J. Math. Cryptol. **?** (**????**), 1–46
DOI 10.1515/jmc-**????**-**???**

On ideal lattices in multicubic fields

Andrea Lesavourey, Thomas Plantard and Willy Susilo

Communicated by ???

Abstract. One family of candidates to build a post-quantum cryptosystem upon relies on euclidean lattices. In order to make such cryptosystems more efficient, one can consider special lattices with an additional algebraic structure such as ideal lattices. Ideal lattices can be seen as ideals in a number field. However recent progress in both quantum and classical computing showed that such cryptosystems can be cryptanalysed efficiently over some number fields. It is therefore important to study the security of such cryptosystems for other number fields in order to have a better understanding of the complexity of the underlying mathematical problems. We study in this paper the case of multicubic fields.

Keywords. Public-key cryptography, Post-quantum cryptography, Number Fields, Ideal lattice, Cryptanalysis, Unit Group, Cubic Field.

1 Introduction

Given a number field K an ideal lattice over K is simply an ideal I of $\mathcal{O}_K$ considered as a $\mathbb{Z}-$module in $\mathbb{R}^n$. It can be represented by an integral basis. In the simplest version of encryption using ideal lattices we can consider a number field K and $I = g\mathcal{O}_K$ a principal ideal with a short g when I is considered as a lattice. Then we have :

- **Public :** K and I
- **Private :** g

The security of the cryptosystem relies on the hardness of finding g or another short generator. Finding a generator is called the Principal Ideal Problem (PIP) and is referred as one of the main tasks of Computational Number Theory by Cohen in [10]. Finding a short generator is referred as the Short Principal Ideal Problem (SPIP). The first advantage of such a system compared to a general lattice based system is that instead of storing a n^2 matrix to designate the lattice we can use a more compact representation. We therefore need less space to store the public and private keys. Moreover the algebraic structure of the fields we are working with allows faster computations. Because of this efficiency, ideal lattices – and

more generally structured lattices – are under a lot of investigation to evaluate the security of lattice-based cryptosystems.

By default an attack to recover the generator g is done in two steps

(i) recover a generator h of I;

(ii) find a short generator given h.

The first step corresponds to the PIP which is considered a hard problem in classical computational number theory. However it is shown that it can be efficiently done by using quantum computing as in [6]. The second is a reduction phase which is the kind of tasks that seem difficult even for quantum computers. In order to solve it, one may use the structure of the set of generators of I and the Log-unit lattice. This strategy was mentioned in [9] where it was claimed that in the case of cyclotomic fields the group of cyclotomic units has a good enough geometry in the Log-unit lattice to help recovering a short secret vector. A proper analysis of this situation has been done in [12] where the authors gave a bound for the norm of the vectors of the dual basis. In [4] the authors studied another family of fields, namely the multiquadratic fields, and were able to recover a short generator of an ideal in classical polynomial time for a wide range of fields.

Results : In this paper we study the case of multicubic fields i.e. fields generated by cube roots of integers. We proved that their algebraic structure is similar to the one of multiquadratic fields so that the framework of the attack of [4] can be adapted to multicubic fields. We are able to compute units of degree 3^n number fields for n up to 5. We were able to conduct experiments on the PIP to find an success rate similar to the ones presented in [4].

Future work : Further work can consist in improving the results on multicubic fields and generalise the approach to number fields generated by p-root of integers for bigger primes p. This could lead to a better understanding on what can be done regarding ideal lattices. Moreover it would be interesting to work on other important tasks of computational number theory over these fields such as computing the class group. Another direction would be to study number fields with more complicated structures in order to look whether we can again find a good basis for the Log-unit lattice or not.

2 Background

Notations : The inner product is denoted by $(\cdot \mid \cdot)$. When we consider a tuple $(\lambda_1, \ldots, \lambda_n)$ we can designate it by $\underline{\lambda}$. An interval in the integers will be noted $[\![a, b]\!]$. Matrices are written between brackets and not parenthesis.

Lattices :

A *lattice* is a discrete subgroup of $\mathbb{R}^n$ where n is a positive integer. A *basis* of a lattice $\mathcal{L}$ is a basis of $\mathcal{L}$ when considered as a $\mathbb{Z}$-module. One way of representing a lattice is then to consider the matrix of a basis of the lattice.

Let us note $\lambda_1(\mathcal{L})$ the norm of the shortest non zero vector of $\mathcal{L}$ i.e.

$$\lambda_1(\mathcal{L}) = \min\{\|u\| \mid u \in \mathcal{L} \setminus \{0\}\}.$$

(i) *Shortest Vector Problem (SVP)* : «Given a a lattice $\mathcal{L}$ of dimension n, find $u \in \mathcal{L} \setminus \{0\}$ such that $\|u\| = \lambda_1(\mathcal{L})$ ».

(ii) *Closest Vector Problem (CVP)* : «Given a lattice $\mathcal{L}$ of dimension n and $t \in \mathbb{R}^n$, find $u \in \mathcal{L}$ such that $\forall v \in \mathcal{L}, \|t-u\| \leqslant \|t-v\|$. »

(iii) *Bounded Distance Decoding (BDD)* : «Given a basis B of a lattice $\mathcal{L}$, a target vector t such that $d(t,\mathcal{L}) < \lambda_1(\mathcal{L})/2$, find the lattice vector $v \in \mathcal{L}$ closest to t. ».

In practice we can consider relaxed versions of these problems with respect to an approximation factor. For general lattices these problems are NP-hard thus at least as hard as factorising for example. Moreover we do not have any result showing that quantum computers can solve these problems for general lattices. These problems are easier to solve if we have a good basis at our disposal i.e. a basis built with relatively short vectors which are nearly orthogonal to each other.

For example one of the problems that we want to solve is the BDD. It can be solved using the dual lattice with Babai's round-off algorithm. Consider a lattice $\mathcal{L}$ generated by a basis $B = \{b_1, \ldots, b_n\}$. We call the *dual basis* of B – noted $B^\vee$ – the basis $\{b_1^\vee, \ldots, b_n^\vee\}$ verifying $B^\vee \subset \mathrm{Span}_{\mathbb{R}}(B)$ and $(b_j^\vee \mid b_i) = \delta_{i,j}$. The lattice generated by B is called the *dual lattice of* $\mathcal{L}$ and noted $\mathcal{L}^\vee$. The matrices representing the bases B and $B^\vee$ verify $(B^\vee)^t.B = B^t.B^\vee = I_n$. In order to solve the BDD problem we may use the round-off algorithm which calculates $B \cdot \lfloor (B^\vee)^t \cdot t \rceil$ together with the following lemma.

Lemma. *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice with base B and let $t = v + e$ for some $v \in \mathcal{L}$ and $e \in \mathbb{R}^n$. If we have $|(b_j^\vee \mid e)| \leqslant \frac{1}{2}$ for all j then the round-off algorithm outputs v.*

Despite the hardness of these problems over random lattices, high-dimensional lattices are large objects and slow to handle. A way of copping with that is to work with lattices with extra algebraic structure such as ideal lattices. However this can introduce a security weakness as it may be easier to find good basis related to such lattices.

Number Fields :

A *number field* K is a field which is a finite extension of $\mathbb{Q}$. It can always be described as a polynomial quotient ring

$$\frac{\mathbb{Q}[X]}{(P(X))}$$

where $P(X)$ is irreducible in $\mathbb{Q}[X]$. Equivalently if we choose θ to be any root of $P(X)$ we can see K as $\mathbb{Q}(\theta)$ the smallest field containing $\mathbb{Q}$ and θ. If we note n the degree of $P(X)$ then the dimension of K over $\mathbb{Q}$ – written $\deg_{\mathbb{Q}} K$ or $[K : \mathbb{Q}]$ – is n.

There are n distinct *complex field embeddings* $K \hookrightarrow \mathbb{C}$ usually noted $\sigma_1, \ldots, \sigma_n$. They map θ to the other complex roots of $P(X)$. We will note $\mathrm{Hom}(K, \mathbb{C})$ this set. Among them we have r_1 real embeddings and r_2 pairs of complex embeddings. The two elements of one given pair are conjugates one from each other. It is the usage to note $\sigma_1, \ldots, \sigma_{r_1}$ the real embeddings and to consider that $\sigma_{j+r_2} = \overline{\sigma_j}$ for all $j \in [\![r_1 + 1, r_1 + r_2]\!]$. Given a complex embedding $\sigma \in \mathrm{Hom}(K, \mathbb{C})$ the set $\{x \in K \mid \sigma(x) = x\}$ is a subfield of K. We will note it $\mathrm{Inv}(\sigma)$ or K_σ when there is *no ambiguity*, especially when related to Proposition 3.18 to follow notations used in [4].

The *Galois Group* of a field extension L/K noted $\mathrm{Gal}(L/K)$ is the group of field automorphisms of L which are congruent to the identity when restricted to K. It is a subset of $\mathrm{Hom}(L, \mathbb{C})$. An extension L/K is called a Galois extension when the cardinality of $\mathrm{Gal}(L/K)$ equals the dimension $[L : K]$. Moreover we have the *Galois correspondence* which states that given a Galois extension L/K there is a one-to-one correspondence between the subgroups of $\mathrm{Gal}(L/K)$ and the subfields of L containing K. Given a subgroup H of $\mathrm{Gal}(L/K)$ we will note $\mathrm{Inv}(H)$ the corresponding subfield of L. In the case of a number field K we say it is a Galois field if it is Galois as an extension of $\mathbb{Q}$. For example the cyclotomic fields are Galois number fields as well as the multiquadratic fields considered in [4]. However this property is not verified by a general number field K and we have to consider the Galois closure of K which is in fact the smallest extension containing all the roots of the irreducible polynomial $P(X)$.

One ring of particular importance is the *ring of integers of K noted $\mathcal{O}_K$*. It consists of the elements of K which are roots of a monic polynomial of $\mathbb{Z}[X]$. This ring as well as its ideals are full rank sub-$\mathbb{Z}$-module of K. In fact for a given ideal I one can find a basis $(b_1, \ldots, b_n)$ of elements of $\mathcal{O}_K$ such that $K = \bigoplus_{i=1}^{n} \mathbb{Q} b_i$,

$L = \bigoplus_{i=1}^{n} \mathbb{Z}b_i$ and $I = \bigoplus_{i=1}^{n} \mathbb{Z}d_i b_i$ with $(d_1, \ldots, d_n) \in \mathbb{Z}^n$. We can see that the images of $\mathcal{O}_K$ and of any ideal I of $\mathcal{O}_K$ under the action of any embedding of K into $\mathbb{R}^n$ are lattices. The usual embedding corresponds to view a number field K as a quotient $\frac{\mathbb{Q}[X]}{(f(X))}$. Then every element $g(X) = g_0 + \cdots + g_n X^n$ of K can be seen as the vector with coordinates $(g_0, \ldots, g_n)$ in $\mathbb{R}^n$. The other fundamental example is called the *Minkowski embedding* and is

$$\begin{array}{rccc} \sigma: & K & \longrightarrow & \mathbb{R}^n \\ & x & \longmapsto & (\sigma_i(x))_{i \in [\![1, r_1+r_2]\!]} \end{array}$$

The group of units of $\mathcal{O}_K$ noted $\mathcal{O}_K^\times$ is the set $\left\{u \in \mathcal{O}_K \mid u^{-1} \in \mathcal{O}_K\right\}$. The group of units of $\mathcal{O}_K^\times$ has a specific structure that we can take advantage of.

Given a number field K of degree n with $n = r_1 + 2r_2$ as before we have

$$O_k^\times \simeq \frac{\mathbb{Z}}{m\mathbb{Z}} \times \mathbb{Z}^{r_1+r_2-1}.$$

This isomorphism which allows to see the units of $\mathcal{O}_K^\times$ modulo its torsion group as a lattice is realised by an important embedding which is the *Log-embedding* of K. It is defined as

$$\begin{array}{rccc} \mathrm{Log}_K: & K^* & \longrightarrow & \mathbb{R}^{r_1+r_2-1} \\ & x & \longmapsto & (\log(|\sigma_i(x)|))_{i \in [\![1, r_1+r_2]\!]} \end{array}$$

The set $\mathrm{Log}_K(\mathcal{O}_K^\times)$ is a lattice of the hyper plane orthogonal to the all ones vector. It is called the *Log-unit lattice*. Sometimes we define the Log-embedding by using all of the embeddings σ_i. By doing so the Log-unit lattice is a lattice of rank $r_1 + r_2 - 1$ in $\mathbb{R}^n$.

Ideal lattice cryptosystem :

Recall that ideal based cryptosystems such as presented in [19] have in general a private key which is a short generator of a public ideal I. The security of such cryptosystems relies on the supposed hardness of finding such a generator given an ideal, problem called the *Short Principal Ideal Problem*. The *Principal Ideal Problem* consists in finding any generator of the principal ideal. i.e. given an ideal $I = g\mathcal{O}_K$, find some h such that $I = h\mathcal{O}_K$. As mentioned the process done to solve the SPIP relies essentially in two steps : solve the PIP and then shorten the retrieved generator. The set of generators of a principal $I = g\mathcal{O}_K^\times$ is $\{gu \mid u \in \mathcal{O}_K^\times\}$. Therefore solving the PIP yields $h = gu$ with $u \in \mathcal{O}_K^\times$. It is therefore possible to retrieve g from h by finding u. This is where we can use

the Log-embedding and the Log-unit lattice. If we transpose the situation with the Log-embedding, for every generator h we have

$$\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u).$$

Using that remark and finding the element of the Log-unit lattice closest to h it is possible to retrieve g. This correspond to solve the Closest Vector Problem (CVP) with respect to the target h and the lattice Log $\mathcal{O}_K^\times$, and even the BDD because we know the generator g is short. The success of such a method is therefore dependent from the particular geometry of the Log-unit lattice meaning that we want to have access to a somehow good basis i.e. orthogonal enough. To do this attack we need to be able to

(i) solve the PIP : this is considered hard classically and can be done in quantum polynomial time

(ii) compute $\mathcal{O}_K^\times$: as the PIP this is considered hard classically and can be done in quantum polynomial time

(iii) shorten a generator h by solving the BDD with respect to $\mathrm{Log}_K(\mathcal{O}_K^\times)$: this will depend on the basis obtained.

Cyclotomic fields :

The most used number fields in cryptography are cyclotomic fields. They are number fields generated by a nth root of 1 noted ζ_n. and are used because of their well-known and simple algebraic structure which allows fast computations. However in [12] Cramer and al. showed that it was possible to recover a short generator of a principal I from another generator. In this paper n is assumed to be a prime power p^r. The cyclotomic field $\mathbb{Q}(\zeta_n)$ has degree $p^{r-1}(p-1)$ and is isomorphic to $\frac{\mathbb{Q}[X]}{(\phi_n(X))}$ where $\phi_n(X) = X^{p^{r-1}(p-1)} + \cdots + X^{p^{r-1}} + 1$ is the nth cyclotomic polynomial.
Recall that we want to apply solve the BDD with respect to the Log-unit lattice. In order to do that we need to have access to the unit group $O_K^\times$. Luckily in the special case of cyclotomic fields we know a subgroup very close, the so-called cyclotomic units. Let C designate this subgroup. It is generated by the set $\{\zeta_n\} \cup \{\frac{\zeta_n^j - 1}{\zeta_n - 1} \mid j \in (\frac{\mathbb{Z}}{n\mathbb{Z}})^\times\}$. Very close means that the index $[O_K^\times : C]$ is finite and very small. In a lot of cases it is even 1. Therefore we have access to a sublattice $\mathrm{Log}_K(C)$ close to the full Log-unit lattice. In [12] the authors take advantage of the structure of C and use tools from analytic number theory over cyclotomic fields to give an upper bound for the norm of the dual vectors of $\mathrm{Log}_K(C)$. They can then solve the BDD over this sublattice using the round-off algorithm and since it is

close enough to the full Log-unit lattice it allows them to retrieve a short generator.

