Isogeny-based Cryptography School **Summer 2021**

Lecture 6: The Hafner-McCurley Class Group Algorithm

Lecturer: Jean-François Biasse *TA: R. Erukulangara and W. Youmans*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

The goal of this lecture is to prove the following theorem due to Hafner and McCurley [3].

Theorem. *Under the Extended Riemann Hypothesis, there is a Las Vegas algorithm for computing the ideal class group of an imaginary quadratic order.*

In the following, we identify ideal classes with reduced quadratic forms of discriminant $-d$ for $d > 0$ such that $-d$ is a quadratic discriminant. Given two forms f_1, f_2, the operation $f_1 f_2$ is the composition and reduction of the result, thus giving the ideal class represented by the product of the ideal classes represented by f_1, f_2.

6.1 Overview

Let us use f_i to denote the equivalence class $[(p_i, b_i, .))]$. These equivalence classes are called as prime forms. We define the subexponential function L by

$$L(d) := e^{\sqrt{\log d \log\log d}}.$$

Let $n = L(d)^{z+o(1)}$ for a fixed positive number z. Then the classes $[(p_i, b_i, .)]$, $1 \leq i \leq n$ generate the class group $\mathrm{Cl}(-d)$ from Bach's bounds. we can define a homomorphism $\varphi : \mathbb{Z}^n \to \mathrm{Cl}(-d)$ by

$$\varphi(x_1, ..., x_n) = \prod_{i=1}^{n} f_i^{x_i}$$

An integer relation on $f_1, ..., f_n$ is the vector $(x_1, ..., x_n) \in \mathbb{Z}^n$ such that $\varphi(x_1, ..., x_n) = \prod_{i=1}^{n} f_i^{x_i} = 1_{\mathrm{Cl}(-d)}$ where $1_{\mathrm{Cl}(-d)}$ is the identity element of the class group $C(-d)$. Relations in $f_1, ..., f_n$ form an additive subgroup of $\mathbb{Z}^n$ (i.e. a Euclidean lattice) which we denote as Λ. Since φ is surjective, we have

$$\mathbb{Z}^n / \Lambda \cong \mathrm{Cl}(-d).$$

Therefore, the computation of $\mathrm{Cl}(-d)$ reduces to the search for relations between the $f_1, \ldots, f_n$. Once enough relations are collected to generate Λ, a polynomial time linear algebra phase yields the quotient $\mathbb{Z}^n/\Lambda$, and therefore the ideal class group $\mathrm{Cl}(-d)$.

The subexponential algorithm of Hafner and McCurley consists in the choice of a factor basis $f_1, \ldots, f_n$, and the resolution of the following two main tasks

- Finding a generating set of elements of Λ, the lattice of relations between factor basis elements.
- Computing the quotient $\mathbb{Z}^n/\Lambda$.

The quotient computation is well understood, and essentially corresponds to the computation of the Smith Normal Form (SNF) of the matrix representing a basis for Λ.

Making a formal case for the run time without the heuristic that elements sampled in Λ behave randomly demands a little bit of care. We need 3 different phases.

- **Phase 1**: For each $k = 1, \dots, n$, we compute a relation whose k-th coefficient is significantly larger than the others. This ensures the fact that at the end of the collection of the first n relations, the lattice Λ_0 they generate has full rank.
- **Phase 2**: We construct additional relations in order to ensure that at the end of this phase, the lattice Λ_1 they generate satisfies $\det(\Lambda_1) \in 2^{O(\log^4 d)}$.
- **Phase 3**: Once we have $\det(\Lambda_1) \in 2^{O(\log^4 d)}$, we use an expensive randomization process to find the few extra relations needed to generate Λ.

