Research Article

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 $\mathbb{Q}(\zeta_{2^s})$

https://doi.org/10.1515/jmc-2015-0046
Received September 24, 2015; revised August 31, 2017; accepted May 7, 2019

Abstract: A family of ring-based cryptosystems, including the multilinear maps of Garg, Gentry and Halevi [Candidate multilinear maps from ideal lattices, in: *Advances in Cryptology—EUROCRYPT 2013*, Lecture Notes in Comput. Sci. 7881, Springer, Heidelberg (2013), 1–17] and the fully homomorphic encryption scheme of Smart and Vercauteren [Fully homomorphic encryption with relatively small key and ciphertext sizes, in: *Public Key Cryptography—PKC 2010*, Lecture Notes in Comput. Sci. 6056, Springer, Berlin (2010), 420–443], are based on the hardness of finding a short generator of a principal ideal (short-PIP) in a number field typically in $\mathbb{Q}(\zeta_{2^s})$. In this paper, we present a polynomial-time quantum algorithm for recovering a generator of a principal ideal in $\mathbb{Q}(\zeta_{2^s})$, and we recall how this can be used to attack the schemes relying on the short-PIP in $\mathbb{Q}(\zeta_{2^s})$ by using the work of Cramer et al. [R. Cramer, L. Ducas, C. Peikert and O. Regev, Recovering short generators of principal ideals in cyclotomic rings, IACR Cryptology ePrint Archive (2015), https://eprint.iacr.org/2015/313], which is derived from observations of Campbell, Groves and Shepherd [SOLILOQUY, a cautionary tale]. We put this attack into perspective by reviewing earlier attempts at providing an efficient quantum algorithm for solving the PIP in $\mathbb{Q}(\zeta_{2^s})$. The assumption that short-PIP is hard was challenged by Campbell, Groves and Shepherd. They proposed an approach for solving short-PIP that proceeds in two steps: first they sketched a quantum algorithm for finding *an* arbitrary generator (not necessarily short) of the input principal ideal. Then they suggested that it is feasible to compute a *short* generator efficiently from the generator in step 1. Cramer et al. validated step 2 of the approach by giving a detailed analysis. In this paper, we focus on step 1, and we show that step 1 can run in quantum polynomial time if we use an algorithm for the continuous hidden subgroup problem (HSP) due to Eisenträger et al. [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 2014 ACM Symposium on Theory of Computing—STOC'14*, ACM, New York (2014), 293–302].

Keywords: Lattice-based cryptography, quantum attack, number theory

MSC 2010: 11T71

Communicated by: Martin Roetteler

***Corresponding author: Jean-François Biasse,** Department of Mathematics & Statistics, University of South Florida, 4202 East Fowler Ave, CMC342, Tampa, FL 33620-5700, USA, e-mail: biasse@usf.edu
Fang Song, Department of Computer Science and Engineering, Texas A&M University, Harvey R Bright Building, 3112 TAMU, College Station, TX 77843-3112, USA, e-mail: fang.song@tamu.edu

1 Introduction

A series of works describes cryptosystems relying on the hardness of finding a small generator of a principal ideal in the ring of integers of $\mathbb{Q}(\zeta_{2^s})$. In particular, this problem allows to describe fully homomorphic schemes, such as that of Smart and Vercauteren [18], or the multilinear maps of Garg, Gentry and Halevi [8]. Moreover, these schemes have been described as quantum safe in the absence of quantum attacks against them. This potential for quantum safety was the main appeal to scientists from the Communications Electronics Security Group (CESG) for the development of SOLILOQUY, a cryptosystem relying on the hardness of finding a short generator of a principal ideal.

Since then, the CESG has interrupted the SOLILOQUY program because there were indications that it was not as quantum safe as they originally thought. Campbell, Groves and Shepherd [4] (referred to as CGS hereafter) released an online draft explaining the design of SOLILOQUY and its apparent weaknesses. Most notably, they observed experimentally that finding a short generator of an ideal in the ring of integers of $\mathbb{Q}(\zeta_{2^s})$ polynomially reduced to finding an arbitrary generator (which relates to the principal ideal problem). This fact was rigorously proved by Cramer et al. [6] shortly thereafter.

The bottleneck of a key-recovery attack against schemes relying on the hardness of finding a short generator of a principal ideal is the resolution of the PIP. A classical subexponential algorithm was described by Biasse and Fieker for this task [1, 2]. A quantum polynomial-time algorithm for solving the principal ideal problem (PIP) in classes of number fields of *fixed* constant degree was described by Hallgren [11]. It consists in reducing this problem to an instance of the hidden subgroup problem (HSP) in $\mathbb{R}^{O(n)}$, where n is the degree of the field, and an efficient quantum algorithm solving the HSP instance. However, the complexity of both computing the reduction and the quantum HSP algorithm decay exponentially with the degree. These two difficulties make it challenging to extend Hallgren's algorithm to solve high-degree PIP. The draft of CGS sketches a quantum algorithm for the PIP in high-degree number fields. As usual, it consists of two components:

(i) they reduce PIP to an instance of the HSP on $\mathbb{R}^{O(n)}$ that is different than the one in [11];
(ii) they outline a quantum algorithm for solving this HSP.

However, the draft contains no detailed analysis to justify either step, leaving the correctness and complexity of their algorithm difficult to verify. The new reduction, component (i), does appear to be efficiently computable, and hence resolves one of the difficulties. Nonetheless, their HSP algorithm, component (ii), does not seem to supersede the quantum HSP algorithm by Hallgren in [11], and many experts suspect that it would work efficiently for arbitrary (i.e., non-constant) n.

Contribution. In this paper, we give a closer look at the quantum PIP algorithm proposed by CGS. We intend to distill the justified and valuable pieces out of it and try to extend them to obtain an algorithm that provably works.