Multiquadratic fields :

Multiquadratic fields are fields which are generated by a sequence of square-roots of integers $\sqrt{d_1}, \ldots, \sqrt{d_n}$. In [4] Bauch and al. proved that it is possible to compute the units $\mathcal{O}_K^\times$ and solve the PIP efficiently using only a classical computer. This goes even further than for cyclotomic fields. They use the full unit group to solve the SPIP corresponding to the second part of an attack on an ideal lattice. In order to be able to do all of that they take advantage of the special structure of a multiquadratic field, particularly that it has a lot of subfields which are multiquadratic fields too. As in the cyclotomic case they exhibit a subgroup of the unit group that they call multiquadratic units. We can note it U. This subgroup is generated by the fundamental units of all quadratic subfields. Under the Log-embedding it constitutes a full rank sublattice of $\mathrm{Log}_K(\mathcal{O}_K^\times)$ and the fundamental units of quadratic subfields form an orthogonal basis. This is the best situation possible to solve lattices problem. However even if $[\mathcal{O}_K^\times : U]$ is finite it is too large to be used in the same way as cyclotomic units are. It is however the fundamental stone to build the whole unit group. The algorithms of [4] rely essentially on the Proposition 5.1 which can be stated as

Proposition. *Let K be a multiquadratic field of dimension 2^n. Then for all $x \in K$*

$$x^2 \in K_1 K_2 K_3$$

where K_1, K_2 and K_3 are multiquadratic subfields of K of dimension 2^{n-1}. Moreover if x is a unit then the fields can be replaced by their unit group.

We see that if it is possible to compute the unit group of multiquadratic fields of degree 2^{n-1} then we can compute a subgroup G of $O_K^\times$ such that $(O_K^\times)^2 < G < O_K^\times$. The authors of [4] then prove that we can retrieve $O_K^\times$ from G with high probability. This last step require to compute square roots of element of K. Therefore in order to construct $O_K^\times$ from the units of subfields of K of degree 2^{n-1} we only have to carry out products and square root operations. All of these can be done quickly in K. The algorithm then works recursively. It will compute the fundamental units of all the quadratic subfields using classical algorithms and will build the whole unit group by doing products and square root extractions.

In order to solve the PIP in multiquadratic fields the authors of [4] use again the previous Proposition. If $I = gO_K$ is a principal ideal then $g^2 = g_1 g_2 g_3$ where the

g_i are the generators of the relative norm ideals $N_{K/K_i}(I)$ which are ideals of $\mathcal{O}_{K_i}$ respectively. As before the algorithm works recursively to compute an element h which is a generator of I^2 then use the unit group to retrieve a generator of I.

The last step of the attack is then carried using the Log-unit lattice and using a rounding algorithm. The results of experiments show a high rate of success.

3 Multicubic fields

In this section we will study *multicubic fields* i.e. number fields generated by cube roots of positive integers. Cubic fields have been well studied and one can find several results in textbooks or papers. See for instance [10] and [2]. We still present some facts useful to our presentation. However we could not find papers on multicubic fields dealing with the results we are interested in. We prove that the structure of multicubic fields is similar to multiquadratic fields so that the attack of Bauch and al. can be adapted.

3.1 First structural results

First we will prove several mathematical fact concerning multicubic fields useful for our study. Let us start with a quick lemma on cubic fields that we will use later.

Lemma 3.1. *Consider p and q two natural integers which are not cubes. Then the cubic fields $\mathbb{Q}(\sqrt[3]{p})$ and $\mathbb{Q}(\sqrt[3]{q})$ are equals if, and only if, the following holds : $p = q \times a^3$ or $p = q^2 \times a^3$, with $a \in \mathbb{Q}$.*

Proof. First we will use the following implication $\mathbb{Q}(\sqrt[3]{p}) = \mathbb{Q}(\sqrt[3]{q}) \implies \sqrt[3]{p} \in \mathbb{Q}(\sqrt[3]{q})$. Thus we can write $p^{\frac{1}{3}} = aq^{\frac{2}{3}} + bq^{\frac{1}{3}} + c$ with $(a, b, c) \in \mathbb{Q}^3$. The integer p not being a cube we know that $p^{\frac{1}{3}}$ does not belong in $\mathbb{Q}$. Therefore we have $(a, b) \neq (0, 0)$. Then we can take the cube of the previous equality which gives $p = \left(aq^{\frac{2}{3}} + bq^{\frac{1}{3}} + c\right)^3$ and by developing the right-hand side we obtain

$$p = (3a^2bq+3ac^2+3b^2c)q^{\frac{2}{3}}+(3a^2cq+3ab^2q+3bc^2)q^{\frac{1}{3}}+(a^3q^2+6abcq+b^3q+c^3).$$

Then p is a rational number so we have the following system

$$\begin{cases} 3a^2bq + 3ac^2 + 3b^2c = 0 & (1) \\ 3a^2cq + 3ab^2q + 3bc^2 = 0 & (2) \\ a^3q^2 + 6abcq + b^3q + c^3 = p. & (3) \end{cases}$$

First, let us show that c is equal to zero. We can calculate $b \times (1) - a \times (2)$ which gives

$$3c(b^3 - a^3 q) = 0.$$

Therefore if we suppose $c \neq 0$ we obtain $b^3 - a^3 q = 0$. But $(a, b) \neq (0, 0)$ so both of them are non zero and we can write

$$q = \frac{b^3}{a^3} = \left(\frac{b}{a}\right)^3$$

which is impossible since q is not a cube. The coefficient c is therefore 0. In (1) this gives $3a^2bq = 0$ which implies $a = 0$ or $b = 0$. Now we use (3) to write

$$\left((a = 0) \Longrightarrow (b^3 q = p)\right) \text{ AND } \left((b = 0) \Longrightarrow (p = a^3 q^2)\right).$$

To summarise, if $\mathbb{Q}(\sqrt[3]{p}) \subseteq \mathbb{Q}(\sqrt[3]{q})$ then $p = a^3 q$ or $p = a^3 q^2$ for some rational a. Similarly if $\mathbb{Q}(\sqrt[3]{q}) \subseteq \mathbb{Q}(\sqrt[3]{p})$ then $q = b^3 p$ or $q = b^3 p^2$ for some rational b. Finally since $p = q \times a \iff q = p \times b$ and $p = q^2 \times a \iff q = p^2 \times b$, we obtain the desired result. □

Definition 3.2. Consider n distinct integers $d_1, \ldots, d_n$ which are not rational cubes. We will call multicubic field generated by $d_1, \ldots, d_n$ the following number field :

$$K = \mathbb{Q}\left(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}}\right).$$

Remark that the sequence elements not being cubes forbids $\mathbb{Q}$ to be a multicubic field. We have not supposed anything more about the defining sequence. For example several elements could be equal to each other. However we can always find a minimal sequence whose length will be proved to be equivalent to the dimension of the corresponding multicubic field.

Proposition 3.3. *Every multicubic field* $K = \mathbb{Q}\left(c_1^{\frac{1}{3}}, \ldots, c_m^{\frac{1}{3}}\right)$ *can be defined by a sequence of cube-free integers* $d_1, \ldots, d_n$ *such that for none of the tuples of exponents* $\underline{\alpha} = (\alpha_1, \ldots, \alpha_n) \in [\![0, 2]\!]^n \setminus \{0, \ldots, 0\}$ *the product* $\prod_{i=1}^{n} d_i^{\alpha_i}$ *is a cube.*

Proof. We will proceed by induction on m. If $m = 1$ then there is nothing to prove. Now suppose that the property is true for a fixed integer $m \geqslant 1$ and consider a multicubic Field $K = \mathbb{Q}\left(c_1^{\frac{1}{3}}, \ldots, c_{m+1}^{\frac{1}{3}}\right)$ defined by $m + 1$ integers.

Denote L the multicubic field defined by the first coefficients $c_1, \dots, c_m$. We have $K = L\left(c_{m+1}^{\frac{1}{3}}\right)$ and by hypothesis L can be defined by cube-free integers $d_1, \dots, d_n$ verifying the desired property. First we can assume that c_{m+1} is cube-free. secondly the integers $d_1, \dots, d_n, c_{m+1}$ define K as a multicubic field. If they verify the property then nothing more needs to be done. Suppose now that

$$\prod_{i=1}^{n} d_i^{\alpha_i} \times c_{m+1}^{\alpha} = a^3$$

for some $(\alpha_1, \dots, \alpha_n, \alpha) \in [\![0, 2]\!]^{n+1}$ and $a \in \mathbb{Z}$. By induction hypothesis the product $\prod_{i=1}^{n} d_i^{\alpha_i}$ is not a cube, therefore $\alpha \neq 0$ and we can write

$$c_{m+1}^{\frac{\alpha}{3}} = \frac{a}{\prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}} \in L$$

meaning that we have $K = L$ and that K verifies the desired property. □

Definition 3.4. A sequence of integers defining a multicubic field K will be called reduced if it verifies the property of Proposition 3.3.

Proposition 3.5. *Consider $K = \mathbb{Q}\left(d_1^{\frac{1}{3}}, \dots, d_n^{\frac{1}{3}}\right)$ a multicubic field such that $d_1, \dots d_n$ is reduced. Then K has exactly $\frac{3^n-1}{2}$ cubic subfields of the form*

$$\mathbb{Q}\left(d_1^{\frac{\alpha_1}{3}} \times \cdots \times d_n^{\frac{\alpha_n}{3}}\right)$$

with $\underline{\alpha} = (\alpha_1, \dots, \alpha_n) \in [\![0, 2]\!]^n \setminus \{0\}$. Moreover if we see $\underline{\alpha}$ and $\underline{\beta}$ as elements of $\mathbb{F}_3^n$ we have

$$\mathbb{Q}\left(d_1^{\frac{\alpha_1}{3}} \times \cdots \times d_n^{\frac{\alpha_n}{3}}\right) = \mathbb{Q}\left(d_1^{\frac{\beta_1}{3}} \times \cdots \times d_n^{\frac{\beta_n}{3}}\right) \iff \underline{\alpha} = \underline{\beta} \text{ or } \underline{\alpha} = 2\underline{\beta}$$

Proof. Consider $\underline{\alpha} \in (\mathbb{F}_3)^n \setminus \{0\}$. There is $i \in [\![1, n]\!]$ such that $\alpha \neq 0$. Therefore the product $d_1^{\alpha_1} \times \cdots \times d_n^{\alpha_n}$ is not a cube so $d_1^{\frac{\alpha_1}{3}} \times \cdots \times d_n^{\frac{\alpha_n}{3}}$ is not rational and therefore generates a subfield of K of degree 3 over $\mathbb{Q}$. The subfields of the form considered are then cubic. Now consider two elements $\underline{\alpha}$ and $\underline{\beta}$ such that

$$\mathbb{Q}\left(d_1^{\frac{\alpha_1}{3}} \times \cdots \times d_n^{\frac{\alpha_n}{3}}\right) = \mathbb{Q}\left(d_1^{\frac{\beta_1}{3}} \times \cdots \times d_n^{\frac{\beta_n}{3}}\right).$$

By Lemma 3.1, this is equivalent to the existence of a rational a such that one of the three following possibilities is true :

$$\begin{cases} d_1^{\alpha_1} \times \cdots \times d_n^{\alpha_n} = d_1^{\beta_1} \times \cdots \times d_n^{\beta_n} \times a^3 & (1) \\ d_1^{\alpha_1} \times \cdots \times d_n^{\alpha_n} = d_1^{2\beta_1} \times \cdots \times d_n^{2\beta_n} \times a^3 & (2) \\ d_1^{\beta_1} \times \cdots \times d_n^{\beta_n} = d_1^{2\alpha_1} \times \cdots \times d_n^{2\alpha_n} \times a^3 & (3) \end{cases}$$

Now consider $\underline{\mu}$ and $\underline{\nu}$ and two non-zero elements of $\mathbb{F}_3^n$. Write in $\mathbb{Z}$ the equality $\mu_i = \nu_i + r_i + 3q_i$ with $0 \leqslant r_i < 3$ for all $i \in [\![0, n]\!]$. Then we have

$$d_1^{\mu_1} \times \cdots \times d_n^{\mu_n} = d_1^{\nu_1} \times \cdots \times d_n^{\nu_n} \times a^3 \iff \prod_{i=0}^{n} d_i^{r_i} = \left(\frac{a}{\prod_{i=0}^{n} d_i^{q_i}}\right)^3 \iff \underline{r} = \underline{0}$$

since no product of d_i's with corresponding exponents less than 2 can be a rational cube except for the trivial one, for we suppose the sequence $d_1, \ldots, d_n$ to be reduced.

Combining this with the three previous possibilities we indeed obtain the searched equivalence relation. The claimed number of such cubic subfields is directly deduced by counting the possible $\underline{\alpha}$ modulo this relation. □

Remark 3.6. (i) Given $K = \mathbb{Q}(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}})$ defined by a reduced sequence and $\underline{\alpha} \in [\![0, 2]\!]^n \setminus \{0\}$ we note $K_{\underline{\alpha}}$ the cubic subfield of K generated by the product $\prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}$.

(ii) Given a fixed multicubic field given by a reduced sequence, the data of a cubic subfield in the specific form is therefore equivalent to the data of a line or an hyperplane of the vector space $(\mathbb{F}_3)^n$. When considering these subfields we will therefore identify $[\![0, 2]\!]^n$ with $\mathbb{F}_3^n$

(iii) In fact we will see that all cubic subfields of a multicubic field are pure cubic fields of the previous form.

In order to study multicubic fields further we need to examine the set of the complex embeddings $\mathrm{Hom}(K, \mathbb{C})$.

3.2 Set of complex embedding

Fix a set of n distinct integers $\{d_1, \ldots, d_n\}$ supposed to constitute a reduced sequence as before and let K be the multicubic field associated to it. The degree of K over $\mathbb{Q}$ is less than 3^n. Given an embedding of K into $\mathbb{C}$, its action can be fully described by its action on each $\sqrt[3]{d_i}$ and therefore by the embedding it defines when restricted to each of the cubic fields $\mathbb{Q}(\sqrt[3]{d_i})$. Now if we fix an integer d_i the polynomial $X^3 - d_i$ factorises as

$$X^3 - d_i = (X - \sqrt[3]{d_i})(X - \zeta_3 \sqrt[3]{d_i})(X - \zeta_3^2 \sqrt[3]{d_i}).$$

We suppose d_i to be cube-free so $X^3 - d_i$ is irreducible over $\mathbb{Q}$. We then have the following isomorphism

$$\mathbb{Q}\left(\sqrt[3]{d_i}\right) \simeq \frac{\mathbb{Q}[X]}{(X^3 - d_i)}$$

and the three embeddings of $\mathbb{Q}\left(\sqrt[3]{d_i}\right)$ into $\mathbb{C}$ are the $\mathbb{Q}-$linear maps which send $\sqrt[3]{d_i}$ respectively to $\sqrt[3]{d_i}$, $\zeta_3\sqrt[3]{d_i}$ and $\zeta_3^2\sqrt[3]{d_i}$. We will note these embeddings $\sigma_i^{(0)}$, $\sigma_i^{(1)}$ and $\sigma_i^{(2)}$. Remark that $\sigma_i^{(0)}$ is the identity, that $\sigma_i^{(1)}$ and $\sigma_i^{(2)}$ are complex embeddings conjugate one to each other. Moreover all this description still applies to any cube-free integer m and the field $\mathbb{Q}\left(m^{\frac{1}{3}}\right)$, especially to the fields $K_{\underline{\alpha}}$. Thus we will similarly denote the three complex embeddings of $K_{\underline{\alpha}}$ by $\sigma_{\underline{\alpha}}^{(0)}$, $\sigma_{\underline{\alpha}}^{(1)}$ and $\sigma_{\underline{\alpha}}^{(2)}$. Finally any embedding $K \hookrightarrow \mathbb{C}$ can be described as

$$\bigotimes_{i=1}^{n} \sigma_i^{(\beta_i)}, (\beta_1, \ldots, \beta_n) \in [\![0, 2]\!]^n.$$

Given such a decomposition, the corresponding embedding will be written $\sigma^{(\underline{\beta})}$.

Remark 3.7. We can see that in this situation too the sets $[\![0, 2]\!]^n$ and $\mathbb{F}_3^n$ can be identified. Then the data of an embedding of K into $\mathbb{C}$ is equivalent to the data of a point in $(\mathbb{F}_3)^n$. We do not know yet if all such points can be obtained, which is equivalent to proving that the dimension of K is 3^n.

