



The special case of cyclotomic fields in quantum algorithms for unit groups

Razvan Barbulescu, Adrien Poulalion

► To cite this version:

Razvan Barbulescu, Adrien Poulalion. The special case of cyclotomic fields in quantum algorithms for unit groups. AFRICACRYPT 2023, Ministry of Communication Technologies of Tunisia; in partnership with the International association of cryptologic research (IACR), Jul 2023, Soussa, Tunisia. pp.229. <hal-04012986>

HAL Id: hal-04012986

<https://hal.science/hal-04012986v1>

Submitted on 7 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

The special case of cyclotomic fields in quantum algorithms for unit groups

Razvan Barbulescu^{1*} and Adrien Poulalion²

¹ Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France
`razvan.barbulescu@u-bordeaux.fr`

² Alice & Bob, Corps des Mines, France
`adrien.poulalion@mines.org`

Abstract. Unit group computations are a cryptographic primitive for which one has a fast quantum algorithm, but the required number of qubits is $\tilde{O}(m^5)$. In this work we propose a modification of the algorithm for which the number of qubits is $\tilde{O}(m^2)$ in the case of cyclotomic fields. Moreover, under a recent conjecture on the size of the class group of $\mathbb{Q}(\zeta_m + \zeta_m^{-1})$, the quantum algorithm is much simpler because it is a hidden subgroup problem (HSP) algorithm rather than its error estimation counterpart: continuous hidden subgroup problem (CHSP). We also discuss the (minor) speed-up obtained when exploiting Galois automorphisms thanks to the Buchmann-Pohst algorithm over \mathcal{O}_K -lattices.

1 Introduction

The difficulty to compute class groups and its cardinality, the class group, plays an important role in cryptography. Notably, they are at the foundation of a time commitments protocol [49], a scheme of homomorphic encryption [15] and a verifiable delay function [50]. Note that the particular cases play an important role as the former two examples use quadratic fields and for the latter the cyclotomic case can be the fastest.

From a perspective of theoretical computer science, these schemes are broken because there exists a quantum algorithm of polynomial time complexity [10]. However, from a cryptographic point of view it is required to obtain a more precise estimation of the amount of qubits and of quantum gates, as it was done for the other primitives of public key cryptography [7, 4, 40, 6, 44, 51, 29, 45, 40].

Let us recall the chronology of the works addressing this question. Note that both in the classical and quantum algorithms class group and unit group are very similar and they are sometimes computed simultaneously in order to have a halting condition, as in the classical algorithm of Buchmann and McCurley. In 1994, Shor designed a quantum algorithm to factor integers and, respectively, solve the discrete log problem in cyclic groups [48]. Kitaev [33] reformulated the algorithms by reducing both problems to finding the set of periods of functions

* The first author has been funded by the Hybrid quantum initiative (HQI) of the France 2030 program.

defined over \mathbb{Z}^r with finite r and taking values in a finite set; this is the hidden subgroup problem (HSP). Note that the parameter r affects the time and space complexity only in a polynomial manner.

In 2002, Hallgren [28]³ reduced the computation of a fundamental unit of a real quadratic field to finding the set of periods of a function defined over \mathbb{R} . More generally, if a function defined over \mathbb{R}^m has a lattice (discrete subgroup) of periods, then finding the set of periods is the continuous hidden subgroup problem (CHSP). As for factoring and discrete log, the unit group is sub-exponential on a classical computer and polynomial on a quantum one.

When $m = 1$, the main difference between HSP and CHSP is the problem of finding $\alpha \in \mathbb{R}$ when given approximations of $k\alpha$ and $\ell\alpha$ for two integers k and ℓ ; this is solved using continued fractions. To compute fundamental units for a family of number fields of constant degree, one has to solve CHSP for $m \geq 1$ bounded by a constant. In contemporary works, Hallgren [27] and Schmidt [46] replaced the continued fractions by an LLL-based algorithm of Buchmann and Pohst [13].

To this point, CHSP and HSP have an identical quantum part and differ in the classical post-treatment. When the degree of the number fields is free to be unbounded in a family of discriminants going to infinity, one has to solve CHSP for unbounded parameters m , possibly as large as the logarithm of the discriminants. In this case, the previous algorithms [27, 46] require an exponential time in m , the degree of the number fields, as it was shown by Biasse and Song in [11, Prop. B.2]⁴.

In 2014, Eisenträger et al. [21]⁵ achieved to make this algorithm polynomial-time for arbitrary degree number fields. For this the periodic function is multiplied by a Gaussian. Before the full version [22] of this work was made public, a second team, de Boer et al. [19], worked on a more thorough analysis of the algorithm and established in 2019 the precise complexities (space and time) for the CHSP.

Our contribution and cryptographic recommendations. The goal of this article is to make possible a precise resource comparison between computing unit and class groups on one side and breaking symmetric cryptosystems like AES on the other side, as requested by the NIST specifications [43]. Indeed, experts in the technology of quantum computers study the NISQ scenario in which error-corrected quantum computers with 100-to-1000 qubits become available whereas a quantum computer with 10^{20} qubits remains unfeasible or too expensive to be considered. In this context, the estimation of the number of qubits as “polynomial” in [22] is not enough. We instantiate the case of unit and class group in the general frame of CHSP [19] and obtain that the number of qubits is $O(m^5)$, e.g. when $m = 10000$ a quantum attack requires 10^{20} error-corrected qubits.

³ The version of 2002 had 6 pages whereas the version published in 2007 has 19 pages.

⁴ The proposition has number 2 in the version of 2015.

⁵ The 10-page-long original version of 2014 doesn’t contain the proofs, which were made public only in the 47-page-long version of 2019.

We continue by investigating possible weak number fields K : is it a security problem if K has automorphisms? Is it a weakness if K is cyclotomic?

In Section 3.3 we explain that automorphisms allow to implement LLL over \mathcal{O}_K and obtain a speed-up which is at most polynomial in the number of automorphisms.

In the case of cyclotomic fields, the speed-up is not automatic: if one runs the general algorithm on these fields, the number of qubits is $O(m^5 \log m)$. However, we make the following observation: **Assume that L is an unknown lattice, $M \subset L$ is known and has a short basis, and one has an algorithm to produce vectors near L^* . Then one can use the short basis of M to bring any vector near L^* to their nearest vector of M^* and they will automatically be in L^* .**

Cryptographic recommendations. A precise estimation of the resources to compute class and unit groups shows that they are more resistant than symmetric cryptosystems if one avoids weak keys. If one wants to use class group based (CGP-based) cryptography then:

- one must avoid fixed degree fields as they use $O(m)$ qubits;
- one must avoid cyclotomic fields as they use $O(m^2 \log m)$ qubits; Also a HSP-based algorithm exists;
- one might prefer to use fields with automorphisms to speed-up the computations as they require $\Omega(m^3)$ qubits and the speed-up concerns only the Buchmann-Pohst algorithm and is polynomial in the number of automorphisms.

Road map. The article is organized as follows. In Section 2, we explain our improvement as a general problem of lattices, outside the context of unit groups and quantum computing. Then, in Section 3 we make a detailed presentation of the quantum algorithms for unit groups and combine results to state the complexity of the algorithms for arbitrary number fields. We exploit the automorphisms in Subsection 3.3. In Section 4 we instantiate the previous complexities in the case of cyclotomic and abelian fields, and compare with our improvement from Section 2. Finally, in Section 4.2 we make an attempt to reduce the unit-group computations to HSP instead of CHSP, this reduces massively the number of qubits needed, especially for cyclotomic fields of index p^k for small primes p .

2 Our improvement seen as a lattice problem

Let $\langle x, y \rangle$ denote the dot product of two vectors $x, y \in \mathbb{C}^n$. Given a lattice $L \subset \mathbb{R}^m$, we call dual of L the lattice

$$L^* = \{y \in \mathbb{R}^m \mid \forall x \in L, \langle x, y \rangle \in \mathbb{Z}\}.$$

Let us recall a series of properties of L^* (see [3] for a reference).

- Lemma 21** 1. If L is generated by the rows of a matrix B then L^* is generated by the rows of $(B^t)^{-1}$; in particular $\det L^* = 1/\det L$;
2. If M is a sublattice of L then $L^* \subset M^*$ and $[L : M] = [M^* : L^*]$.

2.1 Informal presentation

The CHSP algorithm to compute unit groups calls L the lattice of units and follows the following strategy:

1. Use a quantum procedure to generate vectors of \mathbb{C}^n in a small ball around each vector of a set of generators of L^* .
2. Apply the Buchmann-Pohst algorithm (which is classical of polynomial time) and the previously found vectors to find a basis of L^* .
3. Invert the matrix of the previously found basis to obtain a basis of L (this is classical of polynomial time).

A characteristic of Buchmann-Pohst's algorithm is that it has a large precision decrease⁶, namely one needs a large number of bits of precision on the vectors computed in Step 1 in order to have much fewer bits of precision on the basis of L^* . Given a precision requirement τ , i.e. $\log \tau$ bits of precision, in [19] one computes that the input of Buchmann-Pohst must have precision which depends on k . In this work we reduce the number of qubits required by Step 1.