Indeed, we show that, combining a valid piece of the reduction, component (i), in CGS with techniques and results from a recent work of Eisenträger et al. [7] (call it EHKS hereafter), one can compute a generator of a principal ideal in $\mathbb{Q}(\zeta_{2^s})$ in quantum polynomial time. Together with the reduction from short-PIP to PIP of Cramer et al. [6], this yields a quantum polynomial-time attack against the FHE scheme of Smart and Vercauteren [18] and the multilinear maps of Garg, Gentry and Halevi [8].

In the appendix, we also point out some potential obstructions of component (ii) (quantum HSP algorithm) of CGS that renders it unlikely to be efficient, based on the state of the art in [11].

In a subsequent work [3], an efficient quantum algorithm is proposed that solves the general S-unit group problem in arbitrary-degree number fields. The PIP problem in general fields is thus solved as well due to simple reduction of computing PIP to finding a proper S-unit group problem.

Overview. Here we give an overview of what the CGS algorithm may fall short of and how to use the recent work EHKS to extend some piece of the CGS algorithm into a correct algorithm for finding a generator of an input ideal.

As is the case in both the constant-degree PIP algorithm by Hallgren [11] and the CGS algorithm, one first reduces the PIP problem to an HSP instance and then uses a quantum algorithm to solve the HSP prob-

lem efficiently. Roughly speaking, the HSP instance describes a function $f\colon G \to S$ for some group G (here $G = \mathbb{R}^{O(n)}$) and set S such that there is a *hidden* discrete subgroup $H \leq G$ for which f is periodic over G and f is injective on G/H (i.e., $f(x) = f(y)$ if and only if $x \in y + H$).

We then need to find H with access to f, which would further allow us to find a generator of the input ideal from H. At the heart of the existing quantum HSP algorithm (e.g., [11]) is a *quantum Fourier sampling* procedure, which essentially generates uniform samples from the Fourier transform of f. In the ideal case, the Fourier transform of f will be peaked at elements of the dual group of H, from which we can recover H efficiently.

However, in reality, one thing inevitable is to *discretize* the function on $\mathbb{R}^{O(n)}$ (and truncate it within a finite window) because computers (classical or quantum) are digital and have finite precision and memory only. In effect, we end up with a discrete function $\tilde{f}\colon \mathbb{Z}^{O(n)} \to S$, and its Fourier transform will become noisy. Namely, a random sample there, by applying the quantum Fourier sampling procedure, is less likely to hit an element in the dual of H. Therefore, to get enough clean samples, one has to repeat many times. By the best known analysis [11], the number of repetitions grows exponentially in the dimension n. The quantum HSP algorithm outlined in CGS does not go beyond Hallgren's algorithm, and hence is unlikely to succeed within polynomial time in n unless with improved analysis.

Instead, the recent work of EHKS proposes a conceptually new notion for HSP over continuous groups such as $\mathbb{R}^n$ (call it continuous HSP). The key distinction is enforcing a stringent *continuity* condition on the function f. Specifically, they require f to be *Lipschitz* so that the change between $f(x)$ and $f(y)$ is bounded by some constant factor of the change between inputs x and y. This additional property ensures that, once we discretize it, its Fourier transform is still concentrated on the dual of H. EHKS then gives a modified quantum Fourier sampling procedure to generate samples from the dual of H (with good approximation) and recover H efficiently. There is also a conceptually novel ingredient in their modified quantum Fourier sampling, which, informally speaking, enables sampling the discrete-*time* Fourier transform (i.e., over $\mathbb{Z}$) of the discretized function rather than its discrete Fourier transform (i.e., over $\mathbb{Z}_N$). This facilitates the analysis and makes the HSP solvable in polynomial time. In the EHKS paper, they showcase the power of this new framework by reducing the problem of computing the unit group of a number field to this continuous HSP on $\mathbb{R}^{O(n)}$, and hence solve the unit group problem efficiently. The reduction generates a function $f = f_{\mathrm{q}} \circ f_{\mathrm{c}}$ that hides the unit group. The function f first computes a basis for a lattice $f_{\mathrm{c}}(x)$, and then encodes the lattice into a quantum state by a *straddle* encoding procedure f_{q}.

A natural idea arises as whether we can recast the CGS reduction, component (i), in the continuous HSP framework and solve it consequently. The CGS reduction is actually similar to the one above for the unit group. They propose a function $f_{\mathrm{CGS}} = f_{q'} \circ F$, where the classical part F computes a lattice from an input $x \in \mathbb{R}^{O(n)}$, and $f_{q'}$ outputs what they call a "quantum fingerprint" of the lattice $F(x)$. While f_{CGS} does hide a generator of the input ideal as a subgroup in $\mathbb{R}^{O(n)}$, the Lipschitz condition is not clear. Luckily, we notice that, by composing F in CGS with the straddle encoding function f_{q} in EHKS, it can be shown to be a valid instance of the continuous HSP. This is possible by observing a nice connection between F and the f_{c} function in EHKS, and reusing many results in EHKS. Details are given in Section 5.

2 An (over-)simplified presentation of quantum computing

In this section, we try to convey the aspects of quantum computing that are relevant to the quantum algorithm described in [4] as well as to other quantum cryptanalysis algorithms without getting too technical. This is achieved at the price of some simplifications. First of all, quantum computations occur on quantum states, which are vectors of the form

$$|x\rangle = \alpha_0|0\rangle + \alpha_2|1\rangle + \cdots + \alpha_{2^k-1}|2^k - 1\rangle,$$

where values involved in this definitions are

- complex numbers α_i such that $\sum_i |\alpha_i|^2 = 1$,
- vectors $|i\rangle$ of $(\mathbb{C}^2)^{\otimes k}$, where $|i\rangle$ is the i-th element of an orthonormal basis.