We will see that the duality of complex embeddings of K relatively to cubic subfields $K_{\underline{\alpha}}$ can be expressed as a duality situation in $(\mathbb{F}_3)^n$ thanks to their geometric interpretation as points and hyperplanes. This will help in proving the following

Theorem 3.8. *Consider K defined by a reduced sequence of integers $d_1, \ldots, d_n$. Then we have*

(i) $[K:\mathbb{Q}] = 3^n$ *and* $\left(\prod_{i=0}^{n} d_i^{\frac{\alpha_i}{3}}\right)_{\underline{\alpha}\in\mathbb{F}_3^n}$ *is a* $\mathbb{Q}-$*basis of* K*;*

(ii) *the set* $\mathrm{Hom}\,(K,\mathbb{C})$ *is exactly* $\left\{\sigma^{(\underline{\beta})} \mid \underline{\beta}\in\mathbb{F}_3^n\right\}$

We will study the action of an element $\sigma^{(\beta)}$ of $\mathrm{Hom}(K,\mathbb{C})$ on a cubic subfield $K_{\underline{\alpha}}$.

Recall that the three possibilities for $\sigma^{(\underline{\beta})}\left(d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}}\right)$ are

$$\begin{cases} \sigma_{\underline{\alpha}}^{(0)}(d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}}) = d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}};\\ \sigma_{\underline{\alpha}}^{(1)}(d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}}) = \zeta_3\times d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}};\\ \sigma_{\underline{\alpha}}^{(2)}(d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}}) = \zeta_3^2\times d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_1}{3}}. \end{cases}$$

We will relate the action of a morphism $\sigma^{(\underline{\beta})}$ on a field $K_{\underline{\alpha}}$ to a geometric relation between $\underline{\alpha}$ and $\underline{\beta}$ as said earlier. Recall that we can think of $\underline{\alpha}$ as an hyperplane and $\underline{\beta}$ as a point in the vector space $(\mathbb{F}_3)^n$. Let us fix some notation. Given $\underline{\alpha}\in(\mathbb{F}_3)^n\setminus\{0\}$ and $t\in\mathbb{F}_3$ we will note $H_{\underline{\alpha}}(t)$ the affine hyperplane of $(\mathbb{F}_3)^n$ defined by the equation

$$\alpha_1X_1+\cdots+\alpha_nX_n = t.$$

Proposition 3.9. *Let* $K=\mathbb{Q}\left(d_1^{\frac{1}{3}},\ldots,d_n^{\frac{1}{3}}\right)$ *be a multicubic field,* $\underline{\alpha}\in(\mathbb{F}_3)^n\setminus\{0\}$ *and* $\underline{\beta}\in(\mathbb{F}_3)^n$*. Then for any* $t\in\mathbb{F}_3$ *we have*

$$(\sigma^{(\underline{\beta})})_{|K_{\underline{\alpha}}} = \sigma_{\underline{\alpha}}^{(t)} \iff \underline{\beta}\in H_{\underline{\alpha}}(t).$$

Proof. We need to evaluate $\sigma^{(\underline{\beta})}$ on $d_1^{\frac{\alpha_1}{3}}\times\cdots\times d_n^{\frac{\alpha_n}{3}}$. We have

$$\begin{aligned}\sigma^{(\underline{\beta})}\left(\prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}\right) &= \bigotimes_{k=1}^{n}\sigma_k^{(\beta_k)}\left(\prod_{k=1}^{n} d_k^{\frac{\alpha_k}{3}}\right) = \prod_{k=1}^{n}\sigma_k^{(\beta_k)}(d_k^{\frac{\alpha_k}{3}}) = \prod_{k=1}^{n}(\sigma_k^{(\beta_k)}(d_k^{\frac{1}{3}}))^{\alpha_k}\\ &= \prod_{k=1}^{n}(\zeta_3^{\beta_k}d_k^{\frac{1}{3}})^{\alpha_k} = \prod_{k=1}^{n}\zeta_3^{\alpha_k\beta_k}\prod_{k=1}^{n}d_k^{\frac{\alpha_k}{3}}\\ &= \zeta_3^{\alpha_1\beta_1+\cdots+\alpha_n\beta_n}\times\prod_{k=1}^{n}d_k^{\frac{\alpha_k}{3}}\end{aligned}$$

Thus we have $(\sigma^{(\underline{\beta})})_{|K_{\underline{\alpha}}} = \sigma_{\underline{\alpha}}^{(t)}$ if, and only if, $\zeta_3^{\alpha_1\beta_1+\cdots+\alpha_n\beta_n} = \zeta_3^t$ which is equivalent to $\alpha_1\beta_1+\cdots+\alpha_n\beta_n = t$. □

Remark 3.10. We see that in order to analyse how the action of the embeddings of K are distributed among the different cubic subfields we have to do some affine geometry. First, the data of a cubic subfield is the same as the data of $\underline{\alpha}$ modulo multiplication by a non-zero element of $\mathbb{F}_3$ or equivalently the vectorial hyperplane $H_{\underline{\alpha}}(0)$. One can verify that the relation of the previous Proposition is coherent with the equality of $K_{\underline{\alpha}}$ and $K_{2\underline{\alpha}}$ by making the observation that $H_{\underline{\alpha}}(2t) = H_{2\underline{\alpha}}(t)$.

Now we will describe more precisely the action of the morphisms σ_i for $i \in [\![0, n]\!]$.

Lemma 3.11. *Let K be a multicubic field defined by a reduced sequence of integers $d_1, \ldots, d_n$. Then for all $\underline{\alpha} \in \mathbb{F}_3^n$, $i \in [\![0, n]\!]$ and $k \in [\![0, 2]\!]$ we have*

$$\sigma_i\left(\prod_{j=0}^{n} d_j^{\frac{\alpha_j}{3}}\right) = \zeta_3^k \times \prod_{j=0}^{n} d_j^{\frac{\alpha_j}{3}} \iff \alpha_i = k.$$

Proof. This is applying the above Proposition and remarking that this is true for a null $\underline{\alpha}$. □

Lemma 3.12. *Let K be a multicubic field defined by a reduced sequence of integers $d_1, \ldots, d_n$. Suppose K verifies the property of Theorem* 3.8. *Then for all $x \in K$ we have*

$$\forall \sigma \in \mathrm{Hom}(K, \mathbb{C}), \exists k \in [\![0, 2]\!], \sigma(x) = \zeta_3^k \times x \iff \exists \underline{\alpha} \in \mathbb{F}_3^n, \exists a \in \mathbb{Q}, x = a \prod_{i=0}^{n} d_i^{\frac{\alpha_i}{3}}.$$

Proof. Consider $x \in K$. We already know that the second assertion implies the first one. Suppose now the fist condition to be true. Since we assumed K to verify the property of Theorem 3.8, x can be written as

$$\sum_{\underline{\alpha} \in \mathbb{F}_3^n} x_{\underline{\alpha}} \left(\prod_{i=0}^{n} d_i^{\frac{\alpha_i}{3}}\right)$$

with $x_{\underline{\alpha}} \in \mathbb{Q}$ for every $\underline{\alpha} \in \mathbb{F}_3^n$ and write $\mathrm{Supp}(x) = \left\{\underline{\alpha} \in \mathbb{F}_3^n \mid x_{\underline{\alpha}} \neq 0\right\}$. There is nothing to prove if $\mathrm{Supp}(x)$ is the void space so we assume it is not trivial. The property being true for all morphisms is equivalent to be true for σ_i for all $i \in [\![1, n]\!]$. Fix such an integer. We can write $x = x_0 + x_1 + x_2$ with

$$x_t = \sum_{\underline{\alpha} \in \mathbb{F}_3^n \mid \alpha_i = t} x_{\underline{\alpha}} \left(\prod_{i=0}^{n} d_j^{\frac{\alpha_j}{3}}\right)$$

for all $t \in [\![0,2]\!]$. By Lemma 2 we have $\sigma_i(x) = x_0 + \zeta_3 \times x_1 + \zeta_3^2 \times x_2$. There is some $k_i \in [\![0,2]\!]$ such that $\sigma_i(x) = \zeta_3^{k_i}(x_0 + x_1 + x_2)$. Let us show that x is equal to x_{k_i}. We will do the calculation for $k = 1$ and omit the two other cases since they are almost identical. Therefore we have $x_0 + \zeta_3 \times x_1 + \zeta_3^2 \times x_2 = \zeta_3(x_0 + x_1 + x_2)$ and we can write $x_0(1-\zeta_3) + x_2(\zeta_3^2 - \zeta_3) = 0$ which leads to $x_0(1-\zeta_3) - x_2(1-\zeta_3)\zeta_3 = 0$. This is equivalent to $x_0 - x_2 \times \zeta_3 = 0$ and since x_0 and $x_2 x$ are real numbers it is equivalent to $x_0 = x_2 = 0$, and we can conclude that we have $x = x_1 = x_{k_i}$.

Remark that we proved that $\text{Supp}(x) \subseteq \left\{\underline{\alpha} \in \mathbb{F}_3^n \mid \alpha_i = k_i\right\}$. Looking at the action of the morphism σ_i forces the element of $\text{Supp}(x)$ to have a fixed ith coordinates. Geometrically $\text{Supp}(x)$ is included in an hyperplane of $\mathbb{F}_3^n$. By considering all of such morphisms we can see that we have

$$\text{Supp}(x) \subseteq \{\underline{\alpha} \in \mathbb{F}_3^n \mid \alpha_i = k_1\} \cap \cdots \cap \{\underline{\alpha} \in \mathbb{F}_3^n \mid \alpha_i = k_n\}$$

which is the point $\underline{k} = (k_1, \ldots, k_n)$. But $\text{Supp}(x)$ is not trivial so it is equal to this point and we can finally write $x = x_{\underline{k}} \prod_{i=0}^{n} d^{\frac{k_i}{3}}$ which gives us the desired result. □

Now that we have these results we can prove Theorem 3.8.

Proof. We will proceed by induction on the length n of the sequence $d_1, \ldots, d_n$. We proved the case $n = 1$ during he discussion at the beginning of the subsection. Now fix some integer $n \geqslant 1$ and suppose the searched results to be true for this n. Let K be a multicubic field defined by a reduced sequence $d_1, \ldots, d_{n+1}$. Consider L the multicubic field defined the reduced sequence of integers $d_1, \ldots, d_n$. Then $K = L\left(d_{n+1}^{\frac{1}{3}}\right)$. First let us show that K has degree 3^{n+1} over $\mathbb{Q}$. Since $[L : \mathbb{Q}] = 3^n$ we need to prove that $d_{n+1}^{\frac{1}{3}}$ does not belong to L. Suppose the contrary. Every element of $\text{Hom}(L, \mathbb{C})$ permutes the roots of $X^3 - d_{n+1}$ therefore sends $d_{n+1}^{\frac{1}{3}}$ to some $\zeta_3^k d_{n+1}^{\frac{1}{3}}$ with $k \in [\![0,2]\!]$. But L verifies the induction hypothesis and we can apply Lemma 3 to $d_{n+1}^{\frac{1}{3}}$ obtaining

$$d_{n+1}^{\frac{1}{3}} = a \times \prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}$$

which implies the equality

$$\mathbb{Q}\left(d_{n+1}^{\frac{1}{3}}\right) = \mathbb{Q}\left(\prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}\right).$$

This is impossible because the sequence $d_1, \ldots, d_{n+1}$ is reduced. Therefore $d_{n+1}^{\frac{1}{3}} \notin K$ and we have $[K : \mathbb{Q}] = 3^{n+1}$. Let us now prove that the complex embeddings of K are exactly those of the described form. Using the induction hypothesis it is clear that there are 3^{n+1} such morphisms and this gives us the desired result. □

We will pursue the study of complex embeddings of multicubic fields by considering its Galois closure. We will be able to deduce from this other structural results on the field considered.

Given a field K we will note in general $\tilde{K}$ its Galois closure. Let us fix K a multicubic field generated by a reduced sequence $d_1, \ldots, d_n$. We will see that $\tilde{K}$ is $K(\zeta_3)$. Given $\sigma \in \mathrm{Hom}(K, \mathbb{C})$ a complex embedding of K we will note $\tilde{\sigma}$ the field morphism of $K(\zeta_3)$ obtained as

$$\begin{array}{ccc} K(\zeta_3) & \longrightarrow & K(\zeta_3) \\ x \in K & \longmapsto & \sigma(x) \\ \zeta_3 & \longmapsto & \zeta \end{array}$$

and τ the morphism which acts as the complex conjugation.

Proposition 3.13. *The Galois closure of K is then $K(\zeta_3)$ and its Galois group is generated by the set $\{\tau\} \times \{\tilde{\sigma_i} \mid i \in [\![1, n\}$. Moreover it is isomorphic to the group*

$$\frac{\mathbb{Z}}{2\mathbb{Z}} \ltimes \left(\frac{\mathbb{Z}}{3\mathbb{Z}}\right)^n = \langle s, r_1, \ldots, r_n \mid s^2 = 1, r_i^3 = 1, sr_isr_i = 1 \rangle.$$

Proof. The field $K(\zeta_3)$ has dimension 2×3^n over $\mathbb{Q}$. Therefore in order to prove that it is Galois with the claimed Galois group it suffices to prove that the last has cardinality 2×3^n. Note this G for the sake of the proof. By the previous study on complex embeddings of K we already know that the group generated by the $\tilde{\sigma_i}$ has order 3^n which divides the order of G. Moreover the complex conjugation has order 2 which again divides the order of G. Therefore 2×3^n divides the order of G which is smaller than the dimension of $K(\zeta_3)$ and we have the desired result.

Now let us prove that G has the announced structure. We already stated that the complex conjugation has order 2. Clearly the $\tilde{\sigma_i}$ commute and we have $\tilde{\sigma_i}^k(d_i^{\frac{1}{3}}) = \zeta_3^k d_i^{\frac{1}{3}}$ proving that all of the $\tilde{\sigma_i}$ have order 3 and that they generate a subgroup isomorphic to $(\frac{\mathbb{Z}}{3\mathbb{Z}})^n$. Let us prove that the last relation holds. For all $i \in [\![1, n]\!]$ we have

$$\tau\tilde{\sigma_i}\tau\tilde{\sigma_i}(d_i^{\frac{1}{3}}) = \tau\tilde{\sigma_i}(\tau(\zeta_3 d_i^{\frac{1}{3}})) = \tau\tilde{\sigma_i}(\zeta_3^2 d_i^{\frac{1}{3}}) = \tau(d^{\frac{1}{3}}) = d^{\frac{1}{3}}$$

and

$$\tau\tilde{\sigma}_i\tau\tilde{\sigma}_i(\zeta_3) = \tau\tilde{\sigma}_i(\tau(\zeta_3)) = \tau\tilde{\sigma}_i(\zeta_3^2) = \tau(\zeta_3^2) = \zeta_3$$

which means that $\tau\tilde{\sigma}_i\tau\tilde{\sigma}_i$ is indeed the identity morphism on $\tilde{K}$. □

Remark 3.14. We can see that any element of $\mathrm{Gal}(\widetilde{K}/\mathbb{Q})$ can be written uniquely as $\tau^{\alpha}\prod_{i=1}^{n}\tilde{\sigma}_i^{\beta_i}$ with $(\alpha, \beta_1, \ldots, \beta_n) \in \mathbb{F}_2 \times \mathbb{F}_3^n$.

As said before we will use the Galois group to study the structure of the multicubic field K. Recall that given a Galois extension M/N there is a correspondence between subgroups of the Galois group $\mathrm{Gal}(M/N)$ and subfields of the extension, which is given by invertible decreasing maps.

Remark 3.15. Let F be a subfield of $\widetilde{K}$. Then F is a subfield of $K = \mathrm{Inv}(\tau)$ if, and only if, the group associated to F contains τ.

One of the first properties that we can deduce from the structure of the Galois group is that the cubic subfields of the form $K_{\underline{\alpha}}$ considered previously are all of the cubic subfield.