6.2 Phase 1

In this section, we show how to create n linearly independent relations between the $(f_i)_{i \leq n}$. We ensure that the matrix $(a_{i,j})$ whose rows are the relation vectors satisfies $|a_{ii}| > \sum_{j \neq i} |a_{i,j}|$, which in turns guarantees that the matrix $(a_{i,j})$ has full rank. To reduce the run time of the relation search, we use the fact that the Cayley graph of $\mathrm{Cl}(-d)$ is an expander graph. Let n_0 be such that

$$f_1, \dots, f_{n_0} = \left\{ \text{Prime forms corresponding to } p \leq \log^{2+\varepsilon}(d) \text{ and } \left(\frac{d}{p}\right) \neq 1 \right\}.$$

Choosing $t = \log(d) \gg C \frac{\log|\mathrm{Cl}(-d)|}{\log\log d}$, we draw random vectors $\vec{x}$ of ℓ_1-norm t until $f \cdot \left(\prod_{i \leq n_0} f_i^{x_i}\right)$ factors as a product of elements of $\mathcal{B}$ (i.e. is $\mathcal{B}$-smooth).

Proposition 6.1. *Under the ERH, there is a Las Vegas algorithm that takes as input a reduced quadratic form f, and returns $\vec{x} \in \mathbb{Z}^n$ of ℓ_1-norm bounded by $2\log d$ such that $f = \prod_{i \leq n} f_i^{x_i}$ in time*

$$L(d)^{1/4z+o(1)} + L(d)^{z+o(1)}.$$

Its probability of success is at least $1 - \frac{1}{d^{1+o(1)}}$.

Proof. Each attempt at drawing $\vec{y} \in \mathbb{Z}^{n_0}$ of ℓ_1-norm $\log d$ such that $\prod_{i \leq n_0} \cdot f$ is $\mathcal{B}$ can be viewed as a random walk in the Cayley graph of $\mathrm{Cl}(-d)$ of length $\log d$. It has a probability at least $\frac{|S|}{2|\mathrm{Cl}(-d)|}$ of landing in a subset $S \subseteq \mathrm{Cl}(-d)$. We choose S to be the classes corresponding to the smooth reduced quadratic forms. For this, it was shown by Seyssen [5] that the probability is at least $1/L(d)^{1/4z+o(1)}$. This means that we can repeat this experiment $L(d)^{1/4z+o(1)}$ times to have a probability $1 - \frac{1}{d^{1+o(1)}}$ of success.

Now rather than testing the smoothness of each of the $L(d)^{1/4z+o(1)}$ reduced forms we collect, we run Bernstein's batch smoothness test [1] only once on the whole set, which has a run time of $L(d)^{1/4z+o(1)} + L(d)^{z+o(1)}$.

Finally, once a suitable $\vec{y}$ such that $f \cdot \prod_i f_i^{y_i} = f'$ for a reduced smooth form f' is found, we decompose $f' = \prod_i f_i^{z_i}$, and we obtain the relation $\prod_i f_i^{z_i - y_i} = 1_{\mathrm{Cl}(-d)}$. Since the norm of f' is less than $\sqrt{d}$, we have that the ℓ_1-norm of $\vec{z}$ is less than $\log d$ and thus the ℓ_1 norm of $\vec{x} := \vec{z} - \vec{y}$ is less than $2\log d$. $\square$

Given the above building block (which will be reused in subsequent phases, and even for applications in future lectures such as DLP, ideal decomposition etc ...), we can easily compute a full rank matrix of relations by choosing $f = f_i^{2nd}$ for each of the n elements $f_i \in \mathcal{B}$. This ensures that the i-th row $(a_{i,j})_{j\leq n}$ has a dominant i-th coefficient as requested.

Proposition 6.2. *Under the ERH, there is an algorithm that outputs n linearly independent relations between elements of $\mathcal{B}$ with probability at least $1 - \frac{1}{d^{1+o(1)}}$ in time*

$$L(d)^{z+o(1)}\left(L(d)^{z+o(1)} + L(d)^{1/4z+o(1)}\right).$$

6.3 Phase 2

At the end of Phase 1, we have a sublattice $\Lambda_0 \subseteq \Lambda$ of full rank with $\det(\Lambda_0) < n^{5n/2}d^n$ by Hadamard bound. Then we add new relations hoping that they do not belong to the previous sublattice of relations. Starting with $\Lambda_1 = \Lambda_0$, each time we find $\vec{x} \notin \Lambda_1$, and update Λ_1 by doing

$$\Lambda_1 \leftarrow \Lambda_1 + \mathbb{Z}\vec{x},$$

the determinant of Λ_1 gets divided by at least a factor 2.