The notation $|x\rangle|y\rangle$ denotes the tensor product of $|x\rangle$ and $|y\rangle$. A quantum algorithm can be viewed as a unitary matrix U in $\mathbb{C}^{2^k\times 2^k}$ acting on a state via $|x\rangle \mapsto U|x\rangle$ (matrix-vector multiplication). A quantum state only gives away information once it is *measured* (according to the chosen basis). This process returns the answer i with probability $|\alpha_i|^2$ and leaves the system in the state $|i\rangle$. Therefore, whatever happens to the original state (for example, $|0\rangle^{\otimes k}$) has to lead to a state whose measurement yields the result of the algorithm with good probability (typically a constant probability). More generally, when a state has the form $|\psi\rangle = \sum_i |\phi_i\rangle \otimes |y_i\rangle$, where the $|\phi_i\rangle$ are vectors of $(\mathbb{C}^2)^{\otimes k_1}$ such that $\sum_i \langle\phi_i, \phi_i\rangle = 1$ and the $|y_i\rangle$ are an orthonormal basis of $(\mathbb{C}^2)^{\otimes k_2}$, then measuring the second register yields the answer y_i with probability $\langle\phi_i, \phi_i\rangle$ and leaves the system in the state $\frac{1}{|\langle\phi_i|\phi_i\rangle|}\phi_i \otimes |y_i\rangle$.

3 Mathematical background

Lattices. A lattice is a discrete additive subgroup of $\mathbb{R}^m$ for some integer m. The first minimum of a lattice $\mathcal{L}$ is defined by

$$\lambda_1 := \min_{\vec{v}\in\mathcal{L}\setminus\{0\}} \|\vec{v}\|, \quad \text{where} \quad \|\vec{v}\| = \sqrt{\sum_{i\le k} v_i^2} \quad \text{is the Euclidean norm.}$$

A basis of $\mathcal{L}$ is a set of linearly independent vectors $\vec{b}_1, \dots, \vec{b}_k$ such that $\mathcal{L} = \mathbb{Z}\vec{b}_1 + \cdots + \mathbb{Z}\vec{b}_k$. The determinant of $\mathcal{L}$ is $\det(\mathcal{L}) = \sqrt{\det(B \cdot B^T)}$, where $B = (\vec{b}_i)_{i\le k} \in \mathbb{R}^{k\times m}$ is the matrix of a basis of $\mathcal{L}$. For a full dimensional lattice $\mathcal{L}$, the best upper bound we know on $\lambda_1(\mathcal{L})$ is in $O(\sqrt{k}\det(\mathcal{L})^{1/k})$. The dual $\mathcal{L}^*$ of the lattice $\mathcal{L}$ is the lattice of vectors $\vec{v}$ of $\mathbb{R}^m$ such that $\vec{u}\cdot\vec{v} \in \mathbb{Z}$ for all $\vec{u} \in \mathcal{L}$.

Number fields. A number field K is a finite extension of $\mathbb{Q}$. Its ring of integers $\mathcal{O}$ has the structure of a $\mathbb{Z}$-lattice of degree $n = [K : \mathbb{Q}]$. A number field has $r_1 \le n$ real embeddings $(\sigma_i)_{i\le r_1}$ and $2r_2$ complex embeddings $(\sigma_i)_{r_1<i\le 2r_2}$ (coming as r_2 pairs of conjugates). The field K is isomorphic to $\mathcal{O}\otimes\mathbb{Q}$. We can embed K in $K_{\mathbb{R}} := K\otimes\mathbb{R} \simeq \mathbb{R}^{r_1}\times\mathbb{C}^{r_2}$ and extend the σ_i's to $K_{\mathbb{R}}$. Let T_2 be the Hermitian form on $K_{\mathbb{R}}$ defined by $T_2(x, x') := \sum_i \sigma_i(x)\overline{\sigma_i}(x')$, and let $\|x\| := \sqrt{T_2(x, x)}$ be the corresponding L_2-norm. The (algebraic) norm of an element $x \in K$ is defined by $\mathcal{N}(x) = \prod_i \sigma_i(x)$. Let $(\alpha_i)_{i\le d}$ such that $\mathcal{O} = \bigoplus_i \mathbb{Z}\omega_i$; then the discriminant of K is given by $\Delta = \det^2(T_2(\alpha_i, \alpha_j))$. The volume of the fundamental domain is $\sqrt{|\Delta|}$, and the size of the input of algorithms working on an integral basis of $\mathcal{O}$ is in $O(\log(|\Delta|))$.

Cyclotomic fields. A cyclotomic field is an extension of $\mathbb{Q}$ of the form $K = \mathbb{Q}(\zeta_N)$, where $\zeta_N = e^{2i\pi/N}$ is a primitive N-th root of unity. The ring of integers $\mathcal{O}$ of K is $\mathbb{Z}[X]/(\Phi_N(X)) = \mathbb{Z}[\zeta_N]$, where Φ_N is the N-th cyclotomic polynomial. When N is a power of two, $\Phi_N(X) = X^{N/2} + 1$, and when $N = p^s$ is a power of $p > 2$, we have $\Phi_N(X) = X^{p^{e-1}(p-1)} + X^{p^{e-1}(p-2)} + \cdots + 1$ (which generalizes the case $p = 2$). Elements $a \in \mathbb{Z}[\zeta_N]$ are residues of polynomials in $\mathbb{Z}[X]$ modulo $\Phi_N(X)$ and can be identified with their coefficient vectors $\vec{a} \in \mathbb{Z}^{\phi(N)}$, where $\phi(N)$ is the Euler totient of N (and the degree of $\Phi_N(X)$). When $N = p^s$ for p a prime, the degree of K satisfies $[K : \mathbb{Q}] = (p-1)p^{s-1}$ and $\Delta = \pm p^{p^{s-1}(ps-s-1)}$; therefore $\log(|\Delta|) \sim n\log(n)$, and we can express the complexity of our algorithms in terms of n (a choice we made in this paper).