In the case of cyclotomic fields, one has a basis for the lattice of cyclotomic units $M \subset L$. This allows to compute a basis of M^* which contains the unknown lattice L^* . Our idea is as follows: when given a vector sufficiently close to L^* , solve CVP with respect to M^* and automatically it will be in L^* . In more detail, let m_1, \dots, m_n be a basis of L . We do Step 1 at low precision and obtain \tilde{y} and then we correct it into y so that $\langle y, m_i \rangle \in \mathbb{Z}^n$ for all $i = 1, 2, \dots, n$. At this point we have a generating set of L^* at high precision. For simplicity of exposition we can still apply the Buchmann-Pohst algorithm. In view of a practical implementation one can also use an algorithm for Hermite normal form.

2.2 Precise statement

Step 1 produces vectors \tilde{y} close to L^* such that $y := \text{CVP}(\tilde{y}, L^*)$ follow a continuous distribution on \mathbb{R}^m which is close to a discrete Gaussian distribution c on L^* . The following definition is a precise description of its output and focuses on the properties of c used in the following sections.

Definition 22 (Dual lattice sampler) *Let $c : L^* \rightarrow \mathbb{C}$ be a map such that $\sum_{\ell^* \in L^*} |c_{\ell^*}|^2 = 1$. Let $1/4 > \epsilon > 0$, $1/2 > \delta > 0$ and $r > 0$ be three parameters. An algorithm is a dual lattice sampler of parameters (ϵ, δ, r) if it outputs a vector $x \in \mathbb{R}^m$ such that, for any finite set $S \subset L^*$, one has*

$$\text{Prob} \left(y \in \bigcup_{\ell^* \in S} B(\ell^*, \delta \lambda_1^*) \right) \geq \sum_{\ell^* \in S} |c_{\ell^*}|^2 - \eta.$$

for a map c satisfying:

⁶ See Th 10 in [19] for a precise estimation.

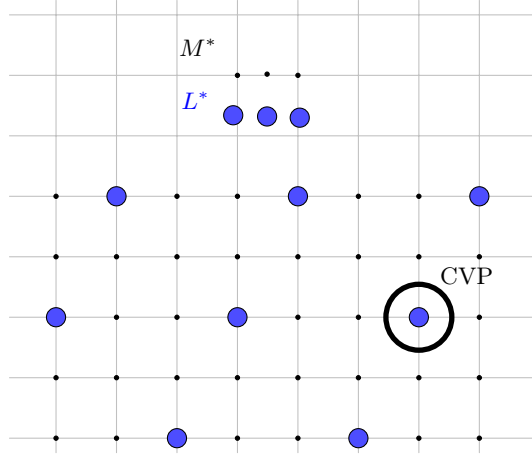


Fig. 1: Illustration of Algorithm 2.

1. **Uniformity property:** there exists $\varepsilon \leq 1/4$ such that, for every strict sublattice $N \subsetneq L^*$:

$$\sum_{\ell^* \in N} |c_{\ell^*}|^2 < \frac{1}{2} + \varepsilon.$$

2. **Concentration property:** There exists $r = r(m)$ and $0 < p < \frac{1}{2} - \varepsilon - \eta$ such that:

$$\sum_{|\ell^*| > r} |c_{\ell^*}|^2 < p.$$

We can now define the problem we tackle.

Problem 1. Let $M \subset L \subset \mathbb{R}^m$ be two lattices. Let $\epsilon, \delta, r > 0$ and $k \in \mathbb{N}$ be parameters. We are given

- a basis of M ;
- an upper bound on $[L : M]$;
- a dual lattice solver for L of parameters δ and η of our choice.

Compute a basis $(z_1, \dots, z_m) \in L$ given by the value of $[L : M]$ and the coordinates of the z_i in a basis of $\frac{1}{[L:M]}M$.

Let B_M be the matrix whose rows are the basis (z_1, \dots, z_m) of M . We propose the following solution to this problem, where we will make precise later the value of δ so that the time complexity is polynomial.

Algorithm 1 Full computation of L using the dual lattice sampler.

Input: – upper bound of $\det L$ of the lattice L and a lower bound on λ_1^*
– $\epsilon, \delta, r > 0$ and a dual sampler of L of these parameters;
– a basis of a sublattice $M \subset L$

Output: a basis of L

- 1: $k \leftarrow m \log_2(\sqrt{m} \text{Lip}(f)) + \log_2(\det L)$ ▷ Value from [19, Th 3]
- 2: **for** $i = 1, 2, \dots, k$ **do** ▷ Step 1 - Quantum
- 3: $\tilde{y}_i \leftarrow \text{output}(\text{dual lattice sampler}, \boxed{\delta \lambda_1^*(L) = 1/2 \|B_M\|})$
- 4: $\boxed{y_i \leftarrow \text{BDD}(\tilde{y}_i, M^*, 1/2 \|B_M\|)}$
- 5: **end for**
- 6: ~~Use Buchmann-Pohst algorithm on (y_1, \dots, y_k) .~~ Compute the Hermite normal form to find a basis (y'_1, \dots, y'_m) of L^* from (y_1, \dots, y_k) . Here the vectors have exact integer coordinates in a basis of M^* . ▷ Step 2 - Classical
- 7: Compute the Smith normal form (SNF) of (y'_1, \dots, y'_m) to obtain the exact value of $[L : M]$. Output $(B^{-1})^t \in \frac{1}{[L:M]} \text{Mat}_m(\mathbb{Z})$, where $B \in \text{Mat}_m(\mathbb{Z})$ is the matrix a basis of L^* written in a basis of M^* . ▷ Step 3 - Classical

Notation 23 For any matrix $B \in \text{GL}_n(\mathbb{C})$ and any $\alpha, \beta \in \mathbb{R}_+ \cup \{\infty\}$ the operator norm is

$$\|B\|_{\alpha, \beta} = \max\{\|Mv\|_{\alpha} \mid \|v\|_{\beta} \leq 1\}.$$

When α and β are not specified⁷, $\|B\|$ denotes $\|B\|_{\infty, 1}$:

$$\|B\| = \|B\|_{\infty, 1} = \max_j \sum_i |B_{i,j}|.$$

When $\alpha = \beta$ we simply write $\|B\|_{\alpha}$. Note that

$$\|B\|_{\infty, 1} \geq \|B\|_2 = \max_i \sqrt{\sum_j |B_{i,j}|^2} \geq \|B\|_{\infty, 1} / \sqrt{m}.$$

In this article, the complexity of the algorithms depends on $O(\log \|B\|_{\infty, 1}) = O(\log \|B\|_2 + \log m)$. Since $\log \|B\| = \Omega(m)$, this justifies that we drop the indices of the operator norm.

Theorem 24 a) Algorithm 1 solves Problem 1 if $\delta \lambda_1^*(M) < 1/(2 \|B_M\|)$. The number of qubits is $O(Qm)$ with Q given in Equation (1).

We prepare the proof with two lemmas.

Lemma 25 (Lemma 5 in [19]) We note $k = \alpha(m + m \log_2 R + \log_2(\det L))$, for an absolute constant $\alpha > 1$.

⁷ The norm operator used in the analysis of the Buchmann-Pohst algorithm [19, Appendix B] is $\|\cdot\|_{\infty, 1}$.

Let $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_k$ be the first k vectors output by a dual basis sampler. For $i = 1, 2, \dots, k$ put $y_i = \text{CVP}(\tilde{y}_i, L)$. Then for any value of the absolute constant $\alpha > 2$ we have

$$\text{Prob}(y_1, \dots, y_k \text{ generate } L) \geq 1 - c^m,$$

where $c < 1$ is an explicitly computable constant.

From now on, we fix such a k , and we use it in Algorithm 1.

Algorithm 2 Babai's BDD solver.

Input: $\tilde{y} \in \mathbb{R}^m$ such that $d(\tilde{y}, M^*) < 1/(2\|B_M\|)$

Output: $\text{CVP}(\tilde{y}, M^*)$, as a vector and as coordinates in a basis of M^*

compute $\tilde{z} := B_M^t \tilde{y}$;

round $z = (z_1, \dots, z_n) := (\lfloor \tilde{z}_1 \rfloor, \dots, \lfloor \tilde{z}_n \rfloor)$

return $y := B_{M^*} z \in M^*$ and $z \in \mathbb{Z}^m$.

Let us put δ such that $d(\tilde{y}, M^*) = \delta \lambda_1^*(M)$. The following result can be seen as a reformulation of a result of Babai.

Lemma 26 ([2] Eq. (4.3)) *Algorithm 2 solves $\text{BDD}(\tilde{y}, M^*, \delta \lambda_1^*(M))$ in classical polynomial time when $\delta \lambda_1^*(M) < 1/(2\|B_M\|_2)$, where B_M is the matrix of a basis of M .*

Proof. We claim that $\lfloor B_M^t \tilde{y} \rfloor = B_M^t y'$ where $y' = \text{CVP}(\tilde{y}, M^*)$. Indeed

$$\|B_M^t \tilde{y} - B_M^t y'\|_\infty \leq \|B_M^t \tilde{y} - B_M^t y'\|_2 \leq \|B_M\|_2 \cdot \|\tilde{y} - y'\|_2 = \|B_M\|_2 \delta \lambda_1^*(M) < 1/2.$$

Since $\lfloor B_M^t \tilde{y} \rfloor = B_M^t y'$, by Lemma 21, $y = B_{M^*} = ((B_M)^t)^{-1} (B_M^t y') = y'$, so the algorithm is correct.