To create relations, we first draw a vector $\vec{y}$ uniformly at random in $W_n(d^2)$ for

$$W_n(t) := \{x : x \in \mathbb{Z}^n, \|x\|_\infty \leq t\},$$

and we compute the reduced form $f = \prod_i f_i^{y_i}$. Then we use Proposition 6.1 to create $\vec{x} \in \mathbb{Z}^n$ such that $f = \prod_i f_i^{x_i}$. Then we have that $\vec{y} - \vec{x} \in \Lambda$ is a relation. The question is "does it belong to Λ_1?". For $\vec{y} - \vec{x}$ to be outside of Λ_1, it suffices that the random vector $\vec{y}$ drawn from $W_n(d^2)$ be sufficiently far from Λ_1. Indeed, we know that the ℓ_1-norm of $\vec{x}$ is less than $2\log d$, therefore it suffices to draw $\vec{y}$ at distance $2\log d + \varepsilon$ from Λ_1 to guarantee that the resulting relation is not in Λ_1. In other words, we need to draw $\vec{y}$ in $W_n(d^2) \setminus V$ for

$$V := \left\{\bigcup B(x, (2+\varepsilon)\log d) : x \in \Lambda_1\right\},$$

where $B(x, r)$ denotes the n-dimensional sphere of radius r for the Euclidean distance, centered at x.

To evaluate the odds of drawing $\vec{y}$ outside of V, we compute an upper bound on the number of integer points in V. We need to answer two questions:

- How many lattice points of Λ_1 are there in $W_n(d^2)$?
- What is an upper bound on the number of integer vectors in each $B(x, (2+\varepsilon)\log d)$?

The first question is answered by [3, Lem. 1] which implies that

$$\left|\Lambda_1 \cap W_n(d^2)\right| \in \frac{(2d^2)^n}{\det(\Lambda_1)} \cdot \left(1 + O\left(\frac{n^3}{d}\right)\right).$$

Then, according to [4, Corollary 1.4], the number of integer elements contained inside each individual n-dimensional sphere is bounded from above by $\frac{3.e^{\pi.k^2.r^2}}{2}$, where r is the radius of the sphere and $k = 10(\log n + 2)$. Choosing $r = (2+\varepsilon)\log d$ yields

$$\left|W_n(d^2) \setminus V\right| \geq (2d^2)^n - \frac{(2d^2)^n}{e^{\log^4 d(1+o(1))}}\left(1 + o(1)\right)$$

whenever $\det(\Lambda_1) \geq e^{log^4 d(1+o(1))}$. In these conditions, the probability of drawing $\vec{y}$ in $W_n(d^2) \setminus V$ is at least $1 + \frac{1}{e^{\log^4 d(1+o(1))}}$.

We repeat this process $\log_2 \left(n^{5n/2} d^n\right) = n^{1+o(1)}$ times to ensure that even with the pessimistic estimates on $\det(\Lambda_0)$ and even if we decrease the determinant by a factor 2 at a time, we end up with $\det(\Lambda_1) < e^{\log^4 d}$ at the end of the phase. We compute the determinant of Λ_1 in time $n^{3+o(1)}$ by reducing it with [6, Th. 58] to the case of a square matrix and then by using the Smith Normal Form algorithm for square nonsingular matrices of [2].

Proposition 6.3. *Under the ERH, Phase 2 produces a sublattice* Λ_1 *such that* $\det(\Lambda_1) < e^{\log^4 d}$ *with probability at least* $1 - \frac{1}{d^{1+o(1)}}$ *in time*

$$L(d)^{3z+o(1)} + L(d)^{z+1/4z+o(1)}.$$

Proof. The cost of each relation is that of the computation of f, which is in $L(d)^{z+o(1)}$, and of the execution of the relation search of Proposition 6.1 which is in $L(d)^{1/4z+o(1)} + L(d)^{z+o(1)}$. This is repeated $L(d)^{z+o(1)}$ times for a total cost of $L(d)^{z+1/4z+o(1)} + L(d)^{2z+o(1)}$. Then the final determinant computation runs in time $L(d)^{3z+o(1)}$ thus proving our statement on the run time.