Fractional ideals in K. Elements of the form $\frac{\mathfrak{I}}{d}$, where $\mathfrak{I} \subseteq \mathcal{O}$ is an (integral) ideal of the ring of integers of K and $d > 0$, are called fractional ideals. They have the structure of a $\mathbb{Z}$-lattice of degree $n = [K; \mathbb{Q}]$, and they form a multiplicative group $\mathcal{I}$. Elements of $\mathcal{I}$ admit a unique decomposition as a power product of prime ideals of $\mathcal{O}$ (with possibly negative exponents). The norm of integral ideals is given by $\mathcal{N}(\mathfrak{I}) := [\mathcal{O} : \mathfrak{I}]$, which extends to fractional ideals by $\mathcal{N}(\mathfrak{I}/\mathfrak{J}) := \mathcal{N}(\mathfrak{I})/\mathcal{N}(\mathfrak{J})$. The norm of a principal (fractional) ideal agrees with the norm of its generator $\mathcal{N}(x\mathcal{O}) = |\mathcal{N}(x)|$.

Units of $\mathcal{O}$. Elements $u \in \mathcal{O}$ that are invertible in $\mathcal{O}$ are called units. Equivalently, they are the elements $u \in \mathcal{O}$ such that $(u)\mathcal{O} = \mathcal{O}$ and also such that $\mathcal{N}(u) = \pm 1$. The unit group of $\mathcal{O}$, where K is a cyclotomic field, has rank $r = \frac{n}{2} - 1$ and has the form $\mathcal{O}^* = \mu \times \langle\epsilon_1\rangle \times \cdots \times \langle\epsilon_r\rangle$, where μ are roots of unity (torsion units) and the ϵ_i are non-torsion units. Such $(\epsilon_i)_{i\le r}$ are called a system of fundamental units of $\mathcal{O}$. Units generate a lattice $\mathcal{L}$ of rank r in $\mathbb{R}^{r+1}$ via the embedding $x \in K \mapsto \mathrm{Log}(x) := (\ln(|\sigma_1(x)|), \dots, \ln(|\sigma_{r+1}(x)|))$, where the com-

plex embeddings $(\sigma_i)_{i\leq n}$ are ordered such that the first $r = \frac{n}{2}$ ones are not conjugates of each other. When $K = \mathbb{Q}(\zeta_{p^s})$, logarithm vectors of units of the form

$$u_j = \frac{\zeta_{p^s}^j - 1}{\zeta_{p^s} - 1} \quad \text{for } j \in \mathbb{Z}_{p^s}^*$$

(the cyclotomic units) generate a sublattice of $\mathcal{L}$ of index $h^+(p^s)$, where $h^+(N\mathfrak{p}^s)$ is the class number of the maximal real subfield of $\mathbb{Q}(\zeta_{p^s})$ [19, Lemma 8.1].

Conjecture 3.1 (Weber class number problem). *For all $s \in \mathbb{Z}_{>0}$, we have $h^+(2^s) = 1$.*

The hidden subgroup problem. The problem of factoring an RSA integer reduces to an instance of the so-called *hidden subgroup problem* (HSP).

Definition 3.2 (Hidden subgroup problem over $\mathbb{Z}$). Given $f\colon \mathbb{Z} \mapsto X$ for a finite set X such that there exists a subgroup $H \leq \mathbb{Z}$ with

$$f(x+g) = f(x) \quad \text{for all } x \in \mathbb{Z} \quad \text{if and only if} \quad g \in H,$$

the hidden subgroup problem is the task of finding H given oracle access to f. This means finding r such that $H = r\mathbb{Z}$.

We want to factor an RSA integer $N = pq$. Let a be coprime with N (if $a \mid N$, the factorization problem is solved) and

$$\mathbb{Z} \xrightarrow{f} \mathbb{Z}/N\mathbb{Z},$$
$$x \to a^x \bmod N.$$

A solution to the HSP with f yields r, the order of $a \bmod N$, and if a is a square, we get

$$(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N.$$

This means that $N \mid (a^{r/2} - 1)(a^{r/2} + 1)$, and $\gcd(N, a^{r/2} - 1)$ may yield a non-trivial factor of N. A generalization of the HSP to $\mathbb{Z}^m$ allows us to solve the discrete logarithm problem in a finite group, and we can even discretize $\mathbb{R}$ to generalize the algorithms for efficiently solving the HSP to $\mathbb{R}^m$, where m is fixed. This allows the computation of the class group, the unit group and the resolution of the PIP in classes of number fields of fixed degree [11]. More details about these methods are given in the appendix.

4 The PIP quantum algorithm proposed by CGS

CGS proposed a quantum algorithm for solving the PIP in $\mathbb{Q}(\zeta_{2^s})^+$. They suggested to combine it with the Gentry–Szydlo (classical) attack [9] to solve the PIP in $\mathbb{Q}(\zeta_{2^s})$. They sketched this method in [4, Section 5], but they did not provide any complexity analysis.

In this section, we review the PIP algorithm proposed in [4], and we illustrate the challenges that would need to be overcome to turn this approach into a quantum polynomial-time algorithm. There are two main steps to the approach of [4]:

(i) A reduction of the PIP in $\mathbb{Q}(\zeta_{2^s})^+$ to the search of the periods of a function from $\mathbb{R}^n \times \mathbb{Z}$ to the lattices in $\mathbb{R}^n$, where $n = \deg(\mathbb{Q}(\zeta_{2^s})^+)$ (an analogue of the HSP).