Proof. Consider F a cubic subfield of K. The associated subgroup H of $\mathrm{Gal}(\widetilde{K}/\mathbb{Q})$ is generated by a set

$$S = \left\{ \tau^{\alpha^{(1)}}\prod_{i=1}^{n}\tilde{\sigma}_i^{\beta_i^{(1)}}, \ldots, \tau^{\alpha^{(r)}}\prod_{i=1}^{n}\tilde{\sigma}_i^{\beta_i^{(r)}} \right\}$$

with $r \geqslant 1$. Since F is real we know that τ belongs to H and we can consider that we have

$$S = \left\{ \tau, \prod_{i=1}^{n}\tilde{\sigma}_i^{\beta_i^{(1)}}, \ldots, \prod_{i=1}^{n}\tilde{\sigma}_i^{\beta_i^{(r)}} \right\}$$

and therefore we can see that the data of H is the same as the data of the subgroup generated by $S \setminus \{\tau\}$ which is a subgroup of $(\frac{\mathbb{Z}}{3\mathbb{Z}})^n$. Moreover we have $[\widetilde{K} : F] = 2 \times 3^{n-1}$ thus the order of H is the same by the Galois correspondence and therefore the group generated by $S \setminus \{\tau\}$ has order 3^{n-1}. Cubic subfields of K are then in one-to-one correspondence with subgroup of $(\frac{\mathbb{Z}}{3\mathbb{Z}})^n$ of order 3^{n-1}. Counting the last is equivalent to counting sub-vector spaces of $\mathbb{F}_3^n$ of dimension 3^{n-1} or 3. Their number is

$$\binom{n}{3}_3 = \frac{(3^n-1)(3^{n-1}-1)\ldots(3^{n-(n-1)-1}-1)}{(3^{n-1}-1)(3^{n-2}-1)\ldots(3-1)} = \frac{3^n-1}{2}.$$

We saw in Proposition 3.5 that there are $\frac{3^n-1}{2}$ cubic subfields of the form $K_{\underline{\alpha}}$. □

The cubic subfields are of particular interest for us because as in the multi-quadratic case, we will compute their units and construct from these the units of K. As we will see later their number is the one we need.

Lemma 3.16. *Any subfield F of K of degree 3^{n-1} is of the form* $\mathrm{Inv}(\tilde{\sigma}^{(\underline{\beta})}, \tau)$ *and is a multicubic field.*

Proof. We have $[\widetilde{K} : F] = 6$ therefore the associated subgroup H of $\mathrm{Gal}(\widetilde{K}/\mathbb{Q})$ has order 6. Since $F \subset K$ we know that τ is in H and by using the orders we can conclude that H is generated by τ and only one $\tilde{\sigma}^{(\underline{\beta})}$ with $\underline{\beta} \neq 0$. Let fix these notations for the proof. We note $I = \{i_1, i_2, \ldots, i_r\}$ the set of indexes of the non-zero coefficients of $\underline{\beta}$. We can suppose $i_1 < \cdots < i_r$. Now consider the sets

$$S = \{d_j^{\frac{1}{3}} \mid j \in [\![1, n]\!] \setminus I\}$$

and

$$T = \left\{ \sqrt[3]{d_{i_1}^{2-\delta_{\beta_{i_1},\beta_{i_k}}}} \, d_{i_k}^{\frac{1}{3}} \mid k \in [\![2, r]\!] \right\}$$

Then the cardinal of T is $r - 1$ and any element of T is invariant under the action of $\tilde{\sigma}^{\underline{\beta}}$. The field

$$L = \mathbb{Q}\left(S \cup T\right)$$

is therefore a field defined by $n - r + r - 1 = n - 1$ cube roots of integers and its elements are invariant under the action of $\tilde{\sigma}^{(\underline{\beta})}$. Recall that we assumed the sequence $d_1, \ldots, d_n$ to be reduced. This implies that neither the elements d_j nor the elements $d_{i_1}^{2-\delta_{\beta_{i_1},\beta_{i_k}}} d_{i_k}$ are cubes. Thus we know that L is a multicubic field. Let us show now that the sequence defined by $S \cup T$ is a reduced sequence. First note for simplicity $\lambda_{i_k} = 2 - \delta_{\beta_{i_1},\beta_{i_k}}$ which is 1 or 2. Consider now without any loss of generality that we have $I = [\![1, r]\!]$. Let $(\alpha_2, \ldots, \alpha_n) \in \mathbb{F}_3^{n-1}$ and assume that

$$P = \prod_{k=2}^{r} (d_1^{\lambda_k} d_k)^{\alpha_k} \times \prod_{k=r+1}^{n} d_k^{\alpha_k} = d_1^{\alpha_1} \prod_{k=2}^{n} d_k^{\alpha_k}$$

– where $\alpha_1 = \sum_{k=2}^{r} \lambda_k \alpha_k$ – is a cube. We can write $\alpha_1 = 3q + r$ with $0 \leqslant r < 3$ thus the product

$$d_1^r \prod_{k=2}^{n} d_k^{\alpha_k}$$

is a cube. But $(r, \alpha_2, \ldots, \alpha_n) \in \mathbb{F}_3$ and the sequence $(d_1, \ldots, d_n)$ is reduced therefore $r = \alpha_2 = \cdots = \alpha_n = 0$. Consequently the sequence defined by $S \cup T$ is

reduced too. Now L is a multicubic field defined by a reduced sequence of length $n-1$ so by Theorem 3.8 it has degree 3^{n-1}. Finally $L \subset F$ and they have the same degree so they are identical which means that F is indeed a multicubic field. □

Theorem 3.17. *Let K be a multicubic field. Any subfield F of K is a multicubic field.*

Proof. We will proceed by induction on $[K : F]$. The previous Lemma states that it is true for $[K : F] = 3^1$. Consider the result to be true for $[K : F] = 3^r$ for some $r \geqslant 1$ and suppose $[K : F] = 3^{r+1}$. As usual note H the subgroup of $\mathrm{Gal}(\widetilde{K}/\mathbb{Q})$ such that $F = \mathrm{Inv}(H)$. Just as before we can write $H = \langle \tau, \tilde{\sigma}^{(\underline{\beta_1})}, \ldots, \tilde{\sigma}^{(\underline{\beta_{r+1}})} \rangle$. Note L the field fixed by the group $\langle \tau, \tilde{\sigma}^{(\underline{\beta_1})}, \ldots, \tilde{\sigma}^{(\underline{\beta_r})} \rangle < H$. By the Galois correspondence we know that $[K : L] = 3^r$ and that F is subfield of L with $[L : F] = 3$. The induction hypothesis states that L is multicubic field and we can apply again the previous Lemma to the extension L/F to conclude that F is a multicubic field too. □

We see that the structure of multicubic fields is similar to the one of multiquadratic fields even if they are not Galois. This structure will allow us to work recursively and fasten considerably our computations. The following result is similar to Lemma 5.1 in [4] and is a generalisation of a result over Bicubic fields proved by Charles Parry in [17].

Notation : For now on if $\tilde{\sigma}$ is an element of $\mathrm{Gal}(\widetilde{K}/\mathbb{Q})$ we will denote by $K_{\tilde{\sigma}}$ the field $\mathrm{Inv}(\tau, \tilde{\sigma}) = \tilde{K}_{\tilde{\sigma}} \cap \mathbb{R}$, and by $H(\tilde{K})$ the subgroup $\{\tilde{\sigma} \mid \sigma \in \mathrm{Hom}(K, \mathbb{C})\}$.

Proposition 3.18. *Let K be a multicubic field with $[K : \mathbb{Q}] > 3$. Consider u and v two elements of $H(\tilde{K})$ which are independent. Then for any $x \in K$ we have*

$$x^3 = x_u x_v x_{uv} x_{u^2 v}$$

where $x_w \in K_w$ for every $w \in \{u, v, uv, u^2 v\}$. Moreover if x is a unit of K then x_w is a unit of K_w for all $w \in \{u, v, uv, u^2 v\}$.

Proof. As mentioned before the proof relies exactly on the same idea that appears in [4, 17]. For every element $x \in K$ we can rewrite the cube as

$$\begin{aligned} x^3 &= \frac{x \cdot u(x) \cdot u^2(x) \cdot x \cdot v(x) \cdot v^2(x) \cdot x \cdot uv(x) \cdot (uv)^2(x)}{u(x) \cdot u^2(x) \cdot v(x) \cdot v^2(x) \cdot uv(x) \cdot (uv)^2(x)} \\ &= \frac{\mathrm{N}_{\tilde{K}/\tilde{K}_u}(x)\mathrm{N}_{\tilde{K}/\tilde{K}_v}(x)\mathrm{N}_{\tilde{K}/\tilde{K}_{uv}}(x)}{\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2 v}}(u(x) \cdot uv(x))}. \end{aligned}$$

Then for all $w \in \{u, v, uv, u^2v\}$ we note x_w the relative norm element corresponding to w in the previous expression. Since x is an element of K any of the norm in the numerator $\mathrm{N}_{\tilde{K}/\tilde{K}_w}(x)$ is in the fact the same as the relative norm $\mathrm{N}_{K/K_w}(x)$ which is an element of K_w. The relative norm $\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2v}}(u(x) \cdot uv(x))$ is in $\tilde{K}_{u^2v}$. However x^3 is in $\mathbb{R}$ as well as the numerator therefore $\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2v}}(u(x) \cdot uv(x))$ is in $\tilde{K}_{u^2v} \cap \mathbb{R} = K_{u^2v}$. The statement concerning units is clear given the algebraic expression of the elements as relative norms. □

3.3 Unit Group

The structure of the group units of a number field is related to its complex embeddings. Consider a multicubic field K defined by a reduced sequence $d_1, \ldots, d_n$. We can see that a multicubic field K has only one real embedding – the identity – and $3^n - 1$ complex ones. Therefore we know that the group of units $\mathcal{O}_K^\times$ is isomorphic to

$$\frac{\mathbb{Z}}{2\mathbb{Z}} \times \mathbb{Z}^{\frac{3^n-1}{2}}.$$

In the special case of the cubic subfields $K_{\underline{\alpha}}$ we have

$$\mathcal{O}_{K_{\underline{\alpha}}}^\times \simeq \frac{\mathbb{Z}}{2\mathbb{Z}} \times \mathbb{Z}.$$

Then for every $\underline{\alpha}$ we can write $\mathcal{O}_{K_{\underline{\alpha}}}^\times = \left\{\pm(\epsilon_{\underline{\alpha}})^k \mid k \in \mathbb{Z}\right\}$ with $\epsilon_{\underline{\alpha}} > 1$ just as in the quadratic case. This specific generating unit will be called the fundamental unit. Just as the authors of [4] defined the subgroup of multiquadratic units we will define the subgroup of multicubic units using the units of cubic subfields.

Definition 3.19. Consider a multicubic field $K = \mathbb{Q}\left(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}}\right)$ defined by a reduced sequence. We call multicubic units and write MCU(K) – or MCU if there is no ambiguity – the subgroup of $\mathcal{O}_K^\times$ generated by the set $\left\{-1, \epsilon_{\underline{\alpha}} \mid \underline{\alpha} \in (\mathbb{F}_3)^n\right\}$.

Just as in the multiquadratic case we will see that MCU is a full-rank subgroup of $\mathcal{O}_K^\times$ and that the basis $\left\{\epsilon_{\underline{\alpha}} \mid \underline{\alpha} \in (\mathbb{F}_3)^n\right\}$ yields an orthogonal basis under the action of the Log-embedding.

Proposition 3.20. *Let K be a multicubic field of degree 3^n. Then we have*

$$(\mathcal{O}_K^\times)^{3^{n-1}} < \mathrm{MCU}.$$

Moreover MCU *is a full-rank subgroup of* $\mathcal{O}_K^\times$ *with* $[\mathcal{O}_K^\times : \mathrm{MCU}]$ *divides* $3^{\frac{3^n-1}{2}}$ *and the set* $\left\{-1, \epsilon_{\underline{\alpha}} \mid \underline{\alpha} \in (\mathbb{F}_3)^n\right\}$ *is in fact a basis.*

Proof. The result is trivial for $n = 1$. Now assume it is true for some fixed $n \geqslant 1$ and let K be a multicubic field of degree 3^{n+1}. As stated in Proposition 3.18 we have

$$(\mathcal{O}_K^\times)^3 < \mathcal{O}_{K_u}^\times \mathcal{O}_{K_v}^\times \mathcal{O}_{K_{uv}}^\times \mathcal{O}_{K_{u^2v}}^\times$$

with u, v being two elements of $H(\tilde{K})$. Then for every $w \in \{u, v, uv, u^2v\}$ the field K_w is a subfield of K of dimension 3^n. Since it is a multicubic field too it verifies the recursion hypothesis. Therefore $(\mathcal{O}_{K_w}^\times)^{3^{n-1}}$ is included in $\mathrm{MCU}(K_w)$ which is itself included in $\mathrm{MCU}(K)$. Thus we have

$$(\mathcal{O}_K^\times)^{3^n} = ((\mathcal{O}_K^\times)^3)^{3^{n-1}} < (\mathcal{O}_{K_u}^\times \mathcal{O}_{K_v}^\times \mathcal{O}_{K_{uv}}^\times \mathcal{O}_{K_{u^2v}}^\times)^{3^{n-1}} < \mathrm{MCU}(K).$$

We have proven the first result. The property on the index follows immediately from $(\mathcal{O}_K^\times)^3 < \mathrm{MCU}(K) < \mathcal{O}_K^\times$ and the fact that the units of a multicubic field of degree 3^n is a free group of rank $\frac{3^n-1}{2}$. The previous tower of groups shows that MCU is indeed a full-rank subgroup of $\mathcal{O}_K^\times$ and since the cardinal of the generating set $\left\{-1, \epsilon_{\underline{\alpha}} \mid \underline{\alpha} \in (\mathbb{F}_3)^n\right\}$ equals the rank of the group we can conclude that this set is a basis of MCU. □

In order to study the geometry of the lattice $\mathrm{Log}_K(\mathrm{MCU})$ we need to evaluate the action of each embedding $\sigma^{(\underline{\beta})}$ on the units $\epsilon_{\underline{\alpha}}$ which is induced by the action of the embedding on the cubic field $K_{\underline{\alpha}}$ and thus on the generator $d_1^{\frac{\alpha_1}{3}} \times \cdots \times d_n^{\frac{\alpha_1}{3}}$. Recall that we introduced a geometrical point of view regarding this duality situation in Subsection 3.2. We will use it to describe properly the vectors $\mathrm{Log}_K(\epsilon_{\underline{\alpha}})$. The following proposition can be deduced from known affine geometric results.

Proposition 3.21. *Consider* r *elements* $\underline{\alpha}_1, \ldots, \underline{\alpha}_r$ *linearly independent in* $(\mathbb{F}_3)^n$. *We have the following geometric facts :*

(i) *For every* r*-tuple* $(t_1, \ldots, t_r)$ *of elements of* $\mathbb{F}_3$*, the intersection* $\bigcap_{k=1}^{r} H_{\underline{\alpha}_k}(t_k)$ *defines an affine variety of dimension* $n - r$.

(ii) *For every* r*-tuple* $(t_1, \ldots, t_r)$ *and every* $\underline{\gamma} \notin \mathrm{Vect}(\underline{\alpha}_1, \ldots, \underline{\alpha}_r)$ *we have*

$$\bigcap_{k=1}^{r} H_{\underline{\alpha}_k}(t_k) = \bigsqcup_{t \in \mathbb{F}_3} \left(\bigcap_{k=1}^{r} H_{\underline{\alpha}_k}(t_k) \right) \cap H_{\underline{\gamma}}(t).$$

If we transfer this in the setting of number fields and embeddings, we can tell that the actions of the embeddings of a multicubic field K into $\mathbb{C}$ are uniformly distributed among the cubic subfields of the form $K_{\underline{\alpha}}$. For example if we fix a cubic subfield each one of the three possible actions comes from exactly 3^{n-1} embeddings $K \hookrightarrow \mathbb{C}$. Then if we choose a second distinct cubic subfield, all of the three previous set have exactly 3^{n-2} elements inducing each of the three complex embeddings of that second subfield.