The probability of success is at least that of succeeding $n^{1+o(1)}$ times at creating a relation outside of Λ that satisfies $\det(\Lambda_1) \geq e^{\log^4 d}$. This means it is at least

$$\left(1 - \frac{1}{e^{\log^4 d(1+o(1))}}\right)^{n^{1+o(1)}} \cdot \left(1 - \frac{1}{d^{1+o(1)}}\right)^{n^{1+o(1)}} = 1 - \frac{1}{d^{1+o(1)}}.$$

□

6.4 Phase 3

At the beginning of Phase 3, we have $\Lambda_1 \subseteq \Lambda$ with $|\Lambda/\Lambda| < e^{\log^4 d}$. Then we create a tower of sublattices $(\Lambda_i)_{2\leq i\leq m}$ of the lattice of relations such that

$$\Lambda_0 \subseteq \Lambda_1 \subseteq \ldots, \subseteq \Lambda_m = \Lambda.$$

The key observation proved in [3, Lem. 2] is that when relations are obtained simply by testing the $\mathcal{B}$-smoothness of elements $f = \prod_i f_i^{x_i}$ for a vector $\vec{x}$ drawn uniformly at random in $W_d(d^2)$, the probability that they belong to a given coset in Λ/Λ_1 is essentially given by $\det(\Lambda)/\det(\Lambda_1)$. This means that new relations have somewhat comparable chances of landing in different cosets of Λ/Λ_1. Once every cost has been hit at least once, the relation collection is complete. We can show that only $\log^4 d(1 + o(1))$ such steps are required to generate the whole lattice Λ with good enough probability. This point is important because unlike in Phase 2, the cost of finding each individual relation is $L(d)^{z+1/4z+o(1)}$ due to the fact that we recompute a new f for each form tested for smoothness. At the end, we use the Smith Normal Form algorithm described in the previous section that relies on [2, 6] to produce $d_1, \ldots, d_n$ such that

$$\mathrm{Cl}(-d) = \mathbb{Z}/d_1\mathbb{Z} \oplus \mathbb{Z}/d_2\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/d_n\mathbb{Z}.$$

Theorem 6.4. *Under the ERH, there is a Las Vegas algorithm to compute* $\mathrm{Cl}(-d)$ *in time* $L(d)^{3/\sqrt{8}+o(1)}$ *with probability at least* $1 - \frac{1}{d^{1+o(1)}}$.

Proof. As mentioned above, each relation obtained by drawing $\vec{x} \in W_n(d^2)$ and test it for smoothness (using [1]) takes time $L(d)^{z+1/4z+o(1)}$. We can also show using [3, Lem. 2] that the probability that the resulting relation is in any given coset of Λ/Λ_1 is in $\frac{\det(\Lambda)}{\det(\Lambda_1)}(1+o(1))$. Then, as observed in [3, Lem. 2] we generate all of Λ with probability at least $1-\frac{1}{d}$ after collecting m relations where m satisfies

$$m \geq \frac{\log|\Lambda/\Lambda_1| + \log d}{\log(2/\alpha)}$$

for some $\alpha = 1 + O\left(\frac{n^3}{d}\right)$. This means that a polynomial number of relations is required, which proves that the run time is again

$$L(d)^{3z+o(1)} + L(d)^{z+1/4z+o(1)}.$$

This value is minimized for $z = 1/\sqrt{8}$, which yields a total run time of $L(d)^{3/\sqrt{8}+o(1)}$. □

References

[1] D. Bernstein. How to find smooth parts of integers. submited to *Mathematics of Computation*.

[2] A. Birmpilis, G. Labahn, and A. Storjohann. A Las Vegas algorithm for computing the smith form of a nonsingular integer matrix. In I. Emiris and L. Zhi, editors, *ISSAC '20: International Symposium on Symbolic and Algebraic Computation, Kalamata, Greece, July 20-23, 2020*, pages 38–45. ACM, 2020.

[3] J.L. Hafner and K.S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2:837–850, 1989.

[4] O. Regev and N. Stephens-Davidowitz. A reverse minkowski theorem. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 941–953, New York, NY, USA, 2017. Association for Computing Machinery.

[5] M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Mathematics of Computation*, 48:757–780, 1987.

[6] A. Storjohann. The shifted number system for fast linear algebra on integer matrices. *J. Complex.*, 21(4):609–650, 2005.