Proof (Proof of Theorem 24.a). We structure the proof in several steps.

step 1. We apply Lemma 25 to the lattice L^* . Then, with probability $1 - c^m$, the vectors y_1, y_2, \dots, y_k generate L^* .

step 2. Let us fix $i \in \{1, 2, \dots, k\}$. By the definition of the dual sampler, with probability greater than $1 - \eta$, $\text{dist}(\tilde{y}_i, L^*) < \delta \lambda_1^*(M^*)$. Hence Lemma 26 applies and Line 4 is executed in classical polynomial time.

Since, $L^* \subset M^*$ and $y'_i := \text{CVP}(\tilde{y}_i, L^*)$ belongs to M^* . Since $d(\tilde{y}_i, L^*) < \lambda_1^*(M)/2$, y'_i is the closest vector of M^* to \tilde{y}_i , or equivalently $y'_i = y_i$. We summarize by $y_i = \text{CVP}(\tilde{y}_i, L)$.

step 3. Algorithm 2 can output z in addition to y . This means that the vectors $y_i \in L^*$ are given by their integer coordinates with respect to a basis of M^* .

In Line 6, one computes the HNF [16, Sec 2.4.2] by manipulating integer coefficients only. This is done in classical polynomial time without loss of precision.

step 4. Since $B_M \in \text{Mat}_k(\mathbb{Z})$, $(B_M^{-1}) \in \frac{1}{\det B_M} \text{Mat}_k(\mathbb{Z})$. Finally, B_M and its SNF [16, Sec. 2.4.4] have the same determinant, which equals $[L : M]$. Hence, the last step computes $[L : M]$ in polynomial time.

3 Previous results on the quantum unit-group calculation

3.1 A summary on CHSP

We first define the Continuous Hidden Subgroup Problem to which the problem of computing the unit group can be reduced.

Definition 31 (Continuous Hidden Subgroup Problem - CHSP [22]) *Let $f : \mathbb{R}^m \rightarrow \mathcal{S}$, where $\mathcal{S} = \oplus_{i \in \{0,1\}^n} \mathbb{C}|i\rangle$ is the space of states of n qubits. The function f is **an (a, r, ε) -oracle hiding the full-rank lattice L** if and only if it verifies the following technical conditions:*

1. L is the period of f , i.e. $\forall x \forall \ell \in L, f(x + \ell) = f(x)$. (periodicity)
2. The function f is a -Lipschitz. (Lipschitz condition)
3. $\forall x, y \in \mathbb{R}^m$ such that $\text{dist}(x - y, L) \geq r$, we have $|\langle f(x) | f(y) \rangle| \leq \varepsilon$. (strong periodicity)

Given an efficient quantum algorithm to compute f , compute the hidden lattice of periods L .

Let us now recall the algorithm which solves CHSP. We mimick the technique which solve the Hidden Subgroup Problem (HSP). For this latter, let $g : G \rightarrow \mathcal{S}$, with G a finite cyclic group. The solution goes as follows:

1. compute the superposition $\sum_{x \in G} |x\rangle$ and then apply g to obtain the superposition of the values of g : $|\psi(g)\rangle = \sum_{x \in G} g(x) |x\rangle$;
2. apply the Quantum Fourier Transform (see [48]) to obtain $\sum_{y \in G} \hat{g}(y) |y\rangle$; this expression is equal to $\sum_{y \in H^\perp} \hat{g}(y) |y\rangle$
3. measure the state to obtain an element $g \in H^\perp$.

A few iterations of the algorithm allow to have a set of generators of H^\perp and hence to compute h .

Coming back to the continuous case, if we were able to implement the Fourier Transform, we would be able to draw random vectors of L^* . If a set of generators is known, then one can extract a basis to completely describe L^* . Then L would be obtained thanks to Lemma 21. The obstacle is here that \mathbb{R}^m is infinite and the Fourier Transform (QFT) is now an integral instead of a finite sum so we cannot compute the QFT precisely. As a way around, we do an approximate computation of the QFT by the means of a Riemann sum, which amounts to restrict the domain to a segment and to discretize it. Set $\rho_\sigma(x) = \exp(-\frac{\pi^2 \|x\|^2}{\sigma^2})$.

Lemma 32 (Thm. 2 in [19]) *There exists dual lattice sampler quantum algorithm of polynomial time which uses $Qm + n$ qubits, where*

$$Q = O\left(m \log\left(m \log \frac{1}{\eta}\right)\right) + O\left(\log\left(\frac{\text{Lip}(f)}{\eta \delta \lambda_1^*}\right)\right). \quad (1)$$

Algorithm 3 A CHSP solver

Input: A function f which can be computed in polynomial time, as in Definition 31 having a hidden period lattice L . We require $\det L$ up to one bit of precision and $\text{Lip}(f)$. A parameter τ of required precision.

Output: $(\tilde{x}_1, \dots, \tilde{x}_m)$ such that (x_1, \dots, x_m) is a basis of L for $x_i = \text{CVP}(\tilde{x}_i)$ and $\|x_i - \tilde{x}_i\| \leq \tau$

1: $k \mapsto \log_2(\sqrt{m} \text{Lip}(f)) + \log_2(\det L)$ ▷ value from [19, th 3]

2: **for** $i = 1, 2, \dots, k$ **do** ▷ Step 1 - Quantum

3: $\tilde{y}_i \leftarrow \text{output}(\text{dual lattice sampler: } \delta\lambda_1^* = \frac{(\lambda_1^*)^3 (\det L)^{-1}}{2^{O(mk)} \|B_{L^*}\|_\infty^m})$ ▷ value from [19,

Appendix B.3 Th 4]

4: **pass** ▷ compare to Algorithm 1

5: **end for**

6: Apply the Buchmann-Pohst algorithm on $(\tilde{y}_1, \dots, \tilde{y}_k)$ and obtain $(\tilde{y}'_1, \dots, \tilde{y}'_m)$ so that $\text{CVP}(\tilde{y}_i, L^*)_{i=1,m}$ is a basis of L^* ▷ Step 2 - Classical

7: Output the rows of $(B^{-1})^t$ where B has rows \tilde{u}'_i . ▷ Step 3 - Classical

Theorem 33 (Space complexity of the CHSP solver, Th 1 of [19]) *We set⁸ $\eta = 1/k^2$. Let N_f be the amount of qubits necessary to encode f . We fix τ the error expected on basis' vectors. Then, Algorithm 3 is correct and requires a number of qubits:*

$$\begin{aligned} N_{\text{qubits}} = & O(m^3 \cdot \log(m)) + O(m^3 \cdot \log(\text{Lip}(f))) + O(m^2 \cdot \log(\det(L))) \\ & + O\left(m \cdot \log \frac{\text{Lip}(f)}{\lambda_1^*}\right) + O\left(m \cdot \log \frac{1}{\lambda_1^* \cdot \tau}\right) + O(N_f). \end{aligned} \quad (2)$$

The complexity of the algorithm depends on λ_1^* which has the advantage that it is an invariant of the lattice. However, λ_1^* is difficult to lower bound and in some applications one has bounds on the coefficients of B_L , the matrix of some basis of L .

Lemma 34 $2^{-3m} \|B_L^t\| \leq \frac{1}{\lambda_1^*} \leq \|B_L\|$.

Proof. The first inequality is [19, Corollary 6 in Appendix B.1].

The second inequality is direct. Let $v \in L^*$ such that $\|v\|_2 = \lambda_1^*$. By Notation 23 we have $\|B_L v\|_1 \leq \|B_L\| \|v\|_\infty$. Since $v \in L^*$, $v \neq 0$, $B_L v \in \mathbb{Z}^n$ and then $\|B_L v\|_1 \geq 1$. Also, $\|v\|_2 \geq \|v\|_\infty$, so

$$1 \leq \|B_L\| \lambda_1^*,$$

which proves the second inequality.

⁸ The complexity of the CHSP solver depends on $\log 1/\eta$ and not on η itself. Hence the contribution of η is hidden in the \tilde{O} notation.

3.2 Reduction of the computation of the unit group to CHSP

In this subsection, we present the method of [22] to reduce the computation of the unit group to the CHSP. For completeness, we reproduce the description of [10, Sec]. Let $G := \mathbb{R}^{n_1+n_2} \times (\mathbb{Z}/2\mathbb{Z})^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$, and the mapping

$$\begin{aligned} \varphi : & (u_1, \dots, u_{n_1+n_2}, \mu_1, \dots, \mu_{n_1}, \theta_1, \dots, \theta_{n_2}) \\ \mapsto & ((-1)^{\mu_1} e^{u_1}, \dots, (-1)^{\mu_{n_1}} e^{u_{n_1}}, e^{2i\pi\theta_1} e^{u_{n_1+1}}, \dots, e^{2i\pi\theta_{n_2}} e^{u_{n_1+n_2}}) \end{aligned}$$

Let σ_1, σ_{n_1} and respectively $\sigma_{n_1+1}, \dots, \sigma_{n_1+n_2}$ be the real and the complex embeddings of K . We consider the Cartesian ring

$$E = \sigma_1(\mathcal{O}_K) \times \dots \times \sigma_{n_1+n_2}(\mathcal{O}_K).$$

An E -ideal is a sub- \mathbb{Z} -lattice A of E which is such that $\forall x \in E, xA \subset A$.