(ii) The search for the periods of a function $\mathbb{R}^n \times \mathbb{Z}$ with an algorithm similar to the HSP algorithm of Hallgren [11].

This means that CGS exhibited a function

$$f\colon G \subseteq \mathbb{R}^m \to \{\text{lattices over } \mathbb{R}^n\} \to \{\text{quantum states}\}$$

for some subgroup G and $m \in \mathbb{Z}_{>0}$ such that $f(x) = f(y)$ if and only if $x = y \bmod \Lambda$ for a lattice $\Lambda \subseteq \mathbb{R}^m$ whose knowledge answers the original problem (the PIP in this case). Step (ii) consists in finding the periods of f in a fashion similar to the resolution of the HSP.

Reduction to the search for the periods of a function. Let $K = \mathbb{Q}(\zeta_{2^s})$, $n = \deg(K^+)$ and $r = n - 1$. Let $\alpha' \in K$ be a totally positive generator (not necessarily small) of the input fractional ideal $\mathfrak{a}$ in the totally real number field K. In the context of the attacks against the short-PIP in K, we know that one of the generators of $\mathfrak{a}$ arises as the relative norm $\mathcal{N}_{K/K^+}(g) = g\overline{g}$ of the secret key g. This relative norm is necessarily totally positive.

Let $u_1, \dots, u_r$ be a generating set of $U^+ \simeq \mathbb{Z}^r$, the totally positive units of the ring of integers $\mathcal{O}$ of K^+. Then every totally positive generator of the principal ideal $\mathfrak{a}$ of K^+ (including $\mathcal{N}_{K/K^+}(g)$) is of the form $\alpha' \cdot u_1^{x_1} \cdots u_r^{x_r}$. Let $\beta \in K$; then $\beta \cdot \mathcal{O} = \mathfrak{a}^{-k}$ for some $k \in \mathbb{Z}$ if and only if $\mathrm{Log}(\beta) = \sum_i x_i \mathrm{Log}(u_i) - k\,\mathrm{Log}(\alpha')$ for some $(x_i)_{i\leq r} \in \mathbb{Z}^r$. This means that the lattice $\Lambda_{\alpha'} \subseteq \mathbb{R}^n \times \mathbb{Z}$ defined by

$$\Lambda_{\alpha'} := \mathbb{Z}(\mathrm{Log}(\alpha'), -1) + \mathbb{Z}(\mathrm{Log}(u_1), 0) + \cdots + \mathbb{Z}(\mathrm{Log}(u_r), 0)$$

consists of all the pairs $(\mathrm{Log}(\beta), k)$, where $k \in \mathbb{Z}$, $\beta \in K$ and $\beta \cdot \mathcal{O} = \mathfrak{a}^{-k}$. This includes elements of the form $(\mathrm{Log}(a), -1)$, where $a \in K^+$ is a totally positive generator of $\mathfrak{a}$. This means that a basis for $\Lambda_{\alpha'}$ yields a totally positive generator of $\mathfrak{a}$.

We now describe a function on $\mathbb{R}^n \times \mathbb{Z}$ whose periods are precisely $\Lambda_{\alpha'}$. For $k \in \mathbb{Z}$ and $v \in \mathbb{R}^n$ (not necessarily corresponding to the valuations of an element in K^+), let us denote by $e^v \cdot \mathfrak{a}^k$ the Euclidean lattice generated by the elements of the form $e^v \cdot a$ for $a \in \mathfrak{a}^k$. Elements in K^+ such as $a \in \mathfrak{a}^k$ correspond to real vectors $(\sigma_1(a), \dots, \sigma_n(a))$, and an element of the form $e^v \cdot a$ is represented by the vector $(e^{v_1}\sigma_1(a), \dots, e^{v_n}\sigma_n(a)) \in \mathbb{R}^n$. We define the function $F\colon G \to \{\text{lattices over } \mathbb{R}^n\}$ by $F(v, k) := e^v\mathfrak{a}^k$. Then $F(v, k) = \mathcal{O}$ if and only if e^v is a generator of $\mathfrak{a}^{-k}$, which is equivalent to $(v, k) \in \Lambda_{\alpha'}$. Therefore, by linearity of F, the periods of F are exactly $\Lambda_{\alpha'}$.

As each element (v, k) of $\Lambda_{\alpha'}$ satisfies $\sum_i v_i = -k\log(\mathcal{N}(\mathfrak{a}))$, the search of the corresponding hidden subgroup can be restricted to the control space

$$G = \left\{(v, k) \in \mathbb{R}^n \times \mathbb{Z} \text{ such that } \sum_i v_i = -k\log(\mathcal{N}(\mathfrak{a}))\right\}.$$

The function F used by CGS is different from the one used by Hallgren in [11] to solve the PIP. In particular, F can be evaluated in polynomial time even when the degree of K grows to infinity. This comes from the fact that it is very similar to the function defined by EHKS to hide the unit group of a number field of arbitrary degree, and the techniques they used to evaluate it in polynomial time readily apply.

Proposition 4.1. *The function F can be evaluated in classical polynomial time.*

Proof. This is immediate by application of the techniques of [7, Section 4]. The key observation is that we can perform a square-and-multiply exponentiation on the ideal with LLL-reductions at each step. □

The function F is then composed by a quantum encoding to identify the lattice $e^v\mathfrak{a}^k$. This task is non-trivial since lattices are over $\mathbb{R}^n$ where, unlike in $\mathbb{Z}^n$, there is no canonical form such as the Hermite normal form. This encoding of lattices is called the "quantum fingerprint", and it gives the map

$$f\colon (v, k) \in G \xrightarrow{F} F(v, k) \xrightarrow{\text{fingerprint}} |\psi_{v,k}\rangle.$$

The details of the procedure to create $|\psi_{v,k}\rangle$ from (v, k) are given in [4, Sections 3.4 and 3.5]. It creates a state of the form $\frac{1}{\Gamma}\sum_{\mathbf{x}\in C_n\cap L}|\mathbf{x}\rangle$, where

- $\mathbf{x} \in \mathbb{Z}^n$ is the scaling of a rational approximation of a vector in $\mathbb{R}^n$,
- L is the lattice $F(v, k)$,
- $\Gamma > 0$ is a normalization factor,
- C_n is a bounded set such that $E_n(\rho - \varepsilon) \cap \mathbb{Z}^n \subseteq C_n \subseteq E_n(\rho + \varepsilon) \cap \mathbb{Z}^n$, where $E_n(\rho)$ is an ellipsoid of radius ρ.

