Index calculus in class groups of non-hyperelliptic curves of genus 3 from a full cost perspective

Claus Diem

University of Leipzig

Introduction

The discrete logarithm problem (DLP) in degree 0 class groups (Picard groups, Jacobian groups) of hyperelliptic curves (including elliptic curves) over finite fields is a well-estabished cryptographic primitive.

Introduction

The discrete logarithm problem (DLP) in degree 0 class groups (Picard groups, Jacobian groups) of hyperelliptic curves (including elliptic curves) over finite fields is a well-estabished cryptographic primitive.

Additionally, the DLP in degree 0 class groups of *non-hyperelliptic* curves of genus 3 has received considerable attention in the last years.

Articles

- Basiri, Enge, Faugère, Gürel. The arithmetic of Jacobian groups of superelliptic cubics. Math. Comp. (2005)

- Basiri, Enge, Faugère, Gürel. Implementing the Arithmetic of $C_{3,4}$-curves. ANTS VI (2004)

- Flon, Oyono. Fast arithmetic on Jacobians of Picard curves. PKC (2004)

- Koike, Weng. Construction of CM-Picard curves. Math. Comp. (2004)

- Bauer, Teske, Weng. Point Counting on Picard Curves in Large Characteristic. Math. Comp. (2006)

Generic approach

We consider the DLP in degree 0 class groups $\mathrm{Cl}^0(\mathcal{C})$ of non-hyperelliptic genus 3 curves $\mathcal{C}$ finite fields $\mathbb{F}_q$.

Recall: For $q \longrightarrow \infty$ the group $\mathrm{Cl}^0(\mathcal{C})$ has $\sim q^3$ elements.

Generic approach

We consider the DLP in degree 0 class groups $\mathrm{Cl}^0(\mathcal{C})$ of non-hyperelliptic genus 3 curves $\mathcal{C}$ finite fields $\mathbb{F}_q$.

Recall: For $q \longrightarrow \infty$ the group $\mathrm{Cl}^0(\mathcal{C})$ has $\sim q^3$ elements.

This means: If the group order is (nearly) prime, *generic methods* have a running time of $\Theta(q^{3/2})$ field operations.

Generic approach

We consider the DLP in degree 0 class groups $\mathrm{Cl}^0(\mathcal{C})$ of non-hyperelliptic genus 3 curves $\mathcal{C}$ finite fields $\mathbb{F}_q$.

Recall: For $q \longrightarrow \infty$ the group $\mathrm{Cl}^0(\mathcal{C})$ has $\sim q^3$ elements.

This means: If the group order is (nearly) prime, *generic methods* have a running time of $\Theta(q^{3/2})$ field operations.

With an index calculus approach, one can however obtain the following heuristic result:

New algorithm

Heuristic result *One can solve "essentially all" instances of the DLP in degree 0 class groups of non-hyperelliptic curves of genus 3 over finite fields* $\mathbb{F}_q$ *in an expected time of* $\tilde{O}(q)$.

New algorithm

Heuristic result *One can solve "essentially all" instances of the DLP in degree 0 class groups of non-hyperelliptic curves of genus 3 over finite fields $\mathbb{F}_q$ in an expected time of $\tilde{O}(q)$.*

This result relies on the fact that any non-hyperelliptic curve of genus 3 can be given as a *plane quartic*, i.e. it can be defined by an equation

$$F(X, Y, Z) = 0 \ ,$$

where F is homogeneous of degree 4.

New algorithm

The result is a special case of the result in

C. Diem: An Index Calculus Algorithm for Plane Curves of Small Degree (ANTS VII).

It is studied in detail in a forthcoming article with E.Thomé.

New algorithm

Problem The algorithm has a huge storage requirement of $\Theta(q)$ field elements.

Thus a naïve application of the algorithm gives a full cost of $\tilde{O}(q^2)$.

New algorithm

Problem The algorithm has a huge storage requirement of $\Theta(q)$ field elements.

Thus a naïve application of the algorithm gives a full cost of $\tilde{O}(q^2)$.

Questions

- Can one reduce the storage requirements?
- Can one use a mesh-based architecture to reduce the running time?

The idea of index calculus

Let $\mathcal{C}/\mathbb{F}_q, a, b \in \mathrm{Cl}^0(\mathcal{C})$ with $b \in \langle a \rangle$. Let $\ell := \mathbb{Z}/\ell\mathbb{Z}$.

The basic index calculus approach:

1. Fix a *factor base* $\mathcal{F} \subset \mathcal{C}(\mathbb{F}_q)$.
2. Find relations of the form $\sum_j r_{i,j} F_j = \alpha_i a + \beta_i b$, store $(r_{i,j})_j$ as the i^{th} row of a sparse matrix R.
3. Find a non-trivial vector v over $\mathbb{Z}/\ell\mathbb{Z}$ with $vR = 0$.
4. If $\sum_i v_i \beta_i \in (\mathbb{Z}/\ell\mathbb{Z})^*$, then

$$x = -\frac{\sum_i v_i \alpha_i}{\sum_i v_i \beta_i}$$

is the solution to the DLP.