This well distributed duality will give rise to a nice geometric situation in the Log-unit lattice. Here we consider the Log map as follow

$$\begin{array}{rccc} \mathrm{Log}_K : & K^* & \longrightarrow & \mathbb{R}^{3^n} \\ & x & \longmapsto & (\log|\bigotimes_{i=1}^{n} \sigma_i^{(\beta_i)}(x)|)_{\beta_i \in \mathbb{F}_3}. \end{array}$$

Proposition 3.22. *Consider a multicubic field* $K = \mathbb{Q}\left(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}}\right)$. *The vectors* $\mathrm{Log}_K(\epsilon_{\underline{\alpha}})$ *with* $\underline{\alpha} \in \mathbb{F}_3^n \setminus \{0\}$ *form an orthogonal family in* $\mathbb{R}^n$.

Proof. Consider $\underline{\alpha}$ and $\underline{\gamma}$ two elements of $(\mathbb{F}_3)^n$ independent over $\mathbb{F}_3$. We will evaluate the scalar product of $\mathrm{Log}_K(\epsilon_{\underline{\alpha}})$ and $\mathrm{Log}_K(\epsilon_{\underline{\gamma}})$.

$$\begin{aligned} \Big(\mathrm{Log}_K(\epsilon_{\underline{\alpha}}) \mid \mathrm{Log}_K(\epsilon_{\underline{\gamma}})\Big) &= \sum_{\sigma \in \mathrm{Hom}(K,\mathbb{C})} \log|\sigma(\epsilon_{\underline{\alpha}})| \times \log|\sigma(\epsilon_{\underline{\gamma}})| \\ &= \sum_{\underline{\beta} \in (\mathbb{F}_3)^n} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\alpha}})| \times \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})|. \end{aligned}$$

Now we will use the geometric properties described before to rewrite the sum over well distributed subsets. First recall that $(\mathbb{F}_3)^n = \bigsqcup_{t \in \mathbb{F}_3} H_{\underline{\alpha}}(t)$ which allows us to write

$$\begin{aligned} \Big(\mathrm{Log}_K(\epsilon_{\underline{\alpha}}) \mid \mathrm{Log}_K(\epsilon_{\underline{\gamma}})\Big) &= \sum_{t \in \mathbb{F}_3} \sum_{\underline{\beta} \in H_{\underline{\alpha}}(t)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\alpha}})| \times \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})| \\ &= \sum_{t \in \mathbb{F}_3} \sum_{\underline{\beta} \in H_{\underline{\alpha}}(t)} \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})| \times \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})| \\ &= \sum_{t \in \mathbb{F}_3} \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})| \sum_{\underline{\beta} \in H_{\underline{\alpha}}(t)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})|. \end{aligned}$$

Now we can decompose the hyperplanes $H_{\underline{\alpha}}(t)$ as $H_{\underline{\alpha}}(t) = \bigsqcup_{s \in \mathbb{F}_3} H_{\underline{\alpha}}(t) \cap H_{\underline{\gamma}}(s)$ and we can write

$$\sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})| = \sum_{s\in\mathbb{F}_3}\left(\sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)\cap H_{\underline{\gamma}}(s)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})|\right)$$
$$= \sum_{s\in\mathbb{F}_3}\left(\sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)\cap H_{\underline{\gamma}}(s)} \log|\sigma^{(s)}_{\underline{\gamma}}(\epsilon_{\underline{\gamma}})|\right).$$

In the right-hand side of the previous equality, every term of the second sum have the same value. Moreover the set we are summing over has 3^{n-2} elements since it is a $(n-2)-$ dimensional affine variety of $(\mathbb{F}_3)^n$. This gives us

$$\sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\gamma}})| = \sum_{s\in\mathbb{F}_3} 3^{n-2}\times\log|\sigma^{(s)}_{\underline{\gamma}}(\epsilon_{\underline{\gamma}})|$$

and the scalar product can be rewritten

$$\left(\mathrm{Log}_K(\epsilon_{\underline{\alpha}}) \mid \mathrm{Log}_K(\epsilon_{\underline{\gamma}})\right) = \sum_{t\in\mathbb{F}_3}\log|\sigma^{(t)}_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})| \sum_{s\in\mathbb{F}_3} 3^{n-2}\times\log|\sigma^{(s)}_{\underline{\gamma}}(\epsilon_{\underline{\gamma}})|$$
$$= 3^{n-2}\left(\sum_{t\in\mathbb{F}_3}\log|\sigma^{(t)}_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})|\right)\left(\sum_{s\in\mathbb{F}_3}\log|\sigma^{(s)}_{\underline{\gamma}}(\epsilon_{\underline{\gamma}})|\right)$$
$$= 3^{n-2}\times\log\left(\prod_{t\in\mathbb{F}_3}|\sigma^{(t)}_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})|\right)\times\log\left(\prod_{s\in\mathbb{F}_3}|\sigma^{(s)}_{\underline{\gamma}}(\epsilon_{\underline{\gamma}})|\right)$$
$$= 3^{n-2}\times\log\left(|N_{K_{\underline{\alpha}}/\mathbb{Q}}(\epsilon_{\underline{\alpha}})|\right)\times\log\left(|N_{K_{\underline{\gamma}}/\mathbb{Q}}(\epsilon_{\underline{\gamma}})|\right).$$

The elements $\epsilon_{\underline{\alpha}}$ and $\epsilon_{\underline{\gamma}}$ are units thus their algebraic norm is ±1 and the scalar product is

$$\left(\mathrm{Log}_K(\epsilon_{\underline{\alpha}}) \mid \mathrm{Log}_K(\epsilon_{\underline{\gamma}})\right) = 3^{n-2}\times\log(1)\times\log(1) = 0.$$

□

The orthogonality of the vectors $\mathrm{Log}_K(\epsilon_{\underline{\alpha}})$ assures that we are in the best situation possible to solve problems in the lattice $\mathrm{Log}_K(\mathrm{MCU})$. However in order to use this sublattice to decode in the Log-unit lattice it would need to be close from $\mathrm{Log}_K(\mathcal{O}_K^\times)$ which is not the case experimentally.

We can evaluate the norm of the basis vector of $\mathrm{Log}_K(\mathrm{MCU})$.

Lemma 3.23. *Consider a multicubic field* $K = \mathbb{Q}\left(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}}\right)$. *Then for all* $\underline{\alpha} \in \mathbb{F}_3^n \setminus \{0\}$ *we have*

$$\|\mathrm{Log}_K(\epsilon_{\underline{\alpha}})\|^2 = 3^{n-1} \times \|\mathrm{Log}_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})\|^2.$$

Proof. By following the same arguments as in previous calculations we can write

$$\|\mathrm{Log}_K(\epsilon_{\underline{\alpha}})\|^2 = \big(\mathrm{Log}_K(\epsilon_{\underline{\alpha}}) \mid \mathrm{Log}_K(\epsilon_{\underline{\alpha}})\big) = \sum_{t\in\mathbb{F}_3} \sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)} \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\alpha}})| \times \log|\sigma^{(\underline{\beta})}(\epsilon_{\underline{\alpha}})|$$

which is

$$\|\mathrm{Log}_K(\epsilon_{\underline{\alpha}})\|^2 = \sum_{t\in\mathbb{F}_3} \sum_{\underline{\beta}\in H_{\underline{\alpha}}(t)} \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})| \times \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})|$$

But $H_{\underline{\alpha}}(t)$ has 3^{n-1} elements so

$$\|\mathrm{Log}_K(\epsilon_{\underline{\alpha}})\|^2 = \sum_{t\in\mathbb{F}_3} 3^{n-1} \times \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})| \times \log|\sigma_{\underline{\alpha}}^{(t)}(\epsilon_{\underline{\alpha}})| = 3^{n-1} \times \|\mathrm{Log}_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})\|^2.$$

□

Now we will be able to express the norm of $\mathrm{Log}_K(\epsilon_{\underline{\alpha}})$ in function of the value of $\epsilon_{\underline{\alpha}}$.

Proposition 3.24. *Consider a multicubic field* $K = \mathbb{Q}\left(p_1^{\frac{1}{3}}, \ldots, p_n^{\frac{1}{3}}\right)$. *Then for all* $\underline{\alpha}$ *we have*

$$\|\mathrm{Log}_K(\epsilon_{\underline{\alpha}})\| = \sqrt{\frac{3^n}{2}} \times \log(\epsilon_{\underline{\alpha}}).$$

Proof. We will use the expression found in the previous lemma and express the quantity $\|\mathrm{Log}_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})\|^2$. First recall some facts. We have $\epsilon_{\underline{\alpha}} > 1$ therefore $\log(|\epsilon_{\underline{\alpha}}|) = \log(\epsilon_{\underline{\alpha}}) > 0$. Moreover the quantity $\sigma_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})$ and $\sigma_{\underline{\alpha}}^{(2)}(\epsilon_{\underline{\alpha}})$ are conjugates thus they have the same modulus. We can write

$$\|\mathrm{Log}_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})\|^2 = (\log(\epsilon_{\underline{\alpha}}))^2 + 2(\log|\sigma_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})|)^2.$$

Then we know that we have

$$\log(\epsilon_{\underline{\alpha}}) + \log|\sigma_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})| + \log|\sigma_{\underline{\alpha}}^{(2)}(\epsilon_{\underline{\alpha}})| = \log(\epsilon_{\underline{\alpha}}) + 2\log|\sigma_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})| = \log|N_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})| = 0$$

which gives

$$\log|\sigma_{\underline{\alpha}}(\epsilon_{\underline{\alpha}})| = -\frac{\log(\epsilon_{\underline{\alpha}})}{2}.$$

By using this equality we obtain

$$\|\mathrm{Log}_{K_{\underline{\alpha}}}(\epsilon_{\underline{\alpha}})\|^2 = (\log(\epsilon_{\underline{\alpha}}))^2 + 2 \times \left(-\frac{\log(\epsilon_{\underline{\alpha}})}{2}\right)^2 = \frac{3}{2} \times (\log(\epsilon_{\underline{\alpha}}))^2$$

and by consequence

$$\|\mathrm{Log}_{K}(\epsilon_{\underline{\alpha}})\|^2 = 3^{n-1} \times \frac{3}{2} \times (\log(\epsilon_{\underline{\alpha}}))^2 = \frac{3^n}{2} \times (\log(\epsilon_{\underline{\alpha}}))^2.$$

The searched equality is found by taking the square root of the previous equation. □

4 Algorithms and experiments

In all the following we will consider multicubic fields defined by reduced sequence. Fix $K = \mathbb{Q}(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}})$ such a field. We proved that K has dimension 3^n over $\mathbb{Q}$ and that the elements of the form

$$\prod_{i=1}^{n} d_i^{\frac{\alpha_i}{3}}$$

with $\underline{\alpha} \in [\![0, 2]\!]^n$ form a basis of $K/\mathbb{Q}$. In fact we can consider the cube-free part of each of these elements which we will do in all the following. Therefore elements of K are represented as vectors of length 3^n with rational coefficients. Moreover we can see K as a relative extension of degree 3 over a multicubic subfield of dimension 3^{n-1} over $\mathbb{Q}$. The most natural is to write K as $L(d_n^{\frac{1}{3}})$ with $L = \mathbb{Q}(d_1^{\frac{1}{3}}, \ldots, d_{n-1}^{\frac{1}{3}})$. If we choose this point of view we can see elements of K as vectors of length 3 with coefficients in the subfield L.

As we saw already one important tool for us is the Log-embedding. As in [4] we will not compute the exact Log-embedding but an approximate version of it, very much like the authors did. This leads us to represent any non zero element $x \in K$ by the pair $(x, \mathrm{ApproxLog}_K(x))$ where x is a vector with rational coefficients as described before and $\mathrm{ApproxLog}_K(x)$ will be a vector as described later.

In the following we make an extensive use of the famous LLL algorithm presented in [16]. We solve multivariate linear systems in the real field which is a classical use of it.

General procedure

In [4] the authors compute units of a multiquadratic field K as follow :

(i) Recursively compute the units of three subfields K_1, K_2, K_3 such that $(\mathcal{O}_K^\times)^2 < U = \prod_{i=1}^3 \mathcal{O}_{K_i^\times} < \mathcal{O}_K^\times$;

(ii) Find non trivial squares of U;

(iii) Calculate their square roots.

For multicubic fields this general procedure can be followed : only replace "squares" by "cubes" and consider four subfields in the first step as in Proposition 3.18. The step (ii) can be directly adapted and is described in Subsections 4.1 and 4.2 However computing cube roots is more complicated as seen in Subsection 4.3.

4.1 Finding Good Primes

As in [4] we will need to be able to find primes verifying fixed cubic conditions with respect to the d_i's. Consider $(d_1, \ldots, d_n)$ a reduced sequence and $C = (c_1, \ldots, c_n) \in \{0,1\}^n$. A good prime for $\underline{d}$ and C is a prime p such that d_i is a cube modulo p if, and only if, c_i is 1.

In particular we need to find good primes p for the condition sequence $(1, \ldots, 1)$ in order to construct morphisms from K^* into finite fields $\mathbb{F}_p$. Remark that the primes should not divide any of the integers d_i. Now if we fix a prime $p > 3$ we have the following situation :

- if $p \equiv 1 \pmod 3$ then $\mathbb{F}_p$ contains a fundamental cube root of unity and $\frac{\mathbb{F}_p^*}{(\mathbb{F}_p^*)^3} \simeq \mathbb{F}_3$;
- if $p \equiv 2 \pmod 3$ then $\mathbb{F}_p$ does not contain a fundamental cube root of unity and $\frac{\mathbb{F}_p^*}{(\mathbb{F}_p^*)^3} \simeq 1$.

Therefore we can have different strategies depending on our goal. If we want the condition $(1, \ldots, 1)$ to be verified we might consider primes only congruent to 2 modulo 3 as long as we do not need a non-trivial cube root of 1 to be in the field $\mathbb{F}_p$. Otherwise we have to consider primes which are congruent to 1 modulo 3.

Let us now describe how the algorithm operates in this case. First we have to draw a prime p and verify that it is not congruent to 2 modulo 3. This happens with probability $\frac{1}{2}$. Then we have to check whether the sequence of cube conditions C is verified by $(d_1, \ldots, d_n)$ and p. We know that $d_i^{\frac{p-1}{3}} \pmod p$ has order 1 or 3 which

is equivalent to d_i being a cube or not. We have therefore Algorithm 1 named `OneGoodPrime` where we make use of two functions : `CheckCubeCondition` which has been explained and `DrawPrime` which corresponds to the way we select the candidates for the prime numbers. One can follow [4] and generate a random prime number in a range given as argument. We could also generate a random prime first and then draw the next prime.

Algorithm 1 Finding a good prime for a sequence $\underline{d}$ and a condition sequence C.

Require: A reduced sequence $(d_1, \ldots, d_n)$ and $C = (c_1, \ldots, c_n) \in \{0,1\}^n$
Ensure: A prime p which does not divide any of the d_i's and such that for all $i \in \{0, n\}$ we have : (d_i is a cube modulo p) $= c_i$.

$b \leftarrow$ `false`
while $b =$ `false` **do**
  $p \leftarrow$ `DrawPrime`
  while $p \pmod 3 \equiv 2$ **do**
    $p \leftarrow$ `DrawPrime`
  end while
  $b \leftarrow \bigwedge_{i=1}^{n}$ `CheckCubeCondition`(d_i, p, c_i) ▷ logical AND
end while
return p

For a random prime $p \equiv 1 \pmod 3$ the probability that the ith cube condition is true is equal to $\frac{2}{3}$ if $c_i = 0$ and $\frac{1}{3}$ if $c_i = 1$. Therefore if we note $\mathrm{Hw}(C)$ for the Hamming weight of C we have

$$\mathbb{P}(\bigwedge_{i=1}^{n} \texttt{CheckCubeCondition}(d_i, p, c_i) = \texttt{true}) = (\frac{1}{3})^{\mathrm{Hw}(C)} \times (\frac{2}{3})^{n-\mathrm{Hw}(C)}.$$

In average the algorithm will try $\frac{3^n}{2^{n-\mathrm{Hw(C)}}}$ primes before finding one verifying the condition sequence C. In particular the probability that all d_i's are cubes in $\mathbb{F}_p$ is $\frac{1}{3^n}$ and the algorithm will try 3^n primes before finding one verifying the condition sequence $C = (1, \ldots, 1)$.