CGS conjectured that the quantum encodings of almost identical lattices have inner product close to 1, while the quantum encodings of essentially different lattices have inner product close to 0. The function f "hides" $\Lambda_{\alpha'}$ in the sense that

$$f(v_1, k_1) = f(v_2, k_2) \iff u := (v_1, k_1) - (v_2, k_2) \in \Lambda_{\alpha'}.$$

Identifying $\Lambda_{\alpha'}$ from the periods of this map is an analogue of the HSP.

Property	Status
The function F hides a lattices that reveals a generator of the ideal.	Proved
The function F can be evaluated in classical polynomial time.	Proved
The quantum fingerprint satisfied the "fidelity" property.	Open question
Assuming $\|\psi_{\mathbf{x}}\rangle$ satisfies the "fidelity" property, step (iii) outputs good approximations of vectors in $\Lambda^*_{\alpha'}$.	Open question

Table 1: Steps towards a proof of a polynomial run time of the PIP algorithm of [4].

Computing the periods of f. The method proposed by CGS for computing the periods of f relies on a similar strategy as the HSP resolution algorithm used by Hallgren in [11] to solve the PIP in classes of number field of fixed degree.

(i) Discretize and bound G, and then create the state

$$|\psi\rangle := \frac{1}{\sqrt{M}} \sum_{(v,k)\in G'} |\psi_{v,k}\rangle|(v,k)\rangle.$$

(ii) Apply the quantum Fourier transform over G to the second register.
(iii) Measure (v,k), and check if we obtain a good approximation of an element in $\Lambda^*_{\alpha'}$.
(iv) Repeat steps (ii) and (iii) until a basis of good approximations of $\Lambda^*_{\alpha'}$ is found.
(v) Find an approximation of a basis of $\Lambda_{\alpha'}$ from $\Lambda^*_{\alpha'}$ with classical methods.

In step (i), M is the normalization factor depending on the radius and the precision of the bounded discretized version G' of G. Table 1 highlights the main steps of the quantum algorithm of CGS and specifies those on which we can rely to prove that there is a quantum attack against schemes relying on the short-PIP. In the appendix, we use a method similar to [11] to analyze the behavior of CGS's algorithm. This analysis implies choices of parameters that were not specified by CGS. Therefore, we cannot formally establish the complexity of the algorithm sketched in [4].

In the rest of the paper, we show how to use the quantum encoding proposed by EHKS to solve the HSP in $\mathbb{R}^{O(m)}$ instead of the quantum fingerprint of CGS. EHKS proved that their quantum encoding enjoyed certain properties (one of them being similar to the "fidelity") which allow us to solve the HSP in polynomial time.

5 A method based on the HSP algorithm of EHKS

In this section, we show how to find the periods of the function F defined in the previous section by using the lattice encoding and the corresponding HSP quantum algorithm of EHKS. This allows us to compute in quantum polynomial time a totally positive generator of an ideal $\mathfrak{a}$ in a totally real number field K. Recent work from EHKS developed a new framework for HSP in $\mathbb{R}^m$, which admits an efficient quantum algorithm even for large values of m. They illustrated this by computing the unit group of a number field of arbitrary degree in polynomial time. In this section, we show how to adapt it to calculate a totally positive generator of a principal ideal given by its $\mathbb{Z}$-basis. The algorithm described in [7] returns generators of a secret discrete subgroup H of $\mathbb{R}^m$ for an arbitrary $m > 0$ hidden in the periods of a function $f\colon \mathbb{R}^m \to \{\text{quantum states}\}$. Let G be a subgroup of $\mathbb{R}^m$ containing H. EHKSsshowed in [7, Theorem 6.1] how to recover generators of H in polynomial time in the input if there is an efficiently computable function f satisfying the following properties for $G = \mathbb{R}^m$:

(i) f is periodic on H, that is, $f(x+u) = f(x)$ for all $x \in G,\ u \in H$;
(ii) f is Lipschitz for some constant a: $\||f(x)\rangle - |f(y)\rangle\| \le a \cdot d_G(x,y)$ for all $x, y \in G$;
(iii) there are $r, \varepsilon > 0$ such that, for all $x, y \in G$, if $d_{G/H}(x,y) \ge r$, then $|\langle f(x)|f(y)\rangle| \le \varepsilon$,

where $d_G(x,y) = \|x-y\|$ and $d_{G/H}(x,y) = \inf_{u\in H}\|x-y-u\|$ for the Euclidean norm $\|x\|$.

To construct such a function, it is possible to start from a function defined on a subgroup G of $\mathbb{R}^m$. As shown in [7, Section 6.1], if a function defined on $G \subseteq \mathbb{R}^m$ hides H and satisfies conditions (ii) and (iii) on all $x, y \in G$, it can be used to define a function on $\mathbb{R}^m$ hiding (the embedding of) H and satisfying (ii) and (iii). For simplicity, we use the following notation.

