-cuántica basada en retículos

Objetivos específicos:

- Estudio de alternativas para la Transformada de Teoría de Números (Transformada Discreta de Fourier aplicada a cuerpos finitos)
- Determinación de parámetros interesantes para criptografía basada en reticulados en las variantes estudiadas

Otro objetivo de interés especial

Línea 2.iii: Criptografía pos-cuántica basada en retículos

Objetivos específicos:

- Estudio de alternativas para la Transformada de Teoría de Números (Transformada Discreta de Fourier aplicada a cuerpos finitos)
- Determinación de parámetros interesantes para criptografía basada en reticulados en las variantes estudiadas

En detalles:

- Ajustar los parámetros de criptografía basada en reticulados para votaciones electrónicas