Complexity : We obtain a complexity essentially in $O(N)$.

4.2 Detecting cubes

One important procedure in [4] consists in finding non trivial products of a given family of K^* which are squares. In the case of multicubic fields we need to detect cubes. Let us describe how it is done. We consider $U = \langle u_1, \dots, u_m \rangle$ a subgroup of K^*. We need to compute non trivial "cubic characters" from U to $\mathbb{F}_3$. To do so we will use several primes p to create non trivial morphisms from $\mathbb{Z}[d_1^{\frac{1}{3}}, \dots, d_n^{\frac{1}{3}}]$ to $\mathbb{F}_p$ which can be extended multiplicatively to U.

In order to create morphisms from $\mathbb{Z}[d_1^{\frac{1}{3}}, \dots, d_n^{\frac{1}{3}}]$ to some $\mathbb{F}_p$ we need to find a p such that every d_i is a cube modulo p i.e. verifying the cubic conditions $C = (1, \dots, 1)$. This is done with Algorithm 1. Such a morphism can be extended to all elements of K whose denominators are not divided by p. In order this morphism to be defined on U it is sufficient that p does not divide the denominators of the u_i's. We then verify that the embeddings of the u_i's are not zero so that the morphism restricted to U is not trivial.

Now suppose a prime p as been selected. Note ϕ_p the morphism it induces as explained before. We want to create a character i.e. a group morphism $U \longrightarrow \mathbb{F}_3$ in order to detect non trivial cubes in U. Similarly to [4] we use the cubic character $\mathbb{F}_p^* \longrightarrow \mathbb{F}_3$ which corresponds to the natural morphism $\mathbb{F}_p^* \twoheadrightarrow \frac{\mathbb{F}_p^*}{(\mathbb{F}_p^*)^3}$. Remark that p needs to be congruent to 1 modulo 3 because we are looking for a non trivial morphism. Note $\zeta_{3,p}$ a fundamental root of unity in $\mathbb{F}_p$. Let us now describe how this morphism can be realised. For any y in $\mathbb{F}_p$ we know that $y^{\frac{p-1}{3}}$ is a cube root of unity in $\mathbb{F}_p$. Therefore it can be expressed as $\zeta_{3,p}^{\lambda_y}$ with $\lambda_y = \log_{\zeta_{3,p},}(y) \in [\![0, 2]\!]$. We can see that the canonical morphism can be written

$$\begin{array}{rcl} \mathbb{F}_p^* & \twoheadrightarrow & \frac{\mathbb{F}_p^*}{(\mathbb{F}_3^*)^3} \\ y & \mapsto & \log_{\zeta_{3,p},}(y). \end{array}$$

As a cubic character induced by p we will therefore consider

$$\begin{array}{rccl} \chi_p : & U & \longrightarrow & \frac{\mathbb{F}_p^*}{(\mathbb{F}_3^*)^3} \\ & u & \longmapsto & \log_{\zeta_{3,p},}(\phi_p(u)). \end{array}$$

Remark that if u is a cube in $\mathcal{O}_K^\times$ then $\phi_p(u)$ is also a cube in $\mathbb{F}_p$ but the opposite is not true in general. So if u is a cube then $u \in \ker \chi_p$. That is why in order to properly detect non trivial cubes in U we need – as in the multiquadratics for

detecting squares – to use several primes. First remark that the character induces a morphism

$$\chi_p : \quad \frac{U}{U\cap(K^*)^3} \quad \longrightarrow \quad \mathbb{F}_3 \ .$$

The group $\frac{U}{U\cap(K^*)^3}$ is isomorphic to some $(\frac{\mathbb{Z}}{3\mathbb{Z}})^{m'}$ with $m' \leqslant r$. Moreover it can be seen as $\mathbb{F}_3$-vector space. Following [8] as in [4] if we consider characters χ_p as a uniformly distributed element of the dual of this vector space we can hope that drawing sufficiently enough of them will detect cubes. We can adapt Lemma 8.1 of [8] to $\mathbb{F}_3$-vector spaces to say that $m' + s$ uniformly drawn primes generate the dual of $\frac{U}{U\cap(K^*)^3}$ with probability at least $1 - 3^{-s}$. Therefore by choosing s large enough the cubic characters $\chi_{p_1}, \ldots, \chi_{p_{m+s}}$ would generate the dual with high probability and the intersection $\bigcap_{i=1}^{s} \ker \chi_{p_i}$ would be the orthogonal of the dual i.e. $U \cap (K^*)^3$. This allows us to have Algorithm 2 which returns a matrix of exponents expressing a generating set of non trivial cubes in $U \cap (K^*)^3$. The fact that the exponent are non trivial means that the cubes are not in U^3 so generate $\frac{U\cap(K^*)^3}{U^3}$.

Algorithm 2 Compute non trivial cubes of a subgroup of K^* – `CubeKernel`

Require: $U = \langle u_1, \ldots, u_m\rangle$ a subgroup of K^*
Ensure: $\underline{\lambda_1}, \ldots, \underline{\lambda_r} \in [\![0, 2]\!]^m$ such that $\prod_{i=1}^{m} u_i^{\lambda_{j,i}}$ is a cube for all $j \in [\![1, r]\!]$
1: Generate sufficiently enough cubic characters $\chi_{p_1}, \ldots, \chi_{p_{m+s}}$
2: $M \leftarrow [\chi_{p_j}(u_i)]_{i,j} \in M_{m,m+s}(\mathbb{F}_3)$
3: $N \leftarrow \ker(M)$ ▷ Left Kernel in $\mathbb{F}_3$
4: **return** N as a matrix in $\mathbb{Z}$

As mentioned before with s large enough we have a very low probability of yielding an exponent vector $\underline{\lambda}$ such that $\prod_{i=1}^{m} u_i^{\lambda_i}$ is not a cube. Like the authors of [4] we never encountered such a case.

Complexity : Generating a cubic character consists in applying Algorithm 1 to find a prime p and reducing the elements $u_1, \ldots, u_m$ modulo p to verify that morphism ϕ_p is defined and non zero on $U = \langle u_1, \ldots, u_m\rangle$. In order to calculate $\phi_p(u_i)$ we need to compute the cube roots of $d_1, \ldots, d_n$, reduce the coefficients of u_i and compute a sum modulo p. All of this can be done in $O(NB)$ with B an upper bound on the number of bits of the coefficients of any of the u_i. This is mainly due to reduction of u_i modulo p. The computation of $m + s$ characters is therefore in $O((m + s)NB)$. We will consider $m + s$ to be equivalent to N asymptotically so we obtain $O(N^2B)$. Finally the computation of the kernel of a matrix of size N over $\mathbb{F}_3$ has complexity N^3 so the complexity of Algorithm 2 is $O(N^3 + N^2B)$.

4.3 Computing cube roots

Consider the following problem : given an element y in a multicubic field $K = \mathbb{Q}(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}})$ which is a cube, compute its cube root. This problem is of particular importance. Indeed recall the general procedure to compute $\mathcal{O}_K^\times$.

(i) Recursively compute the units of four subfields K_1, K_2, K_3, K_4 such that $(\mathcal{O}_K^\times)^3 < U = \prod_{i=1}^4 \mathcal{O}_{K_i^\times} < \mathcal{O}_K^\times$;

(ii) Find non trivial cubes of U;

(iii) Calculate their cube roots.

We proved in Proposition 3.18 that the first step holds. We saw in 4.2 how to find non trivial cubes. The third step is the last to address. In [4] the authors showed how to compute efficiently square roots in multiquadratic fields using only a few polynomial expressions.

In a multiquadratic field $E = F(\sqrt{d})$ – with F a subfield of E – consider $h = g^2$. Then if we write $h = h_0 + \sqrt{d}h_1$ and $h = g_0 + \sqrt{d}g_1$ we have $h_0 = g_0^2 + dg_1^2$ and $h_1 = 2g_0g_1$. Moreover the algebraic norm $\mathrm{N}_{E/F}(h) = \mathrm{N}_{E/F}(g)^3$ is an element of E. So if we can compute square-roots efficiently in F we can know $\mathrm{N}_{E/F}(g) = g_0^2 - g_1^2 d$ and then retrieve g_0 and g_1 using h_0 and h_1. This require to compute one more square-root in F. The only obstacle in this procedure is the sign since a square-root may have two distinct solutions. Doing such errors at each level of the recursive process can lead to an exponential number of possibilities to verify. However the authors of [4] overcame this difficulty and provided an efficient recursive algorithm to compute square-roots in multiquadratic fields.

The problem of sign does not appear with cube roots. However the polynomial equations are more complex. Write $x = x_0 + x_1 d_n^{\frac{1}{3}} + x_2 d_n^{\frac{2}{3}}$ and $y = y_0 + y_1 d_n^{\frac{1}{3}} + y_2 d_n^{\frac{2}{3}}$. Then we have :

$$\begin{cases} y_0 = x_0^3 + x_1^3 d_n + x_2^3 d_n^2 + 6x_0x_1x_2d_n \\ y_1 = 3(x_0^2x_1 + x_1^2x_2d_n + x_2^2x_0d_n) \\ y_2 = 3(x_0^2x_2 + x_1^2x_0 + x_2^2x_1d_n) \\ \mathrm{N}_{K/L}(x) = x_0^3 + x_1^3d_n + x_2^3d_n^2 - 3x_0x_1x_2d_n. \end{cases}$$

There is no straightforward way of transforming these equations into a cube that we could take advantage of. Therefore we chose to use a real embedding and a LLL reduction. This allows to progressively increase the needed precision and

save the real lattice used to recover the coefficients. Let us now describe the procedures composing this algorithm. We use a function called `RealBasisEmbedding` which creates the vector of the basis elements of the multicubic field K computed in $\mathbb{R}$ to a given precision. Then we can create the matrix representing the basis as a lattice. Note $\mathbf{v}$ the row vector $\texttt{RealBasisEmbedding}((d_1,\ldots,d_n),l)$. We choose as a "real basis matrix" the following

$$\mathrm{LLL}\left(\left[\mathbf{v}|C\cdot\mathrm{Id}\right]\right)$$

where C is a coefficient chosen to avoid errors due to the precision. We typically used $C = \lfloor\frac{3^n}{12}\rceil$. Now if a basis lattice matrix has been computed for a given precision here how one can try to fasten the computation of a basis lattice matrix to a bigger precision. First note $\mathbf{v_{l_1}}$ and $\mathbf{v_{l_2}}$ the real basis vectors given up to two precisions $l_1 < l_2$. We can write

$$\mathrm{LLL}\left(\left[\mathbf{v_{l_1}}\mid C\cdot\mathrm{Id}\right]\right) = U\times\left[\mathbf{v_{l_1}}\mid C\cdot\mathrm{Id}\right]$$

with U being a unitary matrix. If we save this unitary operator we can then first calculate

$$U\times\left[\mathbf{v_{l_2}}\mid C\cdot\mathrm{Id}\right]$$

then apply the LLL algorithm to finally reduce the lattice. This reduction is done by multiplying by a unitary operator V and the full reduction can be written

$$\mathrm{LLL}\left(\left[\mathbf{v_{l_2}}\mid C\cdot\mathrm{Id}\right]\right) = V\times U\times\left[\mathbf{v_{l_2}}\mid C\cdot\mathrm{Id}\right].$$

Therefore we can now save $V\times U$ and use the same process if we need to actualise again the precision.

Algorithm 3 Compute a matrix representing the real embedding of the matrix of a multicubic field – `RealLattice`

Require: A LLL-reduced real lattice matrix of $(d_1,\ldots,d_n)$, a unitary operator U, a precision l

Ensure: A LLL-reduced real lattice matrix of $(d_1,\ldots,d_n)$ at precision l and the corresponding unitary operator

1: $\mathbf{v}\leftarrow\texttt{RealBasisEmbedding}(\underline{d},l)$
2: $M\leftarrow U\times\left(\left[\mathbf{v}|C\cdot\mathrm{Id}\right]\right)$
3: $L,V\leftarrow\mathrm{LLL}(M)$ ▷ $L=\mathrm{LLL}(M)=VM$
4: **return** $L, V\times U$

Now recall that we want to compute cube roots. Given L a real lattice matrix for K here how we can expect to do so. Consider $y \in K$ as before. First compute x up to precision l in $\mathbb{R}$. To do so, compute y up to a large enough precision so that the cube root in $\mathbb{R}$ does not create errors. Note `RealEmbedding` this procedure and the returned value x_l. Then create the row vector $\mathbf{x} = [x_l \mid \mathbf{0} \mid B]$ with B being a coefficient larger than the maximum euclidean norm of the rows of L. we can then build the matrix

$$\left[\begin{array}{c|c} L & \mathbf{0} \\ \hline \mathbf{x} & \end{array}\right]$$

and apply a LLL algorithm to it. This can be seen as overall reduction of

$$\left[\begin{array}{c|c|c} \mathbf{v} & C \times \mathrm{Id} & \mathbf{0} \\ \hline x_l & \mathbf{0} & B \end{array}\right]$$

which would reduce the last vector with respect to the real basis lattice. Considering the shape of the last matrix we expect the central part of the last row vector to be the vector of coefficients of Cx in K. We note `CubeRootCandidate` this procedure.

Algorithm 4 Compute a candidate for a cube root in a multicubic field – `CubeRootCandidate`

Require: An cube element $y = x^3$ in a multicubic field K of dimension N, a precision l and a real basis lattice of K for precision l

Ensure: x' a candidate for x

1: $x_l \leftarrow \texttt{RealEmbedding}(y, l)$

2: $\mathbf{x} \leftarrow [x_l \mid \mathbf{0} \mid B]$

3: $M \leftarrow \mathrm{LLL}\left[\begin{array}{c|c} L & \mathbf{0} \\ \hline \mathbf{x} & \end{array}\right]$

4: $x' \leftarrow (M_{N,2}, \ldots, M_{N,N+1})$

5: **return** x'

Once we have this candidate we can check its validity by computing its cube and looking whether it is y or not. If not we can increase the precision and find another candidate. We can evaluate the needed precision with a function `PrecisionEvaluation`. This function takes y and n the number of primes defining K in argument. Experiments suggest that for a given degree the precision is linear in $\log(\|y\|_2)$. However the slope increases with n and seems to be multiplied by a coefficient between 2 and 3. We have chosen to use 3 so the slope for

K of dimension 3^n is 3^{n-1}.

Complexity : The algorithm consists essentially in applying several LLL with coefficients of size given by `PrecisionEvaluation`. Note B an upper bound on the bit size of coefficients of y. Then the complexity of `CubeRootCandidate` would be $O(N^5B^2)$. We might have to increase the precision but experimentally it is only done a few times. We expect the complexity to stay in $O(N^5B^2)$.

Algorithm 5 Computing a cube root in a multicubic field – `MC_CubeRoot`

Require: An cube element $y = x^3$ in a multicubic field $K = \mathbb{Q}(d_1^{\frac{1}{3}}, \ldots, d_n^{\frac{1}{3}})$
Ensure: The cube root x of y
1: $l \leftarrow$ `PrecisionEvaluation`(y, n)
2: $L, U \leftarrow$ `RealLattice`$(\underline{d}, l)$
3: $x' \leftarrow$ `CubeRootCandidate`(y, l, L)
4: **while** $(x')^3 \neq y$ **do**
5: $\quad l \leftarrow 2l$
6: $\quad L, U \leftarrow$ `RealLattice`$(\underline{d}, l, L, U)$
7: $\quad x' \leftarrow$ `CubeRootCandidate`(y, l, L)
8: **end while**
9: **return** x'

4.4 Computing units

We will describe in this section the algorithm used to compute the units of a multicubic field. As mentioned before we will mainly proceed as in the multiquadratic case. We will recursively compute the units of chosen subfields and then retrieve the whole group by detecting cubes and computing their cube root. Therefore the algorithm can be seen as computing subgroup $\mathrm{MCU}(K)$ and then deduce $\mathcal{O}_K^\times$ only by doing products and cube root in successive subfields. Moreover we represent any unit at each step of the algorithm for K as $(u, \mathrm{ApproxLog}_K(u))$ even if we are computing the units of a subfield. This can be done easily because we can compute the approximate logarithm of any element of $\mathrm{MCU}(K)$ by a function `CubicApproxLog`. Then we compute the approximate logarithm of other units by doing only sums and divisions by 3. Since the lattice generated by the multicubic units in the Log-unit representation has an orthogonal basis we compute $\mathrm{ApproxLog}_K(\mathcal{O}_K^\times)$ starting by an orthogonal basis of a sublattice and then only adding and dividing by three these vectors.