We define

$$\begin{aligned} g : G & \rightarrow \{E\text{-ideals}\} \\ x & \mapsto g(x)E. \end{aligned}$$

Definition 35 (Ex. 5.3 of [22]) Set $\mathcal{H} = \otimes_{i \in \mathbb{N}} \mathbb{C}|i\rangle$, $\mathcal{H}_Q = \otimes_{i=0}^Q \mathbb{C}|i\rangle$ and $\pi_Q : \mathcal{H} \rightarrow \mathcal{H}_Q$ the projection on the first Q qubits. We define the straddle encodings of parameter ν , $\text{str}_\nu : \mathbb{R} \rightarrow \mathcal{H}$, $\text{str}_{m,\nu} : \mathbb{R}^m \rightarrow \mathcal{H}^m$ and $f : \{\text{lattices of } \mathbb{R}^m\} \rightarrow \mathcal{H}^m$ as follows:

$$\begin{aligned} - & |\text{str}_\nu(x)\rangle = \cos(\frac{\pi}{2}t)|k\rangle + \sin(\frac{\pi}{2}t)|k+1\rangle, \text{ where } k = \lfloor x/\nu \rfloor, t = x/\nu - k; \\ - & |\text{str}_{m,\nu}(x_1, \dots, x_m)\rangle = \otimes_{i=1}^m |\text{str}_\nu(x_i)\rangle; \\ - & |\text{str}_{\text{lattice},m,\nu}(L)\rangle = \gamma^{-1/2} \sum_{x \in L} e^{-\pi\|x\|^2/s} |\text{str}_{m,\nu}(x)\rangle \\ & \text{with } \gamma = \sum_{x \in L} e^{-2\pi\|x\|^2/s^2}. \end{aligned}$$

Note that computing $\text{str}_{\text{lattice},m,\nu}$ up to an error τ requires $O(m\nu \log \tau)$ qubits. In Appendix A we propose an alternative function f which can be used for totally real K and doesn't use the straddle encoding.

Finally, one defined f as follows:

$$f : G \xrightarrow{g} \{E\text{-ideals}\} \xrightarrow{|\text{str}\rangle} \{\text{quantum states}\} = \mathcal{H}_Q^m. \quad (3)$$

The following result is the conjunction of several results from [22].

Theorem 36 (Theorems 5.5, D.4 and B.3 of [22]) Let K be a number field of degree n , discriminant D , unit rank m and regulator R . Let $L := \text{Log } \mathcal{O}_K^*$ which is the hidden period of the function f of Equation (3).

Set $s = 3 \cdot 2^{2n} \sqrt{nD}$ and $\nu = 1/(4n(s\sqrt{n})^{2n})$. Then $f_5 = \otimes^5 f$ is an (r, a, ε) -oracle with $\varepsilon = 243/1024$, $\text{Lip}(f) = a = \frac{\sqrt{\pi n s}}{4\nu} + 1$ and $r = s\sqrt{n}^{n-1} 2\nu\sqrt{m}$, where c is an explicitly computable constant. In particular, $\varepsilon < 1/4$, $r < \lambda_1(L)/6$ and

$$\log_2 \text{Lip}(f) = O(m^2 + m \log D).$$

Proof. Note first that if $K = \mathbb{Q}$ or is quadratic imaginary then $m = 0$ and there is nothing to be computed. Also note that $n/2 - 1 \leq m \leq n - 1$ so we can interchange $O(m)$ and $O(n)$ in the asymptotic complexity.

By [22, Th 5.5] f is an (r, ∞, ε) -oracle hiding the lattice $L = \log \mathcal{O}_K^*$ with $r = (s\sqrt{n})^{n-1} 2\nu\sqrt{n}$ and $\varepsilon' = 3/4$.

By [22, Th D.4] f is a -Lipschitz with $a = \frac{\sqrt{\pi ns}}{4\nu} + 1$.

By [22, Lem E.1] $f_5 := f \otimes f \otimes f \otimes f \otimes f$ is an $(5 \text{ Lip}(f), r, (\varepsilon')^5)$ -oracle hiding the same lattice.

In particular, $(\varepsilon')^5 = 243/1024 < 1/4$ satisfies the requirements of the CHSP solver (see Def 22).

The condition $r < \lambda_1(L)/6$ is satisfied as $\lambda_1(L) \geq 1/2$ by [22, Lem B.3] and

$$r = O\left(\frac{s\sqrt{n}^{n-1}2}{4n(s\sqrt{n})^{2n}}\right) \leq \frac{1}{2s^{n-1}} \leq \frac{1}{6 \cdot 2^n \sqrt{D}} < \frac{1}{12}.$$

Finally, we have

$$\log_2 \text{Lip}(f) = O(ms) = O(m^2 + m \log D).$$

Corollary 37 *Let K be a number field of discriminant D and unit rank m . For any error bound $\tau > 0$. There exists a quantum algorithm of time $\text{poly}(m, \log D, \log \tau)$ using a number of qubits*

$$N_{\text{qubits}} = O(m^5 + m^4 \log D) + O(m \log \tau^{-1})$$

which, for a set of units $\mu, \varepsilon_1, \dots, \varepsilon_m$ such that μ is a root of unity and the other have infinite order and

$$\mathcal{O}_K^* \simeq \mu^{\mathbb{Z}/\omega} \times \varepsilon_1^{\mathbb{Z}} \times \dots \times \varepsilon_m^{\mathbb{Z}},$$

the algorithm outputs $\sigma_i(\log(\sigma_i(\varepsilon_j)))_{1 \leq i \leq m, 1 \leq j \leq n} + \tau_{i,j}$ with $\tau_{i,j} \in \mathbb{R}$ such that $|\tau_{i,j}| \leq \tau$.

Proof. By Theorem 36, there exists a function f which hides $\text{Log } \mathcal{O}_K^*$ and which is an (a, r, ε) -oracle such that $r \leq \lambda_1(L)/6$ and $\varepsilon < 1/4$. By Theorem 33 there exists a polynomial-time algorithm which uses a number of qubits as in Equation (2).

In the CHSP solver one takes stores the values of f on Qm qubits, so the $O(N_f) = O(Qm)$. We are left with estimating Q . For this, the main part is Lemma 21:

$$O(\log(1/\lambda_1^*)) = \log_2(\|B_L\|) = O(m + \frac{1}{m} \log D).$$

We used the fact that L admits an LLL-reduced basis, which is enough to bound $1/\lambda_1^*$, but this cannot be computed because we have no basis of L until the end of the CHSP algorithm. We inject this in Equation (2) and obtain the announced value.

Given $\text{Log } \mathcal{O}_K^*$, we are left with computing $\mu, \varepsilon_1, \dots, \varepsilon_m$. Let ω be the number of roots of unity of K and recall that ω divides D . By multiplying the time by $\log D$, we can enumerate the divisors of D , so we can assume that we know ω .

By Dirichlet's Theorem, an m -tuple such that $\text{Vol}(\text{Log}(\varepsilon_j)) = R$ and a root of unity of order ω , form a basis of the unit group.⁹

Remark 38 ¹⁰ *The analytic class number formula states that $Rh = \sqrt{D}^{1+o(1)}$. Very little is known on the distribution of h but no conjecture, e.g. the Cohen-Lenstra heuristic, is contradictory with the fact that $h = 1$ and $R = \sqrt{D}^{1+o(1)}$ for a proportion of $1 - o(1)$ of the number fields. Hence, $\max_i(\|\log \varepsilon_i\|) \sim R^{1/m}$ and the unit group is fully determined only when $\log \tau = \Omega(\sqrt{D})$. In that case CHSP doesn't compute the full unit group in polynomial-time.*

Note however that in the case of classical algorithms there is no algorithm which computes a partial information on the regulator without fully computing it (see for instance [12]). So the algorithm studied in this article suggests a quantum advantage in the case of unit groups.

Example 1. 1. If $K = \mathbb{Q}(\zeta_n)$ with prime n then $m = (n - 1)/2$ and $\log_2 D = (m - 2) \log_2 m$. If one applies Algorithm 1 without taking notes that it is a very particular, the algorithm uses $O(m^5 \log m)$ qubits.
2. If K is a Kummer extension, i.e. $K = \mathbb{Q}(\sqrt[n]{D})$ for some integer without powers D , then $O(\log \text{disc } K) = O(\log D)$ and the number of qubits is $O(n^5 + n^4 \log D)$. The value of n and D are independent. When n is fixed, i.e; the case of real quadratic fields, the number of qubits is $O(\text{disc } K)$. When $n \sim \log D$, the number of qubits is $(\log D)^5 / (\log \log D)^4$.

The large number qubits used by the present algorithm is in contrast with Shor's algorithm, which requires $2m + O(1)$ qubits in the case of factorization and discrete logarithms and $7m + O(1)$ in the case of elliptic curve discrete logarithms. The paragraph "Conclusion and Research Directions" at the end of [19, Sec 1] states that, even if the complexity in $\log D$ is ignored and one uses approximation techniques in the quantum Fourier transform, the algorithm requires $\Omega(m^3)$ qubits. We refer to Appendix B for an informal discussion about the security levels one can propose in the case of quantum algorithms.

3.3 Exploiting automorphisms

In this section we tackle the case of cyclotomic fields without using the cyclotomic units. Instead we use the fact that K has Galois automorphisms.