The algorithm

The algorithm uses a *double large prime variation*.

This means:

Let $\mathcal{L} := \mathcal{C}(\mathbb{F}_q) - \mathcal{F}$ be the set of *large primes*. Now relations with up to two large primes are considered. These are stored in a so-called *graph of large prime relations*. This is a graph on the set $\mathcal{L} \,\dot{\cup}\, \{*\}$.

The algorithm

The algorithm uses a *double large prime variation*.

This means:

Let $\mathcal{L} := \mathcal{C}(\mathbb{F}_q) - \mathcal{F}$ be the set of *large primes*. Now relations with up to two large primes are considered. These are stored in a so-called *graph of large prime relations*. This is a graph on the set $\mathcal{L} \,\dot{\cup}\, \{*\}$.

Relations with one large primes P are stored as labeled edges between $*$ and P.

Relations with two large primes P, Q are stored as labeled edges between P and Q.

Relation generation

There are two obvious ways to generate relations.

1. Let $P_0 \in \mathcal{C}(\mathbb{F}_q)$ be fixed. Choose α, β independently uniformly at random in $\mathbb{Z}/\ell\mathbb{Z}$, compute an effective divisor D of degree ≤ 3 with

$$[D] - \deg(D) \cdot [P_0] = \alpha a + \beta b\,.$$

Relation generation

There are two obvious ways to generate relations.

1. Let $P_0 \in \mathcal{C}(\mathbb{F}_q)$ be fixed. Choose α, β independently uniformly at random in $\mathbb{Z}/\ell\mathbb{Z}$, compute an effective divisor D of degree ≤ 3 with

$$[D] - \deg(D) \cdot [P_0] = \alpha a + \beta b\,.$$

2. Let D_∞ be the intersection of $\mathcal{C}$ with the line given by $Z = 0$ ($\deg(D_\infty) = 4$). Compute the line L through two elements F_i, F_j of the factor base, let $F_i + F_j + D_{i,j}$ be the intersection divisor of L and $\mathcal{C}$ on $\mathcal{C}$. Then

$$[F_i] + [F_j] + [D_{i,j}] - [D_\infty] = 0\ .$$

Relation generation

To analyze the second approach, we have this proposition:

Proposition *Let us choose $P_1, P_2 \in \mathcal{C}(\mathbb{F}_q)$ uniformly at random. Then the probability that the intersection divisor of the line through P_1 and P_2 is completely split is $\sim 1/2$.*

The following approach is taken in the work for ANTS in order to minimize the heuristic assumptions.

Relation generation

1. Choose a factor base $\mathcal{F}$ of size $\lceil 2 \cdot \sqrt{q} \rceil$ uniformly at random from the set of all subsets of $\mathcal{C}(\mathbb{F}_q)$.
2. Construct a graph of large prime relations as follows:

   For $i < j$ do

   Compute the divisor $D_{i,j}$.
   If $D_{i,j}$ splits, insert the corresponding edge in the graph of large prime relations.
3. Construct a shortest-path tree with root $*$ in the graph.

Relation generation

4. Generate relations of the form

$$[D] - \deg(D) \cdot [P_0] = \alpha a + \beta b \ ,$$

where $\alpha, \beta \in \mathbb{Z}/\ell\mathbb{Z}$ are chosen uniformly at random and $\deg(D) \leq 3$.

If D splits into elements of the factor base or vertices of the tree, use the tree to substitute the vertices of the tree involved by sums of elements of the factor base.

Why?

The analysis of the algorithm relies on a heuristic comparison of the graph constructed with a *binomial random graph* in which each edge apears (independently of the other edges) with a certain probability.

After an appropriate graph / tree has been constructed, the analysis relies on no further heuristic assumptions.

Reduction of the storage requirements

We modify the algorithm:

1. We do not construct the full graph of large prime relations but only a tree.

Reduction of the storage requirements

We modify the algorithm:

1. We do not construct the full graph of large prime relations but only a tree.
2. We start off by searching for a multiple αa of a which is represented by a completely split divisor. We do the same for a multiple βb of b. Then we fix the factor base, thereby inserting the elements needed to express αa and βb. Afterwards we only generate relations of the form
$$[F_i] + [F_j] + [D_{i,j}] - [D_\infty] = 0 \ .$$

Reduction of the storage requirements

We modify the algorithm:

1. We do not construct the full graph of large prime relations but only a tree.
2. We start off by searching for a multiple αa of a which is represented by a completely split divisor. We do the same for a multiple βb of b. Then we fix the factor base, thereby inserting the elements needed to express αa and βb. Afterwards we only generate relations of the form

$$[F_i] + [F_j] + [D_{i,j}] - [D_\infty] = 0 \ .$$

Because the construction of the tree is less efficient than the construction of the full graph, we fix a factor base of size $\lceil \log(q) \cdot \sqrt{q} \rceil$.