Definition 5.1 ((a, r, ε)-oracle). Let G be a subgroup of $\mathbb{R}^m$ and f a map $G \to \{\text{quantum states}\}$. We say that f is an (a, r, ε)-oracle on G if it satisfies conditions (ii) and (iii) for some (a, r, ε).

Our goal is to find an efficiently computable (a, r, ε)-oracle on $G = \mathbb{R}^m$ that hides a subgroup H of $\mathbb{R}^m$ which reveals a generator of the input principal ideal. Then it can be used with the HSP algorithm of [7] to find a totally positive generator of a principal ideal in a totally real field in polynomial time in n, $\log(|\Delta|)$, $\log(a)$, $\log(\frac{1}{\varepsilon})$ and r, where Δ is the discriminant of the field.

5.1 Review of the HSP algorithm of EHKS

To compute the unit group, EHKS used a function of the form $f(x) = |e^x \cdot \mathcal{O}\rangle$, where $e^x \cdot \mathcal{O}$ is the lattice generated by the elements of the form $e^x \cdot \omega_i$ for $\mathcal{O} = \sum_i \mathbb{Z}\omega_i$. Such a function hides the unit group of the order $\mathcal{O}$ because $f(x + u) = f(x)$ if and only if $e^u \cdot \mathcal{O} = \mathcal{O}$, which means that e^u is a unit in $\mathcal{O}$. It is derived from a function $f_G \colon G \subseteq \mathbb{R}^m \to \{\text{quantum states}\}$, where G is a hyperplane containing H. They show that if f_G is an (a, r, ε)-oracle on G that hides H, then it can be extended to $f \colon \mathbb{R}^m \to \{\text{quantum states}\}$ satisfying (i), (ii) and (iii). The first step of the description of a function hiding the unit group is to find a classical function f_c on a certain hyperplane G; then we compose it with a quantum encoding f_q, and finally, we extend $f_G = f_q \circ f_c$ to a function f on $\mathbb{R}^m$ that satisfies (i), (ii) and (iii).

Classical function The function F used by CGS is very similar to the classical oracle f_c used in [7]. The latter is defined by

$$\begin{aligned} G \subseteq \mathbb{R}^m &\xrightarrow{f_c} \{\text{lattices in } \mathbb{R}^k\}, \\ v &\longrightarrow e^v \cdot \mathcal{O}. \end{aligned}$$

Here $G \subseteq \mathbb{R}^{r_1+r_2} \times (\mathbb{Z}/2\mathbb{Z})^{r_1} \times (\mathbb{R}/\mathbb{Z})^{r_2}$ is the hyperplane such that $\sum_{i \le r_1+r_2} v_i = 0$. In particular, it contains the elements x of the number field K such that $\mathcal{N}(x) = \pm 1$ via the correspondence

$$x \leftrightarrow (\log|\sigma_1(x)|, \dots, \log|\sigma_{r_1+r_2}(x)|, \operatorname{sign}(\sigma_1(x)), \dots, \operatorname{sign}(\sigma_{r_1}(x)), \theta_1, \dots, \theta_{r_2}),$$

where the θ_j are the phases of the complex embeddings $\sigma_j(x)$. Then, for

$$v = (v_1, \dots, v_{r_1+r_2}, \delta_1, \dots, \delta_{r_1}, \theta_1, \dots, \theta_{r_2}),$$

we define the exponentiation

$$e^v = \left((-1)^{\delta_1} e^{v_1}, \dots, (-1)^{\delta_{r_1}} e^{v_{r_1}}, e^{2i\pi\theta_1} e^{v_{r_1+1}}, \dots, e^{2i\pi\theta_{r_2}} e^{v_{r_1+r_2}}\right) \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}.$$

This can be naturally embedded into $\mathbb{R}^k$ for $k = r_1 + 2r_2$, and in the case of v corresponding to an $x \in K$, we have $e^v = x$. Multiplication in $\mathbb{R}^k$ being considered component-wise, we have $e^v \cdot \mathcal{O} = \mathcal{O}$ if and only if v corresponds to a unit of $\mathcal{O}$. This also implies that $f_c(v_1) = f_c(v_2)$ if and only if $v_1 - v_2 = u$, where e^u is a unit of $\mathcal{O}$.

The quantum encoding. The properties that $f_G = f_q \circ f_c$ has to satisfy also depend on the quantum encoding that was chosen, which is one of the important contributions of EHKS. Let $g_s(\cdot)$ be the Gaussian function $g_s(x) := e^{-\pi\|x\|^2/s^2}$, $x \in \mathbb{R}^k$. For any set $S \subset \mathbb{R}^k$, denote $g_s(S) := \sum_{x \in S} g_s(x)$. Given a lattice L, the quantum encoding f_q maps L to the lattice Gaussian state via

$$\begin{aligned} \{\text{lattices over } \mathbb{R}^k\} &\xrightarrow{f_q} \mathcal{S} \quad (\text{unit vectors in a Hilbert space}), \\ L &\longrightarrow |L\rangle := \gamma \sum_{v \in L} g_s(v)|\mathrm{str}_{v,k}(v)\rangle, \end{aligned}$$

where γ is a normalization factor. Here $|\mathrm{str}_{v,k}(v)\rangle$ is the straddle encoding of a real-valued vector $v \in \mathbb{R}^k$, as defined in [7]. Intuitively, we discretize the space $\mathbb{R}^k$ by a grid $v\mathbb{Z}^k$, and we encode the information about v by a superposition over all grid nodes surrounding v. Specifically, for the one-dimensional case, the straddle

encoding of a real number is

$$x \in \mathbb{R} \mapsto |\mathrm{str}_v(x)\rangle := \cos\left(\frac{\pi}{2}t\right)|j\rangle + \sin\left(\frac{\pi}{2}t\right)|j+1\rangle,$$

where $j := \lfloor \frac{x}{v} \rfloor$ denotes the nearest grid point no bigger than x, and $t := \frac{x}{v} - j$ denotes the (scaled) offset. Repeating this for each coordinate of $v = (v_1, \ldots, v_n)$, we get $|\mathrm{str}_{v,k}(v)\rangle := \bigotimes_{i=1}^{n} |\mathrm{str}_v(v_i)\rangle$. To analyze our function hiding generators of a principal ideal $\mathfrak{a}$, we rely on the properties of the quantum encoding of the function hiding the unit group of $\mathcal{O}$.