The main impact of this section is to reduce the practical cost of the Buchmann-Pohst step (which is non-quantum). Indeed, this step consists in applying LLL to a lattice L which has more algebraic structure: it is a $\mathbb{Z}[\zeta_k]$ -module for an integer k . From an asymptotic point of view, Fieker and Stehlé [24] proved that, if one is to find a new $\mathbb{Z}[\zeta_k]$ -basis of L which is shorter in a sense to be specified later, one has the best asymptotic complexity up to a polynomial factor if one follows the steps:

⁹ We used here a different proof than in [22] where one reduces the case of CHSP defined on arbitrary abelian groups to the case of CHSP over \mathbb{R}^m .

¹⁰ We are indebted to Bill Allombert who has made this objection.

1. forget the $\mathbb{Z}[\zeta_k]$ -module structure;
2. reduce the basis of the underlying \mathbb{Z} -lattice;
3. compute again the $\mathbb{Z}[\zeta_k]$ -module structure.

From a practical point of view though, we shall explain that it is faster to work directly with the $\mathbb{Z}[\zeta_k]$ -module structure.

The structure of $\mathbb{Z}[G]$ -lattice. In the following, a lattice which has a structure of R -module for some ring R is called a R -lattice. When $G := \text{Aut}(K/\mathbb{Q})$, the group of Galois automorphisms, is abelian, as it is the case for cyclotomic fields, $\mathcal{O}_K^*/(\mathcal{O}_K^*)_{\text{tors}}$ has a structure of $\mathbb{Z}[G]$ -lattice. Indeed, for $\sigma \in G$ and $u \in \mathcal{O}_K^*$ we set $\sigma \cdot u = \sigma(u)$ and for set $\sigma_1, \dots, \sigma_k \in G$ of ring generators of $\mathbb{Z}[G]$, and any $\lambda = \sum_{e_1, \dots, e_k} c_{e_1, \dots, e_k} \sigma_1^{e_1} \cdots \sigma_k^{e_k}$ we set

$$u^\lambda = \prod_{e_1, \dots, e_k} (\sigma_1^{e_1}(u) \cdots \sigma_k^{e_k}(u))^{c_{e_1, \dots, e_k}}.$$

Lemma 39 *Let $G := \text{Aut}(F/\mathbb{Q})$ and $\sigma \in G$ of order k .*

1. *Then \mathcal{O}_F^* is an $\mathbb{Z}[\zeta_k]$ -lattice.*
2. *If k is not of the form*

$$k = 2^\varepsilon \prod p_i^{e_i} \text{ with } \varepsilon \in \{0, 1\} \text{ and } p_i \equiv 1 \pmod{4}$$

then \mathcal{O}_F^ is a $\mathcal{O}_{\mathbb{Q}(i\sqrt{d})}$ -lattice for some divisor d of k .*

Proof. 1. If $\sigma \in G$, every $\mathbb{Z}[G]$ -lattice is a $\mathbb{Z}[\sigma]$ -lattice. In order to prove that Φ_k is the minimal polynomial of the endomorphism associated to σ , let $u \in \mathcal{O}_F^*$ and set $v = u^{\Phi_k(\sigma)}$. Then $v^{\sigma^{-1}} = u^{\sigma^{k-1}} = u/u = 1$ and further $\sigma(v) = v$. But $\sigma \in \text{Aut}(F/\mathbb{Q})$, so $v \in \mathcal{O}_F^* \cap \mathbb{Q} = \pm 1 \in (\mathcal{O}_F^*)_{\text{tors}}$. Equivalently, $\Phi_k(\sigma)(u) \in (\mathcal{O}_F^*)_{\text{tors}}$ and Φ_k is an annihilating polynomial of σ . Since it is irreducible, it is the minimal polynomial of σ . But ζ_k has the same minimal polynomial, so $\mathbb{Z}[\sigma] \simeq \mathbb{Z}[\zeta_k]$.

2. It is a classical result that, if one sets $p^* = (-1)^{\frac{p-1}{2}} p$ for a prime p , then

$$\mathbb{Q}(\sqrt{p^*}) \subset \mathbb{Q}(\zeta_p).$$

When $k' \mid k$ one has $\mathbb{Q}(\zeta_{k'}) \subset \mathbb{Q}(\zeta_k)$. Hence if k is divisible by 4 or a prime $p \equiv 3 \pmod{4}$, one can take $d = 4$ or $d = p$ in the statement of the lemma.

Norm-Euclidean rings \mathcal{O}_K . Recall that $\mathbb{Z}[i] = \mathbb{Z}[\zeta_4]$ and $\mathcal{O}_{\mathbb{Q}(i\sqrt{3})} = \mathbb{Z}[\zeta_3]$ are norm-Euclidean: let $k = 3$ or 4 , in order to divide $a \in \mathbb{Z}[\zeta_k]$ by $b \in \mathbb{Z}[\zeta_k]$ one rounds each coordinate of $a/b \in \mathbb{Q}[\zeta_k]$ and denotes the result by q , and then one sets $r = a - bq$; if $N_{K/\mathbb{Q}}(r) < N_{K/\mathbb{Q}}(b)$ for all a, b we say that $\mathbb{Z}[\zeta_k]$ is norm-Euclidean. Napias [42] showed that the LLL algorithm extends naturally to the norm-Euclidean rings $\mathbb{Z}[i]$ and $\mathbb{Z}[\zeta_3]$.

Definition 310 (norm-Euclidean ring, Prop 1.2.5 of [14], Def 2 of [32])
The Euclidean minimum of a number field K is

$$\mathfrak{M}_K := \max_{x \in K} \min_{y \in \mathcal{O}_K} N_{K/\mathbb{Q}}(x - y).$$

In particular, when K is imaginary quadratic, $\mathfrak{M}_K = \max_{x \in \mathbb{C}} \min_{y \in \mathcal{O}_K} |x - y|^2$. If $\mathfrak{M}_K < 1$ we say that K is norm-Euclidean.

Example 2 (Prop 1.2.4 of [14], Prop 3 of [32]). The following fields are norm-Euclidean:

- $\mathbb{Q}(\zeta_k)$ for $k \in \{1, 3, 4, 5, 7, 8, 9, 11, 12, 15, 16, 20, 24\}$;
- $\mathbb{Q}(i\sqrt{d})$ for $d \in \{1, 2, 3, 7, 11\}$.

Definition 311 (\mathcal{O}_K -LLL-reduced basis) *Let $K = \mathbb{Q}(i\sqrt{d})$ be an imaginary quadratic field. We identify K with its embedding in \mathbb{C} such that $\sqrt{-d}$ is mapped to a complex number of positive imaginary part.*

We consider the dot product $\langle \cdot, \cdot \rangle$ of \mathbb{C}^m and we set $\|x\| = \sqrt{\langle x, x \rangle}$, which is a norm.

Let b_1, b_2, \dots, b_m be independent in \mathbb{C}^m . We say that $L := \oplus_{j=1}^m \mathbb{Z}[\zeta_k]b_j$ is a $\mathbb{Z}[\zeta_k]$ -lattice and $\{b_j\}$ is a basis. Let $\{b_1^ = b_1, b_2^*, \dots, b_m^*\}$ be the Gram-Schmidt orthogonalization of $\{b_j\}$ with respect to the dot product. We set $\mu_{i,j} = \langle b_i, b_j^* \rangle / \|b_j^*\|^2$. We say that $\{b_j\}$ is \mathcal{O}_K -LLL-reduced with respect to $\mathfrak{M}_K < \delta < 1$ if the following two conditions hold:*

1. $\|\mu_{i,j}\| \leq \mathfrak{M}_K$ for $1 \leq j < i \leq m$; (size reduced)
2. $\|b_i^*\|^2 + \|\mu_{i,i-1}\|^2 \|b_{i-1}^*\|^2 \geq \delta \|b_{i-1}^*\|^2$. (Lovász condition)

In two contemporary works Kim and Lee [32] and Camus [14] extended LLL to all norm-Euclidean rings $\mathbb{Z}[\zeta_k]$ and $\mathcal{O}_{\mathbb{Q}(i\sqrt{D})}$. In particular, when K is a norm-Euclidean quadratic imaginary field, [14, Th. 1.3.8] states that any \mathcal{O}_K -reduced basis b_1, \dots, b_m of L is such that

$$\|b_j\| \leq \left(\frac{1}{\delta - \mathfrak{M}_K} \right)^{j-1} (\det L)^{1/m}.$$

A direct application of \mathcal{O}_K -LLL is the \mathcal{O}_K variant of Buchmann-Pohst algorithm, that we propose in Algorithm 4.