Reduction of the storage requirements

Now a tree of size $q^{3/4}$ suffices:

We need $\#\mathcal{F} \approx \log(q) \cdot q^{1/2}$ divisors $D_{i,j}$ which split into vertices of the tree or elements of the factor base.

Heuristically, the probability that a divisor $D_{i,j}$ splits into vertices of the tree is $\sim 1/2 \cdot (\frac{q^{3/4}}{q})^2 = 1/2 \cdot q^{-1/2}$.

Reduction of the storage requirements

Now a tree of size $q^{3/4}$ suffices:

We need $\#\mathcal{F} \approx \log(q) \cdot q^{1/2}$ divisors $D_{i,j}$ which split into vertices of the tree or elements of the factor base.

Heuristically, the probability that a divisor $D_{i,j}$ splits into vertices of the tree is $\sim 1/2 \cdot (\frac{q^{3/4}}{q})^2 = 1/2 \cdot q^{-1/2}$.

$\Longrightarrow$ We need about $2 \cdot q^{1/2}$ tries to produce one relation which splits into vertices of the tree.

$\Longrightarrow$ Heuristically, after $O(\log(q) \cdot q)$ tries we have enough relations over the factor base.

Reduction of the storage requirements

Heuristic Result *One can solve "essentially all" instances of the DLP in class groups of non-hyperelliptic curves of genus 3 in an expected time of* $\tilde{O}(q)$*, with a storage requirement of* $\Theta(q^{3/4})$ *field elements.*

Parallelization

We use a 3-dimensional mesh based architecture.

For the relation geneartion we need $q^{3/4}$ nodes.

The linear algebra can be performed on a mesh with $\tilde{O}(q^{1/2})$ nodes.

Final heuristic results

- One can perform the *relation generation* on a 3-dimensional mesh of size $q^{3/4}$ in a time of $\tilde{O}(q^{1/2})$, leading to a full cost of $\tilde{O}(q^{5/4})$.
- One can perform the *linear algebra* on a 3-dimensional mesh of size $\tilde{O}(q^{1/2})$ in a time of $\tilde{O}(q^{2/3})$, leading to a full cost of $\tilde{O}(q^{7/6})$.

Final heuristic results

- One can perform the *relation generation* on a 3-dimensional mesh of size $q^{3/4}$ in a time of $\tilde{O}(q^{1/2})$, leading to a full cost of $\tilde{O}(q^{5/4})$.
- One can perform the *linear algebra* on a 3-dimensional mesh of size $\tilde{O}(q^{1/2})$ in a time of $\tilde{O}(q^{2/3})$, leading to a full cost of $\tilde{O}(q^{7/6})$.
- One can solve the DLP with a full cost of $\tilde{O}(q^{17/12})$.

Final heuristic results

- One can perform the *relation generation* on a 3-dimensional mesh of size $q^{3/4}$ in a time of $\tilde{O}(q^{1/2})$, leading to a full cost of $\tilde{O}(q^{5/4})$.
- One can perform the *linear algebra* on a 3-dimensional mesh of size $\tilde{O}(q^{1/2})$ in a time of $\tilde{O}(q^{2/3})$, leading to a full cost of $\tilde{O}(q^{7/6})$.
- One can solve the DLP with a full cost of $\tilde{O}(q^{17/12})$.
- One can solve $q^{1/6}$ instances of the DLP in different groups with a full cost of $\tilde{O}(q^{17/12})$.
- One can solve $q^{1/4}$ instances of the DLP in different groups with a full cost of $\tilde{O}(q^{3/2})$.

On the relation generation

We generate $q^{3/4}$ relations in parallel, one on each node. The elements of the tree are stored on the nodes too. They are accessed via a hash function.

Generating the $q^{3/4}$ relations and inserting the edges into the tree takes a time of $\tilde{O}(q^{1/4})$.

We have to do this $\tilde{O}(q^{1-3/4}) = \tilde{O}(q^{1/4})$ times.

$\Longrightarrow$ Heuristically, the running time is $\tilde{O}(q^{1/2})$.

On the relation generation

We have to guarantee that the growth of the tree is not slowed down by the parallelized construction of the tree.

Heuristically this is the case:

It can be proven:

Let us consider the growth of a “random tree” on $\mathcal{L} \,\dot{\cup}\, \{*\}$ which is obtained by uniformly und independently selecting potential new edges between any two vertices.

Then the growth of the tree is only slowed down by a logarithmic factor by choosing $q^{3/4}$ edges in parallel.

(The proof is very technical.)

Related result

Let us consider the DLP in degree 0 class groups of (nearly) prime order of curves of a fixed genus $g \geq 4$ over finite fields.

Related result

Let us consider the DLP in degree 0 class groups of (nearly) prime order of curves of a fixed genus $g \geq 4$ over finite fields.

Then we have the following heuristic result:

On a 2-dimensional (!) mesh, one can solve the DLP in such groups with a full cost which is asymptotically smaller than the full cost of generic methods.

For example: For $g = 4$, one can solve the DLP with a full cost of $\tilde{O}(q^{31/16})$.

A question

Is there a *formal* model of “mesh” which is suited for a full cost analysis?