In Algorithm 6 we use of several sub-algorithms namely

(i) `CubicUnitGroup`;

(ii) `BasisFromGeneratingSet`;

(iii) `UnitsFromCubes`.

The first one is the classical unit group algorithm implemented in Magma. We apply it only to compute the multicubic units. The last two algorithms are adapted from [4] in the multicubic case. `BasisFromGeneratingSet` takes into argument a generating set of a subgroup of $\mathcal{O}_K^\times$ and returns a basis. It is done by reducing the corresponding generating family in the Log_K-representation. If the subgroup U is given by a generating family $(u_1, \ldots, u_m)$ we apply a LLL algorithm on the matrix

$$\left[\begin{array}{ccc|c} 1 & & & 2^l \times \mathrm{ApproxLog}_K(u_1, l) \\ & 1 & & 2^l \times \mathrm{ApproxLog}_K(u_2, l) \\ & & \ddots & \vdots \\ & & 1 & 2^l \times \mathrm{ApproxLog}_K(u_m, l) \end{array}\right]$$

to reduce the matrix of the $\mathrm{ApproxLog}_K(u_i, l)$ and recover as well V the unitary transform. We therefore obtain a basis of $\mathrm{ApproxLog_K}(U)$ and can compute the corresponding elements of K by using V. The stretched Identity matrix allows to recover a matrix V with relatively small relations in a way similar to what did the authors of [4].

The function `UnitsFromCubes` computes a generating set of $\mathcal{O}_K^\times$ given a generating set of a subgroup U such that $(\mathcal{O}_K^\times)^3 < U < \mathcal{O}_K^\times$. Let us write $(u_1, \ldots, u_m)$ a generating set of U. The algorithm computes exponent vectors using the `CubeKernel` algorithm and obtains a basis of non trivial cubes in U. Then it computes their cube roots $(v_1, \ldots, v_r)$ using `MC_CubeRoot` and returns the family $(u_1, \ldots, u_s, v_1, \ldots, v_r)$. Following [4] it is not hard to see that the returned family generates the whole group $\mathcal{O}_K^\times$. Remark that the approximate logarithm of the resulting new vectors can be computed by sums and division by three.

Complexity : The complexity of the algorithm is $\mathrm{Poly}(N, B)$ where B is an upper-bound on the bit-size of the elements we are computing.

4.5 Principal Ideal Problem

The main goal is to find a short generator of a principal ideal of K. This problem is usually done by finding a generator first and finding a short vector using the Log-unit lattice. Since we can compute the unit group we "only" need to find a generator of an ideal. An ideal I can be described by several representations.

- an integral basis

Algorithm 6 Compute the unit group of a multicubic field – `MC_Units`

Require: A reduced sequence $(d_1, \ldots, d_n)$ defining a multicubic field, a precision factor l.

Ensure: A basis $\{u_1, \ldots, u_r\}$ of the torsion-free part of unit group $\frac{\mathcal{O}_K^\times}{\langle \pm 1 \rangle}$

1: **if** $n = 1$ **then**
2: $u \leftarrow \texttt{CubicUnitGroup}(K)$
3: **return** $(u, \texttt{CubicApproxLog}(u, l))$
4: **else**
5: Choose v, w two independent elements of $H(\widetilde{K})$ and recursively compute basis of $U = \mathcal{O}_{K_v}^\times \mathcal{O}_{K_w}^\times \mathcal{O}_{K_{vw}}^\times \mathcal{O}_{K_{v^2w}}^\times$
6: $V \leftarrow \texttt{UnitsFromCubes}(U)$ ▷ Algorithm 2 and Algorithm 5
7: $U \leftarrow \texttt{BasisFromGeneratingSet}(\langle U, V \rangle)$
8: **return** U
9: **end if**

- the two element representations that is used to fasten ideal based cryptosystem such as in [9, 19]

We consider the more basic situation which is the first one. It has the advantage of being more general. However it is a much bigger representation and operations may be much slower. For example one fundamental operation on ideals for the PIP algorithm in multiquadratic fields and multicubic fields is the relative norm computation. Given an ideal I of a number field K and L a subfield of K the relative norm of I with respect to K/L is the ideal of L generated by the norms $\mathrm{N}_{K/L}(x)$ for $x \in K$. If K/L is a Galois extension then we have

$$\mathrm{N}_{K/L}(I) = \prod_{\sigma \in \mathrm{Gal}(K/L)} \sigma(I).$$

This is for example the case if K and L are multiquadratic fields. Multicubic fields are not Galois however the situation is pretty similar. Instead of computing the product over $\mathrm{Gal}(K/L)$ we compute it over the complex embeddings which are the identity when restricted to L and the product is done in $\widetilde{K}$. Then one way of computing $\mathrm{N}_{K/L}(I)$ given an integral basis $(b_1, \ldots, b_n)$ is to calculate all of the products $\prod_\sigma \sigma(b_\sigma)$ with $b_\sigma \in \{b_1, \ldots, b_n\}$, express them in basis of $\mathcal{O}_K$, then reduce the matrix obtained by calculating its Hermite Normal Form (HNF) for example and finally intersect with F. We can see that this requires to compute $[K : L] - 1$ product of ideals of K. The complexity of the HNF is linear in the degree of K however it is still quite slow. In Algorithm 7 the fields considered

are K a multicubic field of dimension 3^n and L a multicubic field of dimension 3^{n-1}. Therefore K/L is a degree 3 extension and the embeddings in $\mathrm{Hom}(K,\mathbb{C})$ which are the identity on L are $\{1,\sigma^{(\underline{\beta})},\sigma^{(2\underline{\beta})}\}$ for a given $\underline{\beta}$. Therefore we need to compute two ideal products which are done by reducing matrices of 3^{2n} vectors in a HNF with 3^n rows.

The PIP algorithm in multicubic fields is similar to the one for multiquadratic fields as their algebraic structure are almost the same. Given an ideal I we compute recursively a generator for each of four norm ideals in subfields, combine them to yield a generator h of I^3 and finally find $\epsilon\in\mathcal{O}_K^\times$ such that $h\epsilon$ is a cube to compute $a=(h\epsilon)^{\frac{1}{3}}$. Like the units computation this relies on the structure of the field and Proposition 3.18. Indeed let us write $I=g\mathcal{O}_K^\times$. Then we know that we have

$$g^3=\frac{\mathrm{N}_{\tilde{K}/\tilde{K}_u}(g)\mathrm{N}_{\tilde{K}/\tilde{K}_v}(g)\mathrm{N}_{\tilde{K}/\tilde{K}_{uv}}(g)}{\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2v}}(u(g)\cdot uv(g))}=\frac{\mathrm{N}_{\tilde{K}/\tilde{K}_u}(g)\mathrm{N}_{\tilde{K}/\tilde{K}_v}(g)\mathrm{N}_{\tilde{K}/\tilde{K}_{uv}}(g)}{u(\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2v}}(g))\cdot uv(\mathrm{N}_{\tilde{K}/\tilde{K}_{u^2v}}(g))}$$

for any independent u,v in $H(\widetilde{K})$. For clarity of the writing note N_1, N_2, N_3, N_4 the four considered relative norm operators and g_1, g_2, g_3,g_4 the four relative norm elements such that

$$g^3=\frac{g_1g_2g_3}{u(g_4)uv(g_4)}.$$

Then for all $i\in[\![1,4]\!]$, g_i is a generator of the principal ideal $\mathrm{N}_i(I)$. If h_i is of $\mathrm{N}_i(I)$ then we have $h_i=g_i\epsilon_i$ with ϵ_i a unit of the fixed subfield so of K. Then we have

$$\frac{h_1h_2h_3}{u(h_4)uv(h_4)}=\frac{g_1g_2g_3}{u(g_4)uv(g_4)}\times\frac{\epsilon_1\epsilon_2\epsilon_3}{u(\epsilon_4)uv(\epsilon_4)}=g^3\frac{\epsilon_1\epsilon_2\epsilon_3}{u(\epsilon_4)uv(\epsilon_4)}.$$

Remark that the map u and v are complex valued maps so it is not direct that $u(\epsilon_4)uv(\epsilon_4)$ is a unit of K. However as in Proposition 3.18 the above formula for g^3 can be applied to any element of K and the denominator is a real element. The product $u(\epsilon_4)uv(\epsilon_4)$ is therefore real and is a unit of K. Finally if we find a unit ϵ such that $h\epsilon$ is a cube we can retrieve $g\eta$ by computing $(h\epsilon)^{\frac{1}{3}}$. This is similar to the situation of multiquadratic fields, the difference being that we have to be careful when manipulating morphisms. Indeed multiquadratic fields are Galois so their complex embeddings are in fact automorphisms and there is no trouble of elements being sent outside of the field.

Once we have calculated the norm ideals, retrieved one generator for each of them and computed the element h as stated before we will find the unit ϵ the same

way we find non trivial cubes in the algorithm for units. Note U the subgroup of K^* generated by h and $\mathcal{O}_K^\times = \langle u_1, \ldots, u_m \rangle$. If we follow [4] we compute enough good cubic characters as in Algorithm 2 then we store separately $M = [\chi_{p_j}(u_i)]_{i,j}$ and the row vector $c_h = [\chi_j(h)]_j$. Then a solution e of $eM = -c_h$ would yield the desired vector of exponents. Alternatively we can compute $\texttt{CubeKernel}(U)$. Then at least one of the kernel vectors e has a non zero coefficient corresponding to h. If it is 1 then e is such that $h\prod_{i=1}^m u^{e_i}$ is a cube. If it is 2 then $h^2\prod_{i=1}^m u^{e_i}$ is a cube. But $2e$ – when taken in $\mathbb{F}_3$ – is in the kernel too and $h\prod_{i=1}^m u^{2e_i}$ is a cube.

Remark that in Algorithm 7 we use the classical algorithm `CubicPrincipalIdeal` to solve the PIP in the cubic subfields as we used `CubicUnitGroup` in Algorithm 6. Moreover it is not precised but we compute the approximate logarithm of the retrieved generators of cubic fields. Then we compute the logarithm of the final generator of I only doing sums and division by three.

Algorithm 7 Solve the PIP – `MC_PIP`

Require: A principal ideal I of a multicubic field K
Ensure: A generator g of I
1: **if** $n = 1$ **then**
2: $g \leftarrow \texttt{CubicPrincipalIdeal}(K)$
3: **return** g
4: **else**
5: Choose u, v two indepedent elements of $H(\widetilde{K})$ and recursively compute generators h_1, h_2, h_3, h_4 of $\mathrm{N}_{K_u}(I), \mathrm{N}_{K_v}(I), \mathrm{N}_{K_{uv}}(I), \mathrm{N}_{K_{u^2v}}(I)$
6: $g \leftarrow \texttt{GeneratorFromCube}\left(\dfrac{h_1h_2h_3}{u(h_4)uv(h_4)}\right)$
7: **return** g
8: **end if**

4.6 Shortening of the generator

Now the problem that we want to solve is the SPIP. Assume that we know that I has a short generator. We considered generators with coefficients drawn uniformly in $\{-1, 0, 1\}$.
Once a generator h of the ideal I is found, one can choose from several techniques to try to recover the secret g or a short enough generator. In [12] the authors used the dual lattice. They considered a subgroup C of $\mathcal{O}_K^\times$ easily computable such that $[\mathcal{O}_K^\times : C]$ is close to 1. They gave a bound on the dual vectors of $\mathrm{Log}_K(C)$ so they proved decryption could be done in this sublattice. The very small gap between

$\mathrm{Log}_K(C)$ and $\mathrm{Log}_K(\mathcal{O}_K^\times)$ allowed a full decryption. In the case of multiquadratic and multicubic fields there is a good subgroup with a perfect decryption situation, namely the multiquadratic units and the multicubic units. They form orthogonal sublattices of the Log-unit of their respective fields. However in both cases the gap between the unit groups and these subgroups is too large to try the same strategy as with cyclotomic fields. However there are efficient algorithms to compute the units and solve the PIP of a wide range of multiquadratic fields so the full Log-unit lattice can be computed efficiently. In the case of multicubic fields we are less efficient but we still manage to compute units in reasonable time for some cases. Finally even if we can compute a basis of the Log-unit lattice it is not certain that we can efficiently recover short generators. This will depend on the geometrical properties of the basis.

In [4] the shortening procedure is a rounding. The authors considered the vector $\mathrm{Log}(h) = \mathrm{Log}(g) + \mathrm{Log}(u)$ expressed in the basis $\mathrm{Log}(\mathcal{O}_K^\times)$ and rounded its coefficient to the nearest integer. The expression of $\mathrm{Log}(g)$ in the basis of $\mathrm{Log}(\mathcal{O}_K^\times$ is obtained by using the inverse of the basis matrix. If the basis is good enough then the rounding is expected to be exact. In the case of multicubic fields we cannot use this rounding method since the Log-unit lattice is not a full rank lattice in its ambient vector space. Instead we used a LLL decryption method. Note L the matrix of the approximate Log-embedding of the units computed, h the vector found by the PIP algorithm and B an upper bound of the norm of the vectors of L. Then consider – similarly to the cube root procedure – the matrix

$$\left[\begin{array}{c|c} L & \mathbf{0} \\ \hline \mathrm{ApproxLog}_K(h) & B \end{array}\right] = \left[\begin{array}{c|c} \mathrm{ApproxLog}_K(u_1) & 0 \\ \mathrm{ApproxLog}_K(u_2) & 0 \\ \vdots & \vdots \\ \mathrm{ApproxLog}_K(u_m) & 0 \\ \hline \mathrm{ApproxLog}_K(h) & B \end{array}\right]$$

and reduce it with a LLL algorithm. If $\mathrm{Log}_K(g)$ is short respectively to the Log-unit lattice this is expected to reduce the last row to the Log-embedding of the closest generator. If we compute the unitary operator corresponding to this LLL reduction we can retrieve u and g.

4.7 Experiments and Results

We present here the data we collected from computations. We considered multicubic fields defined by prime sequences $(p_1, \ldots, p_n)$. We did computations essen-

Algorithm 8 Shorten a given generator of an ideal – `ShortGen`

Require: A generator h of a principal ideal I, $\mathcal{O}_K^\times = \langle u_1, \ldots, u_m \rangle$
Ensure: A candidate g for a short generator.
1: $L \leftarrow [\text{ApproxLog}_K(u_i)]_{i \in [\![1,m]\!]}$
2: $B \leftarrow \max\{\|L[i]\|_2 \mid i \in [\![1,m]\!]\}$
3: $M, V \leftarrow \text{LLL}(\left[\begin{array}{c|c} L & \mathbf{0} \\ \hline \text{ApproxLog}_K(h) & B \end{array}\right])$
4: **return** $h^{V_{m+1,1}} \times \prod_{i=2}^{m+1} u_{i-1}^{V_{m+1,i}}$

tially for multicubic fields defined by n primes with n equal to 2, 3 and 4. These correspond to fields of dimension 9, 27 and 81. We did some computations for fields defined by 5 primes i.e. with dimension 243.

Computing $\mathcal{O}_K^\times$

Recall that we compute units of a multicubic field K recursively and at each step the main procedure is `CubeRoot` presented in Algorithm 5. The efficiency of the overall algorithm is strongly related to the efficiency of `CubeRoot` and tends to be dominated by it. This is illustrated by the times computed in Table 1. In Figure 1 we can find the times for $n = 2$ printed. It illustrates well the correlation between the time taken to compute the units and the time taken to compute cube roots. If we analyse the function `CubeRoot` we can see that it depends on the dimension, the sequence defining the field K and the norm of the elements it is given. Therefore together with the times we computed the number of cube roots computed by the last call to `CubeRoot` in Algorithm 6 and the average of the logarithm of their norms. We can understand from these data why the algorithm does not scale as the algorithm in [4]. The norm of the elements from which we compute cube root seems to scale poorly and we have to compute more cube roots when the degree increases. Moreover the efficiency seems to decrease quickly with increasing primes.