Theorem 312 (adaptation to \mathcal{O}_K -lattices of Th 3.1 in [13]) *Algorithm 4 is correct and terminates in $O((k+m)^6(m + \log(D^{1/m}/\mu)))$ operations on a classical computer.*

Proof. The proof is a verbatim translation of the correctness proof in the case of \mathbb{Z} -lattices. By Equation 3.3, if b_1, \dots, b_m is a reduced basis of a lattice such that $\det L \leq D$, then $\max(\|b_1\|, \dots, \|b_m\|) \leq B$. This is used in the proof of [13, Prop 2.2] to show that the successive minima of the lattice generated by the columns

Algorithm 4 Buchmann-Pohst over \mathcal{O}_K

Input: a \mathcal{O}_K -lattice $L \subset \mathbb{C}^m$ given by approximations $\tilde{g}_1, \dots, \tilde{g}_k \in \mathbb{C}^m$ of $g_1, \dots, g_k \in \mathbb{C}^m$ which \mathcal{O}_K -span L ; a lower bound μ of $\lambda_1(L)$; an upper bound D on $\det L$

Output: approximations of a basis b_1, \dots, b_m over \mathcal{O}_K of L

1: $B \leftarrow c_K^m D^{1/m}$; $C \leftarrow (B/\mu)^m \gamma_m^{1/2}$; $\tilde{M} \leftarrow (k\sqrt{m}/2 + \sqrt{k})C$; $q \leftarrow \lceil \log_2((\sqrt{mk} + 2)\tilde{M}2^{(k-1)/2}/\mu) \rceil$

2: the the \mathcal{O}_K -LLL-reduction of the following matrix on the left to obtain the matrix on the right

$$\begin{pmatrix} \boxed{\tilde{g}_1 2^q} & \cdots & \boxed{\tilde{g}_k 2^q} \\ & 1 & \\ & & \ddots \\ & & & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} \boxed{\tilde{c}_1} & \cdots & \boxed{\tilde{c}_k} \\ \boxed{\mathbf{m}_1} & \cdots & \boxed{\mathbf{m}_k} \end{pmatrix}$$

3: **return** $\tilde{c}_{k-m+1}, \tilde{c}_2, \dots, \tilde{c}_k$

of the matrix in Algorithm 4 are bounded by \tilde{M} . We use the upper bound of the norm of an \mathcal{O}_K -reduced basis of [14, Prop 1.3.8] instead of [38, (1.12)] to obtain the equivalent of [13, Eq (9)]:

$$\|\tilde{c}_j\| \leq 2^{(k-1)/2} \tilde{M} \quad 1 \leq j \leq k-m. \quad (4)$$

For any vector $(\tilde{c}, \mathbf{m}) \in \oplus_{j=1}^k \mathbb{Z}(\tilde{c}_j, \mathbf{m}_j)$ such that $\mathbf{m} = (\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(k)})$ is not a relation of b_1, \dots, b_k , i.e. $\mathbf{m} \notin L^*$, i.e. $\sum_{j=1}^k \mathbf{m}^{(j)} \tilde{b}_j \neq 0$. Then, the arguments in the proof of [13, Th 3.1] can be copied in a verbatim manner to obtain:

$$\|\tilde{c}_j\| > 2^{(k-1)/2} \tilde{M} \quad 1 \leq j \leq k. \quad (5)$$

Since Equations (4) and (5) are contradictory for $1 \leq j \leq k-m$, the only possibility is that $\mathbf{m} \in L^*$. We do the same transformations on the matrix with b_j instead of \tilde{b}_j and call c_j the vectors which replace \tilde{c}_j in this case. What we have proved is that $c_1 = c_2 = \dots = c_{k-m} = 0$.

Since the transformations done in the LLL algorithm on the columns of the matrix are reversible, they preserve the rank of any subset of rows of the matrix. The rank of the first m rows of this matrix is $\dim L = m$. Since $c_1 = c_2 = \dots = c_{k-m} = 0$, the vectors c_{k-m+1}, \dots, c_k form a basis of L .

LLL over $\mathbb{Z}[i\sqrt{d}]$ as a practical improvement The constant hidden in the big Oh of implementation is (inverse) proportional to $\log \delta$, which (inverse) proportional to $\log(\delta - \mathfrak{M}_K)$. Hence the difference is of only a few percentages. The dependence of the time in the rank is quartic, so we replace a number of operations of high-precision real numbers by $2^4 = 16$ times less complex numbers at the same level of precision. Finally, using the Karatsuba trick, a multiplication of complex numbers costs 3 multiplications of real numbers, so the overall gain is a factor $16/3 \approx 5.33$. An implementation [23] of $\mathbb{Z}[i\sqrt{d}]$ -LLL shows that the reduction

of an \mathcal{O}_K -lattice is ≈ 5 times faster when its algebraic structure is used when compared to forgetting it and using only the underlying structure of \mathbb{Z} -lattice: they used $\mathcal{O}_K = \mathbb{Z}[i]$, the coordinates of the basis vectors have 512 bits and $\delta = 0.99$ both over \mathbb{Z} and $\mathbb{Z}[i]$.¹¹

Consequences in cryptography. Given our current contribution, the speed-up due to automorphisms is 5.33. It is an open question to extend it to a larger class of \mathcal{O}_K -rings (e.g. [37] investigate if LLL can be extended to arbitrary \mathcal{O}_K by solving CVP instances in dimension $[\mathcal{O}_K : \mathbb{Z}]$). If Buchmann-Pohst can be extended, the best speed-up that one can target is $[\mathcal{O}_K : \mathbb{Z}]^3$. Indeed, the dependence of LLL in the dimension is quartic whereas the cost of the arithmetic over \mathcal{O}_K is quadratic or quasi-linear if fast arithmetic is used

The speed-up due to the automorphisms is polynomial in the number of automorphisms. This is a familiar situation because a similar speed-up happens for automorphisms in the case of classical algorithms for factorization, discrete logarithms in finite fields and discrete logarithms on elliptic curves (see e. g. [30, Sec 4.3],[5, Sec 5.3]). For instance, the attacks in [20] against ECDSA using endomorphisms of the curve achieved a polynomial speed-up in an algorithm of exponential complexity. The community reacted by suggesting to use elliptic curves with endomorphisms as their effect was now considered to be benign, e.g. [25]. In a similar manner, if the security of the class group of a given degree and determinant is considered sufficient, we suggest to use fields with automorphisms in order to speed-up the protocol, e.g. it has been done by XTR in cryptosystems based on discrete logarithms in finite fields [39].

4 A new algorithm using cyclotomic units

When $K = \mathbb{Q}(\zeta_m)$ for an arbitrary integer m we define the group of cyclotomic units to be the subgroup C of K^* generated by -1 , ζ_m and $\zeta_m^j - 1$ with $j = 1, 2, \dots, m-1$, intersected with the group of units \mathcal{O}_K^* . We follow the notations of [18, Sec. 3], in particular m and k don't have the same meaning as in the other sections. Factor $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ and, for each index i , put $m_i = m/p_i^{\alpha_i}$.

For $j = 1, \dots, m-1$ we set

$$v_j = \begin{cases} 1 - \zeta_m^j, & \text{if for all } i \text{ we have } m_i \nmid j, \\ \frac{1 - \zeta_m^j}{1 - \zeta_m^{m_i}}, & \text{otherwise for the unique } i \text{ such that } m_i \mid j. \end{cases}$$

4.1 Unconditional results

Lemma 41 (Theorem 4.2 in [35]) *The lattice $M := \text{Log } C$ admits the system of generators $\{b_j = \text{Log}(v_j) \mid j = 1, 2, \dots, m-1\}$.*

¹¹ A similar speed-up was obtained in the case of $\mathbb{Z}[\zeta_k]$ -LLL in [32]: an example took 20 s over $\mathbb{Z}[\zeta_k]$ compared to 75 s over \mathbb{Z} .

Lemma 42 (Lemma 3.5 in [18]) *For any integer $j \in \{1, 2, \dots, m-1\}$ we have $\|1 - \zeta_m^j\|_2 = O(\sqrt{m})$. Hence $\|B_M\|_2 = O(\sqrt{m})$.*

Theorem 43 *Algorithm 1 computes a basis of the unit group of $\mathbb{Q}(\zeta_m)$ in $\text{poly}(m)$ time and uses $O(m^2 \log m)$ qubits.*

Proof. The only quantum step of Algorithm 1 is the dual lattice sampler, whose parameters are $\delta = \|B_L\| / \lambda_1^*(L)$ and $\eta = 1/k^2$ where

$$k = \log_2(\sqrt{m} \text{Lip}(f)) + \log_2(\det L)$$

is given in step 1 of the algorithm. In this section m is not necessarily equal to the rank m' of $\mathbb{Z}[\zeta]^*$ but $O(m') = O(m)$ so that we use m and m' interchangeably. By Lemma 32, the sampler used $O(Qm)$ qubits with Q as below. We write $a \ll b$ for $a = O(b)$. We use Th 36: $\log \text{Lip}(f) \ll m^2 + m \log D$ and then $k \ll m^2 + m \log D$. We also use Lemma 42: $\log 1/(\delta \lambda_1^*(L)) \ll \log(1/\|B_M\|) \ll \log m$ and $\log D = (m-2) \log m \ll m \log m$.

$$\begin{aligned} Q &\ll m \log(m \log \frac{1}{\eta}) + \log \frac{1}{\delta \lambda_1^*(L)} + \log(\text{Lip}(f)) \\ &\ll (m \log m + m \log \log k) + \log m + m^2 + m \log D \\ &\ll m \log m + m \log \log D + \log m + m^2 + m \log D \\ &\ll m^2 + m \log D \\ &\ll m^2 \log m. \end{aligned}$$

This is to be compared to the $O(m^5 + m^4 \log D)$ qubits used by number fields in general (Cor 37) and $O(m^5 \log m)$ used by cyclotomic field if the Algorithm 3 were used without taking profit of the cyclotomic units (Ex 1).

Remark 44 *Our improvement can be used whenever the lattice of units has a sublattice admitting a short basis and this is not limited to the cyclotomic fields and, more generally, abelian fields. For instance, Kihel [31] proposed a family of fields with dihedral Galois group which have a full-rank subgroup of units which are short. Although one expects a speed-up in all these cases, the cyclotomic case is even more special. Indeed, the parameter $\delta \leq \|B_M\| / \lambda_1^*(L)$ depends in a large extent on the index $[L : M]$. By [36, Ex 8.5] (see also [17, Th 2.8]), in the case of cyclotomic fields of prime conductor, $[L : M] = h^+(m)$. By Conjecture 1 this is small, so the impact of the full-rank subgroup is more important in this case.*

4.2 Consequences of a recent conjecture for cyclotomic fields

One denotes $h(m)$ the class group of $\mathbb{Q}(\zeta_m)$, $h^+(m)$ the class group of its maximal real subfield, i.e. $K := \mathbb{Q}(\zeta_m + \zeta_m^{-1})$, and one sets $h^-(m) = h(m)/h^+(m)$. A folklore conjecture states that $h^-(m)$ is large but easy to compute whereas $h^+(m)$ is small but hard to compute. A recent work compiled existing conjectures and pushed further the numerical computations, so one can give a precise form to the folklore conjecture.

Conjecture 1 (Assumption 2 in [18]). For all integers m ,

$$h^+(m) \leq \text{poly}(m)$$

for a fixed polynomial poly .

Consequence of the conjecture : class number in real cubic fields without quantum computers Marie-Nicole Gras [26] proposed an algorithm to compute class numbers of cyclic cubic fields in polynomial time with respect to a bound on h . If m is the conductor of a cyclic cubic field, then Conjecture 1 implies that h is polynomial in m . Note also that Schoof [47] proposed an algorithm for $\mathbb{Q}(\zeta_m)^+$ with prime m which is faster when a small bound on $h^+(m)$ is known.

Reduction of unit group computation to HSP

Theorem 45 *Under Conjecture 1 the unit group of $K = \mathbb{Q}(\zeta_m)^+$ and its class group can be computed by an HSP algorithm in polynomial time and space.*

Proof. Let $N = \text{poly}(m)!$ and note that Conjecture 1 implies that $\mathcal{O}_K^* \subset C^{1/N}$. Let $m' = \varphi(m)/2 - 1$ and let $\varepsilon_1, \dots, \varepsilon_{m'}$ be a basis of $C/\langle \zeta_m, -1 \rangle$. Note that it can be computed using the HNF algorithm from the vectors v_j of Lemma 41. Since K is totally real, its only roots of unity are ± 1 . For any $c \in C$, we call vectorization of c and denote $\text{vect}(c)$ the unique $(m' + 1)$ -tuple in $(e_0, e_1, \dots, e_{m'}) \in \mathbb{Z}/2\mathbb{Z} \times (\mathbb{Z}/N\mathbb{Z})^{m'}$ such that $c = (-1)^{e_0} \prod_{j=1}^{m'} \varepsilon_j^{e_j}$.

We set

$$f : \mathbb{Z}/2\mathbb{Z} \times (\mathbb{Z}/N\mathbb{Z})^{m'} \rightarrow \left\{ \begin{array}{l} \text{canonical representations} \\ \text{of Kummer extensions} \end{array} \right\} \quad (6)$$

$$(e_0, e_1, \dots, e_{m'}) \mapsto K(\zeta_N, \sqrt[m']{(-1)^{e_0} \prod_{j=1}^{m'} \varepsilon_j^{e_j}})$$

We claim that the period of f is

$$\text{period}(f) = \{\text{vect}(u) \mid u \in (\mathcal{O}_K)^*\}.$$

Indeed, let (e_j) and (z_j) be such that $f(e_0, \dots, e_{m'}) = f(e_0 + z_0, \dots, e_m + z_{m'})$ and set $\varepsilon = (-1)^{e_0} \prod_j \varepsilon_j^{e_j}$ and $\zeta = (-1)^{z_0} \prod_j \varepsilon_j^{z_j}$.

We have $K(\zeta_N, \sqrt[m']{\varepsilon}) = K(\zeta_N, \sqrt[m']{\varepsilon\zeta})$. Let $d := [K(\zeta_N, \sqrt[m']{\varepsilon}) : K(\zeta_N)]$. Then the criterion of isomorphism of Kummer extensions [34, page 58] states that $\sqrt[m']{\zeta}$ belongs to $K(\zeta_N)$. But $\sqrt[m']{\zeta} \in \mathbb{R}$ and the maximal real subfield of $K(\zeta_N)$ is K , so $\sqrt[m']{\zeta} \in \mathcal{O}_K^*$. Conversely, if $\sqrt[m']{\zeta} \in \mathcal{O}_K^*$ it is direct that $(z_0, \dots, z_{m'})$ is a period of f .

Finally, to compute the complexity of the algorithm, we write $a = \mathcal{P}(b)$ for $a = b^{O(1)}$ and $a = \text{polylog}(b)$ for $a = (\log b)^{O(1)}$. By [33], the cost of the HSP is $\text{polylog}(N)$. Since, for any n , $n! \leq n^n$, we have

$$\text{time} = \text{polylog}(N) = \mathcal{P}(\text{poly}(m) \log(\text{poly}(m))) = \mathcal{P}(\mathcal{P}(m)) = \mathcal{P}(m).$$

5 Conclusion and open questions

1. The unit group algorithms follow a parallel path to those of other problems like the class group e.g Buchmann's algorithm computes the two groups together and the quantum algorithm for class groups [9] is a generalization of the one of Hallgren et al [22]. It is interesting to have a precise estimation of the number of qubits for class groups depending on the bound on the class number, on the computation of discrete logarithms in the class group etc.

2. The idea of a small dual basis can be generalized: a) to all abelian fields; b) to Galois fields with simple group of automorphisms and known units e.g. [31].
3. The possibility of exploiting automorphisms of arbitrary order depends on the possibility of a Buchmann-Pohst algorithm for \mathcal{O}_K -lattices. One might explore such an algorithm which reduces the size of the norms on average but can locally increase them.
4. The quantum algorithms being probabilistic, the output is not certified. Can one do a classical algorithm to prove that a given set of units generate a subgroup which is ℓ -saturated?
5. The technology of quantum computers is very new and it is not known how many physical qubits correspond to a given number of logical ones. Have the gates used by a HSP algorithms less errors and hence require less error-correction? A precise analysis goes beyond the scope of this article.

A An alternative function hiding the units

We place ourselves in the case where K is totally real. As before we call n its degree. Let P be a polynomial which defines K and let $\alpha_1, \dots, \alpha_n$ be the n roots of P in \mathbb{R} . Let

$$w : \{P \in \mathbb{R}[x] \mid \deg P \leq n-1\} \xrightarrow{P} \mathbb{R}^n \xrightarrow{\mapsto} (P(\alpha_1), \dots, P(\alpha_n)).$$

Clearly, w is a linear isomorphism.

Assume that $\text{disc}(f)$ is squarefree and then $\mathcal{O}_K = \mathbb{Z}[x]/P$. We define

$$f : \mathbb{R}^n \rightarrow (\mathbb{R}_n[x] \bmod \mathbb{Z}_n[x]) \times (\mathbb{R}_n[x] \bmod \mathbb{Z}_n[x]) \\ (x_1, \dots, x_n) \mapsto (w^{-1}(e^{x_1}, \dots, e^{x_n}), w^{-1}(e^{-x_1}, \dots, e^{-x_n})) \bmod \mathbb{Z}_n[x]^2.$$

We claim that the set of periods of f form a lattice Λ such that

$$2 \text{Log } \mathcal{O}_K^* \subset \Lambda \subset \text{Log } \mathcal{O}_K^*.$$

Indeed, let ε be in \mathcal{O}_K^* and set $(x_1, \dots, x_n) = \text{Log}(\varepsilon^2)$. Then $w^{-1}(e^{x_1}, \dots, e^{x_n})$ is the representative of ε^2 in the normal basis of K . The coordinates are all integers because ε^2 is in $\mathcal{O}_K = \mathbb{Z}[x]/\langle P \rangle$. Similarly, $w^{-1}(e^{-x_1}, \dots, e^{-x_n})$ is the representative of ε^{-2} in the normal basis. Its coordinates are integers because $\varepsilon^{-2} \in \mathcal{O}_K^*$.

Conversely, let (x_1, \dots, x_n) be such that $f(x_1, \dots, x_n) = (0, \dots, 0)$ and let $\varepsilon = w^{-1}(e^{x_1}, \dots, e^{x_n})$. Since $f(x_1, \dots, x_n) = 0$, the coordinates of ε in the normal basis are integers, so $\varepsilon \in \mathcal{O}_K$. Similarly, $\varepsilon^{-1} \in \mathcal{O}_K$, so ε is a unit.

Finally, given Λ , one solves a linear system over $\mathbb{Z}/2\mathbb{Z}$ to find $\text{Log } \mathcal{O}_K^*$. Since K is totally real, its roots of unity are ± 1 and $\text{Log } \mathcal{O}_K^*$ completely determines \mathcal{O}_K^* .