Complexity : An analysis of the norm of the units that the algorithm compute cube roots of can be found in Appendix and gives a bound essentially polynomial in $N^{\frac{n}{2}} \prod_{i=1}^{n} d_i$. This gives a complexity for the overall algorithm essentially in $O(\text{Poly}(N^{\frac{n}{2}} \prod_{i=1}^{n} d_i))$.

We obtained better results that the standard algorithm implemented in Magma.

First prime	2	3	5	7	11	13	17	19	23	29
$\mathcal{O}_K^\times$ (times in s)	0.260	0.260	0.260	0.270	0.290	0.350	0.330	0.360	0.480	0.320
`CubeRoot` (times in s)	0.010	0.010	0.010	0.010	0.000	0.050	0.060	0.070	0.180	0.010
# cube roots	3	3	1	1	1	1	1	2	3	1
Average logarithm of the Norm of cubes	3	18	31	45	24	215	270	175	162	70

(a) $n = 2$

First prime	2	3	5	7	11	13	17	19	23	29
$\mathcal{O}_K^\times$ (times in s)	2.110	2.250	2.490	4.500	2.780	18.780	4.060	24.810	9.230	24.420
`CubeRoot` (times in s)	0.060	0.180	0.350	2.310	0.350	15.980	1.020	16.540	5.950	16.490
# cube roots	3	4	3	4	2	5	4	5	4	3
Average logarithm of the Norm of cubes	13	29	46	127	83	404	112	398	313	781

(b) $n = 3$

First prime	2	3	5	7	11	13	17
$\mathcal{O}_K^\times$ (times in s)	39.670	71.160	157.460	873.670	7479.250	9862.540	29308.850
`CubeRoot` (times in s)	19.220	47.270	130.240	832.780	7370.470	9271.600	28425.140
# cube roots	14	12	10	11	11	11	13
Average logarithm of the Norm of cubes	29	75	168	533	1090	2178	3295

(c) $n = 4$

First prime	2	3	5
$\mathcal{O}_K^\times$ (times in s)	16026.410	87701.680	566029.130
`CubeRoot` (times in s)	15246.560	85036.150	562127.470
# cube roots	36	36	48
Average logarithm of the Norm of cubes	63	199	531

(d) $n = 5$

Table 1. Times and data for Algorithm 5 and 6 for number fields defined by consecutive primes for $n = 2$, 3, 4 and 5

For example we can see in Table 2 the times to compute units for consecutive primes and $n = 2$. We can see that the size of primes has a strong impact. It took 2540.490 seconds to compute the units of the field defined by $(2, 3, 5)$ and did not retrieve the units of the field defined by $(3, 5, 7)$ after 34 hours.

Retrieving a short generator

For each given size of keys (except 243) we chose two sequences. The first is the n consecutive primes and the second follows an arithmetic progression i.e. p_1 is fixed and the p_{k+1} is $\texttt{NextPrime}(p_k + 4)$ for each k.

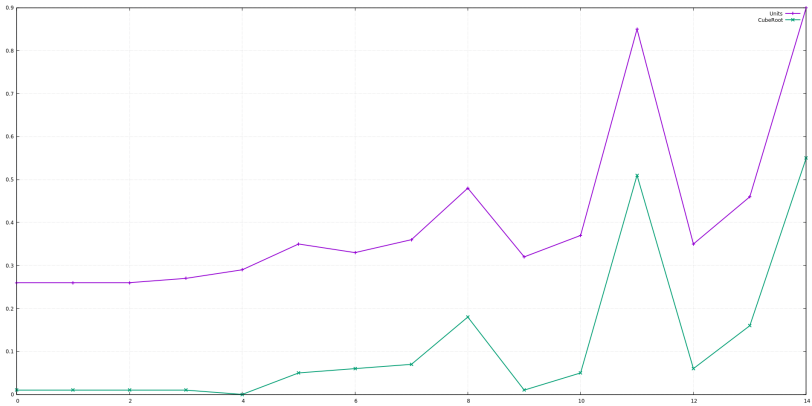

Figure 1. Times to compute $\mathcal{O}_K^\times$ and the cube roots for fields defined by consecutive primes and $n = 2$

First prime	2	3	5	7	11	13	17	19	23	29
$\mathcal{O}_K^\times$ (times in s)	0.190	0.200	0.240	0.520	2.190	8.510	0.700	61.480	200.540	503.640

Table 2. Times to compute $\mathcal{O}_K^\times$ with `UnitGroup` of Magma for $n = 2$

As mentioned before we considered keys as vectors of coefficients drawn uniformly at random in $\{-1, 0, 1\}$. The data are presented in Table 3. For each n and each progression the first row is the percentage of exact decoding and the second is the percentage of shorter generators – exact of strictly shorter generators – retrieved.

We can remark that the probability of success seems to converge to 1 as the primes of the defining sequence increase. The probability of failure is particularly big when the smaller primes are in the sequence, especially two. The same phenomenon were noticed in [4]. Moreover we can see that the rate of generators retrieved which were strictly shorter than the key follow the inverse pattern. It is quite high compared to the rate of retrieved key when the latest is low and $n = 2$ and tends to 0 otherwise. For the multicubic field defined by the sequence $(2, 3, 5, 7, 11)$ we retrieved exactly 74.02% of the keys and no shorter generator and for the field defined by $(3, 5, 7, 11, 13)$ we retrieved exactly all of the keys.

These results tend to show that multicubic fields should not be used to build cryptosystems. Even if we are still too slow to attack dimensions of cryptographic

interest the results we obtained suggest that the we can easily recover short vectors using the Log-unit lattice. Finally in post-quantum perspective we have to think that computing $\mathcal{O}_K^\times$ and solving the PIP can be done efficiently. Therefore the fact that the algorithms presented in this paper are slow is not completely relevant. We are essentially interested in the quality of the basis of the Log-unit lattice.

First prime	2	3	5	7	11	13	17	19	23	29
Consecutive	35.20	90.80	98.40	98.20	100.0	100.0	99.70	99.80	100.0	100.0
	46.20	91.50	98.40	98.20	100.0	100.0	99.70	99.80	100.0	100.0
Arithmetic	69.90	95.10	98.60	97.40	100.0	99.80	100.0	99.80	100.0	100.0
	75.20	95.10	98.60	97.40	100.0	99.80	100.0	99.80	100.0	100.0

(a) $n = 2$

First prime	2	3	5	7	11	13	17	19	23	29
Consecutive	46.00	93.30	100.0	99.91	100.0	100.0	100.0	100.0	100.0	100.0
	46.40	93.30	100.0	99.91	100.0	100.0	100.0	100.0	100.0	100.0
Arithmetic	84.10	99.59	100.0	99.50	100.0	n/a	n/a	n/a	n/a	n/a
	84.10	99.59	100.0	99.50	100.0	n/a	n/a	n/a	n/a	n/a

(b) $n = 3$

First prime	2	3	5	7	11	13	17	19
Consecutive	64.20	99.91	100.0	100.0	100.0	100.0	100.0	100.0
	64.20	99.91	100.0	100.0	100.0	100.0	100.0	100.0
Arithmetic	95.00	100.0	100.0	100.0	100.0	n/a	n/a	n/a
	95.00	100.0	100.0	100.0	100.0	n/a	n/a	n/a

(c) $n = 4$

Table 3. Percentages of keys recovered for $n = 2$, 3 and 4

Bibliography

[1] L. Babai, On Lovász' lattice reduction and the nearest lattice point problem, *Combinatorica* 6(1), (1986)

[2] P. Barrucand, Quelques aspects de la théorie des corps cubiques, *Séminaire Delange-Pisot-Poitou. Théorie des nombres* 16, (1974-1975), 1–10.

[3] P. Barrucand, J. Loxton, H. C. Williams, Some explicit upper bounds on the class number and regulator of a cubic field with negative discriminant, *Pacific Journal of Mathematics*, 128(2), (1987), 209–222.

[4] J. Bauch, Daniel J. Bernstein, H. de Valence, T. Lange, C. Van Vredendaal, Short generators without quantum computers : The case of multiquadratics, in: *Advances in Cryptology, EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings.*, Lecture Notes in Computer Science 10210, LNCS Springer Verlag (2017), 27–59.

[5] K. Belabas, Topics in computational algebraic number theory, in: *Journal de théorie des nombres de Bordeaux*, 16(1), (2004), 19–63

[6] J.-F. Biasse, F. Song, Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, in: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 893-902

[7] W. Bosma, J. Cannon, and C. Playoust, The Magma algebra system. I. The user language, *J. Symbolic Comput.*, 24 (1997), 235âĂŞ265.

[8] J. P. Buhler, H. W. Lenstra, Jr., C. Pomerance, Factoring integers with the number field sieve, in: *The development of the number field sieve*, Lecture Notes in Math., 1554, Springer, Berlin (1993), 50-94

[9] P. Campbell, M. Groves, D. Shepherd, *Soliloquy : a cautionary tale*, ETSI 2nd Quantum-Safe Crypto Workshop, 2014, `http://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf`

[10] H. Cohen, *A Course in computational algebraic number theory*, Springer-Verlag, Berlin, Heidelberg, 1995

[11] H. Cohen, *Advanced Topics in Computational Number Theory*, Graduate Texts in Mathematics, Springer, New York, 1999.

[12] R. Cramer, L. Ducas, C. Peikert, O. Regev, Recovering Short Generators of Principal Ideals in Cyclotomic Rings, in: *Fischlin M., Coron JS. (eds) Advances in Cryptology*, Lecture Notes in Computer Science, vol 9666. Springer, Berlin, Heidelberg (2016)

[13] K. Eisenträger, S. Hallgren, A. Kitaev and F. Song, A Quantum Algorithm for Computing the Unit Group of an Arbitrary Degree Number Field, in: *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, ACM, New York (2014)

[14] C. Gentry, *A fully homomorphic encryption scheme*, Ph.D. thesis, Stanford University, 2009. `https://crypto.stanford.edu/craig`

[15] C. Gentry and S. Halevi, Implementing Gentry's fully-homomorphic encryption scheme, in: *Kenneth G. Paterson, editor, Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, Lecture Notes in Computer Science, 6632, Springer (2011), 129–148

[16] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen*, 261(4), (1982), 515–534.

[17] C. J. Parry, Class number formulae for bicubic fields, *Illinois J. Math.*, **1**, (1977),148–163

[18] P. Samuel, *Algebraic theory of numbers*, Paris, Hermann; Boston, Houghton Mifflin Co., 1970

[19] N. P. Smart, F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes, in: Phong Q. Nguyen and David Pointcheval,editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*,Lecture Notes in Computer Science, 6056, Springer (2010),420–443

Appendix : An upper bound on the norm of cubes

We will give an upper bound for the norm of the units that we compute the cube root of in Algorithm 6 depending on the degree and the sequence defining the multicubic field K. Note $(d_1, \ldots, d_n)$ this sequence. We assume that they are prime numbers. Let us fix some notations. Note v a unit in K found by `CubeKernel` in Algorithm 6. As stated before `MC_CubeRoot` consists essentially in several LLL procedures applied to matrices of size approximately N with coefficients of size evaluated by `PrecisionEvaluation`. In order to find the complexity of the LLL used to compute the cube root of u we need to bound $\texttt{PrecisionEvaluation}(v) = 3^n \times \log(\|v\|)$. We therefore need to evaluate $\|v\|$. We will consider here the infinity norm. For $x, y \in K$ we have

$$\|xy\|_\infty \leqslant \|x\|_\infty \|y\|_\infty \prod_{i=1}^{n}(1 + 2d_i).$$

Note $C(k, \underline{d}) = \sup\{\|v\|_\infty \mid v \in \mathcal{O}_L^\times, [K : L] = 3^{n-k}\}$ the maximal infinity norm of a unit of a subfield of K of dimension 3^k found in the course of Algorithm 6. Note $U = \langle u_1, \ldots, u_m\rangle$ the subgroup computed in Algorithm 6 before computing the procedure `CubeKernel`. We have $m \leqslant 4 \times \frac{3^{n-1}-1}{2}$ and $v = u_1^{e_1} \times \cdots \times u_m^{e_m}$ with $(e_1, \ldots, e_m) \in [\![0, 2]\!]^m$. Typically the previous exponent tuple has a fair proportion of 0 but we will assume we are in the worst case which is half of 1 and half of 2. Therefore v is calculated by a product of $3 \times (3^{n-1} - 1)$ terms so we have

$$\|v\|_\infty \leqslant C(n-1, d)^{3\times(3^{n-1}-1)} \times \left(\prod_{i=1}^{n}(1 + 2d_i)\right)^{3\times(3^{n-1}-1)-1}.$$

If we note $P(1 + 2\underline{d})$ the product we have

$$\log(\|\mathrm{v}\|_\infty) \leqslant 3(3^{n-1} - 1)\log(C(n-1, d)) + 3(3^{n-1} - 1)\log(P(1 + 2\underline{d})).$$

If we assume that $\|x\|_\infty \leqslant \|x^3\|_\infty \leqslant$ for any $x \in K$ we can write

$$\begin{aligned}\log(\|\mathrm{v}\|_\infty) \leqslant\ & 3(3^{n-1} - 1) \times 3(3^{n-2} - 1)\log(C(n-2, d)) \\ & + 3(3^{n-1} - 1) + (3(3^{n-1} - 1) \times 3(3^{n-2} - 1))\log(P(1 + 2\underline{d})).\end{aligned}$$

Then if we note $c_k = 3^k - 1$ we have

$$\begin{aligned}\log(\|\mathrm{v}\|_\infty) \leqslant\ & 3^{n-1}c_{n-1}c_{n-2}\ldots c_1\log(C(1, d)) \\ & + (3c_{n-1} + 3^2c_{n-1}c_{n-2} + \cdots + 3^{n-1}c_{n-1}c_{n-2}\ldots c_1)\log(P(1 + 2\underline{d}))\end{aligned}$$

which gives rise to

$$\log(\|\mathrm{v}\|_\infty) \leqslant 3^{n-1}3^{\frac{(n-1)n}{2}}\log(C(1,d))+3^{\frac{(n-1)n}{2}}(3+3^2+\cdots+3^{n-1})\log(P(1+2\underline{d}))$$

and therefore

$$\log(\|\mathrm{v}\|_\infty) \leqslant 3^{n-1}3^{\frac{(n-1)n}{2}}\log(C(1,d)) + 3^{\frac{(n-1)n}{2}} \times 3 \times \frac{3^{n-1}-1}{2}\log(P(1+2\underline{d})).$$

Thus we can bound $\log(\|\mathrm{v}\|_\infty)$ by $N \times N^{\frac{\log_3(N)}{2}}(\log(C(1,d)) + \log(P(1+2\underline{d}))$. Now we assume that the infinity norm of a fundamental unit of cubic field is less than its real embedding (it is verified by computations made) or at least a polynomial of evaluated in it. Therefore we can write

$$\log(\|\mathrm{v}\|_\infty) \leqslant N \times N^{\frac{\log_3(N)}{2}}(\log(P(1+2\underline{d}) + \max\{\log(\epsilon_{\underline{\alpha}}) \mid \underline{\alpha} \in \mathbb{F}_3^n \setminus \{0\}\})$$

and by using an upper bound for the regulator of pure cubic fields and the discriminant of a pure cubic field we have

$$\log(\|\mathrm{v}\|_\infty) \leqslant N \times N^{\frac{\log_3(N)}{2}}(\log(P(1+2\underline{d})) + P(\underline{d}))$$

where $P(\underline{d}) = \prod_{i=1}^n d_i$.

If we compare this bound to the experimental results the subexponential part seems reasonable. However the norm of the elements that we work with during the algorithm seem to be lower in general.

Received **???**.

Author information

Andrea Lesavourey,, Australia.
E-mail: `al880@uowmail.edu.au`

Thomas Plantard,, Australia.
E-mail: `thomaspl@uow.edu.au`

Willy Susilo,, Australia.
E-mail: `wsusilo@uow.edu.au`