B A discussion on quantum security levels

The NIST post-quantum challenge [43] is willingly open on the definition of the computational resources on a quantum computer:

“Here, computational resources may be measured using a variety of different metrics (e.g., number of classical elementary operations, quantum circuit size, etc.)”

A study conducted by Mosca et al. [41] revealed that part of the experts consider that on the medium term one should not consider attacks which uses billions of qubits but one should protect against attacks which use hundred qubits. A similar situation happens in the case of classical algorithms: in 2015 the recommendations of the standardisation agencies (NIST, ANSSI, etc) was to use 2048-bit RSA keys. However, in [1] the authors made a precise estimation that RSA 1024 can be broken with their implementation of NFS on a million cores and they made a study that 98% of a sample of million+ servers use RSA 1024. Moreover, more than 30% were supporting RSA 768 which had been broken since 2009.

It is then necessary to be more precise on the quantum resources. We propose a classification on the time complexity and the number of qubits, as in Table 1 (for a review of the quantum attacks on the various public-key primitives see e.g. [8]). In this light, a verifiable delay function [50] based on the class group of a number field of high-degree and non-cyclotomic is more secure than RSA.

	polynomial time	subexponential time	exponential time
$O(m)$ qubits	factorization discrete log EC discrete log IQC		lattices error correction codes
$O(m^5 \log m)$ qubits	high degree CGP		
superpolynomial space		isogenies	

Table 1: Classification of cryptographic primitives w.r.t. the resources of quantum attacks: factorization, discrete logarithms, elliptic curve (EC) discrete logarithms, class group of orders of imaginary quadratic fields (IQC), supersingular isogenies, lattices, error correction cryptosystems and class group (CGP) computations.

References

1. David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel

- Thomé, Luke Valenta, et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.
2. László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
 3. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
 4. Gustavo Banegas, Daniel J Bernstein, Iggy van Hoof, and Tanja Lange. Concrete quantum cryptanalysis of binary elliptic curves. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 451–472, 2021.
 5. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *Advances in Cryptology–Eurocrypt 2015*, volume 9056 of *Lecture notes in computer science*. Springer, 2015.
 6. Stephane Beauregard. Circuit for Shor's algorithm using $2n+3$ qubits. *Quantum Information & Computation*, 3(2):175–185, 2003.
 7. Daniel J Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies. In *Advances in Cryptology–Eurocrypt 2019*, pages 409–441. Springer, 2019.
 8. J-F Biasse, X Bonnetain, E Kirshanova, A Schrottenloher, and F Song. Quantum algorithms for attacking hardness assumptions in classical and post-quantum cryptography. 2022.
 9. Jean-François Biasse and Fang Song. On the quantum attacks against schemes relying on the hardness of finding a short generator of an ideal in $\mathbb{Q}(\zeta_{p^n})$. Technical report, Technical report, Tech Report CACR 2015-12, 2015.
 10. Jean-François Biasse and Fang 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*, pages 893–902. SIAM, 2016.
 11. Jean-François Biasse and Fang Song. On the quantum attacks against schemes relying on the hardness of finding a short generator of an ideal in $\mathbb{Q}(\zeta_{2^k})$. *Journal of Mathematical Cryptology*, 13(3), 2019.
 12. Johannes Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. *Séminaire de théorie des nombres*, Paris, 1989(1990):27–41, 1988.
 13. Johannes Buchmann and Michael Pohst. Computing a lattice basis from a system of generating vectors. In *Eurocal'87: European Conference on Computer Algebra*, volume 378 of *Lecture notes in computer science*, 1989.
 14. Thomas Camus. *Méthodes algorithmiques pour les réseaux algébriques*. PhD thesis, Université Grenoble Alpes, 2017.
 15. Guilhem Castagnos and Fabien Laguillaumie. Linearly homomorphic encryption from. In *Topics in Cryptology–CT-RSA 2015*, volume 9048 of *Lecture notes in computer science*, 2015.
 16. H Cohen. A course in computational algebraic number theory. *Graduate Texts in Mathematics*, 1996.
 17. Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 559–585. Springer, 2016.

18. Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Mildly short vectors in cyclotomic ideal lattices in quantum polynomial time. *Journal of the ACM*, 68(2):1–26, 2021.
19. Koen de Boer, Léo Ducas, and Serge Fehr. On the quantum complexity of the continuous hidden subgroup problem. In *Advances in cryptology–Eurocrypt 2020*, volume 12106 of *Lecture notes in computer science*, 2020.
20. Iwan Duursma, Pierrick Gaudry, and François Morain. Speeding up the discrete log computation on curves with automorphisms. In *Advances in Cryptology–Asiacrypt’99*, volume 1716 of *Lecture notes in computer science*, 1999.
21. Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang 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*, pages 293–302, 2014.
22. Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. Long version of the submission to STOC 2014 for A quantum algorithm for computing the unit group of an arbitrary degree number field, 2019. Available online at <https://www.cse.psu.edu/~sjh26/units-stoc-submission.pdf>.
23. Philippe Elbaz-Vincent and Etienne Marcatel. An extension of the fplll library to hermitian lattices. *ACM Communications in Computer Algebra*, 55(2), 2021.
24. Claus Fieker and Damien Stehlé. Short bases of lattices over number fields. In *Algorithmic Number Theory–ANTS IX*, pages 157–173. Springer, 2010.
25. Robert P Gallant, Robert J Lambert, and Scott A Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In *Advances in Cryptology–CRYPTO 2001*, volume 2139 of *Lecture notes in computer science*. Springer, 2001.
26. Marie Nicole Gras. Méthodes et algorithmes pour le calcul numérique du nombre de classes et de unités des extensions cubiques cycliques de \mathbb{Q} . *Journal für die reine und angewandte Mathematik*, 1975.
27. Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 2005.
28. Sean Hallgren. Polynomial-time quantum algorithms for Pell’s equation and the principal ideal problem. *Journal of the ACM (JACM)*, 54(1):1–19, 2007.
29. Thomas Häner, Martin Roetteler, and Krysta Svore. Factoring using $2n + 2$ qubits with toffoli based modular multiplication. *Quantum Information & Computation*, 17(7-8), 2017.
30. Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in Cryptology–CRYPTO 2006*, volume 4117 of *Lecture notes in computer science*. Springer, 2006.
31. Omar Kihel. Groupe des unités pour des extensions diédrales complexes de degré 10 sur \mathbb{Q} . *Journal de théorie des nombres de Bordeaux*, 13(2), 2001.
32. Taechan Kim and Changmin Lee. Lattice reductions over euclidean rings with applications to cryptanalysis. In *IMA International Conference on Cryptography and Coding*, pages 371–391. Springer, 2017.
33. A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.
34. Helmut Koch. *Algebraic number theory*, volume 62 of *Encyclopedia of Mathematical Sciences*. Springer, 1997.
35. Radan Kučera. On bases of the Stickelberger ideal and of the group of circular units of a cyclotomic field. *Journal of Number Theory*, 40(3), 1992.

36. C Lawrence. Introduction to cyclotomic fields, volume 83 of Graduate Texts in Mathematics. 1997.
37. Changmin Lee, Alice Pellet-Mary, Damien Stehlé, and Alexandre Wallet. An LLL algorithm for module lattices. In Advances in Cryptology–Asiacrypt 2019, volume 11922 of Lecture notes in computer science, 2019.
38. Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. Mathematische annalen, 261, 1982.
39. Arjen K Lenstra and Eric R Verheul. The XTR public key system. In Advances in Cryptology–CRYPTO 2000, volume 1880 of Lecture Notes in Computer science, 2000.
40. Xia Liu, Huan Yang, and Li Yang. CNOT-count optimized quantum circuit of the Shor’s algorithm. arXiv preprint arXiv:2112.11358, 2021.
41. Michele Mosca and Marco Piani. 2021 quantum threat timeline report, 2022. Available online at [https://info.quintessencelabs.com/hubfs/Quantum-Threat-Timeline-Report-2021-full-report-final%20\(1\).pdf](https://info.quintessencelabs.com/hubfs/Quantum-Threat-Timeline-Report-2021-full-report-final%20(1).pdf).
42. Huguette Napias. A generalization of the LLL-algorithm over euclidean rings or orders. Journal de théorie des nombres de Bordeaux, 8(2):387–396, 1996.
43. NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. Available online at <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf>.
44. J Proos. Shor’s discrete logarithm quantum algorithm for elliptic curves. Quantum Information and Computation, 3(4), 2003.
45. Martin Roetteler, Michael Naehrig, Krysta M Svore, and Kristin Lauter. Quantum resource estimates for computing elliptic curve discrete logarithms. In Advances in Cryptology–Asiacrypt 2017, volume 10624 of Lecture notes in computer science, 2017.
46. Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pages 475–480, 2005.
47. René Schoof. Class numbers of real cyclotomic fields of prime conductor. Mathematics of Computation, 72(242), 2003.
48. Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science, pages 124–134. Ieee, 1994.
49. Sri Aravinda Krishnan Thyagarajan, Guilhem Castagnos, Fabian Laguillaumie, and Giulio Malavolta. Efficient CCA timed commitments in class groups. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021.
50. Benjamin Wesolowski. Efficient verifiable delay functions. In Advances in Cryptology–Eurocrypt 2019, volume 11478 of Lecture notes in computer science, 2019.
51. Christof Zalka. Shor’s algorithm with fewer (pure) qubits. arXiv preprint quant-ph/0601097, 2006.