An (a, r, ε)-oracle on $\mathbb{R}^n$. We will only be concerned with totally real fields, where $r_1 = n$ and $r_2 = 0$, and only *positive* units will be relevant for our purpose. We later restrict our discussion to this special case, which is much simpler since we do not need to consider the complex coordinates. We use $f_{\mathbb{R},\mathrm{c}}$ to denote the classical part (instead of f_{c}) to indicate this special case. Let $f_{\mathbb{R},\mathrm{c}}$ be the classical oracle defined by

$$G \xrightarrow{f_{\mathbb{R},\mathrm{c}}} \{\text{lattices in } \mathbb{R}^n\},$$
$$v \longrightarrow e^v \cdot \mathcal{O}.$$

Here $G \subseteq \mathbb{R}^n$ is the hyperplane such that $\sum_{i \le n} v_i = 0$. In particular, an element x of the number field K such that $\mathcal{N}(x) = 1$ leads to

$$x \mapsto (\log|\sigma_1(x)|, \ldots, \log|\sigma_n(x)|).$$

Then, for $v = (v_1, \ldots, v_n)$, we define the exponentiation $e^v = (e^{v_1}, \ldots, e^{v_n}) \in \mathbb{R}^n$. Multiplication in $\mathbb{R}^n$ being considered component-wise, we have $e^v \cdot \mathcal{O} = \mathcal{O}$ if and only if v corresponds to a unit of $\mathcal{O}$. This also implies that $f_{\mathbb{R},\mathrm{c}}(v_1) = f_{\mathbb{R},\mathrm{c}}(v_2)$ if and only if $v_1 - v_2 = u$, where u is a totally positive unit of $\mathcal{O}$.

Proposition 5.2 ([7, Theorem 5.7]). *$f_{\mathbb{R}} := f_{\mathrm{q}} \circ f_{\mathbb{R},\mathrm{c}}$ is an (a, r, ε)-oracle on $\mathbb{R}^n$ with*

$$a = \frac{\sqrt{\pi n}s}{4v} + 1, \quad \varepsilon = \frac{3}{4}, \quad r = \log\left(1 + (s\sqrt{n})^{n-1} 2v\sqrt{n}\right)$$

and grid parameters $s = 2^{2n}\sqrt{n|\Delta|}$, $v = \frac{1}{4n(s\sqrt{n})^{2n}}$, where Δ is the discriminant of the field.

5.2 Computing a generator of a principal ideal in a totally real field

In this section, we assume that we are given the $\mathbb{Z}$-basis of a principal ideal $\mathfrak{a}$ of an order $\mathcal{O}$ in a totally real field K of degree n. Moreover, we assume that $\mathfrak{a}$ has a totally positive generator. We show that there is a polynomial time algorithm to compute $(\log|g|_1, \ldots, \log|g|_n)$, where g is a totally positive generator of $\mathfrak{a}$, $n = \deg(K)$ and $|g|_i = |\sigma_i(g)| = \sigma_i(g)$ is the i-th Archimedean valuation of g. We reduce this problem to an instance of the HSP, and we use the framework of EHKS. We start from the same classical oracle as the function F defined by CGS which we compose with f_{q} and extend to $\mathbb{R}^n$. The main observation that allows us to reuse the analysis of the oracle $f_{\mathbb{R},\mathrm{c}}$ hiding the (totally positive) unit group in [7] is that $F(v, j) = f_{\mathbb{R},\mathrm{c}}(v - jg)$, where e^g is an arbitrary (totally positive) generator of $\mathfrak{a}$. The classical function we use is the same as the one of CGS

$$G \subseteq \mathbb{R}^n \times \mathbb{Z} \xrightarrow{F} \{\text{lattices in } \mathbb{R}^n\},$$
$$(v, j) \to e^v \cdot \mathcal{O} \cdot \mathfrak{a}^{-j}.$$

The function $f_{\mathrm{q}} \circ F$ can be then extended from G to $\mathbb{R}^m$ while preserving the essential continuity properties that allow us to reuse the framework of EHKSfor the resolution of the continuous HSP. The careful analysis of the properties of $f_{\mathrm{q}} \circ F$ and that of its extension to $\mathbb{R}^n$ lead to Proposition 5.5 which shows that there is a polynomial-time algorithm to find the generator of a principal ideal in a number field.

A function hiding generators of $\mathfrak{a}$. The rest of the section is devoted to analyzing and extending $f_{\mathrm{q}} \circ F$ to a function $f_{\mathfrak{a}}$ defined on $\mathbb{R}^m$ that hides the lattice of the totally positive generators of $\mathfrak{a}$ and satisfies the HSP conditions (i), (ii) and (iii). The formal statement appears in Proposition 5.5, which is proven based on a few intermediate steps (Propositions 5.3 and 5.4). Given $f_{\mathfrak{a}}$, the quantum HSP algorithm of EHKS computes a totally positive generator efficiently.

We start off analyzing the properties of $f_q \circ F$.

Proposition 5.3. *With the F and f_q defined above, $s = 2^{2n}\sqrt{n|\Delta|}$ and $\nu = \frac{1}{4n(s\sqrt{n})^{2n}}$, we have that $f_G := f_q \circ F$ is an (a, r, ε)-oracle on G for*

$$a = \frac{\sqrt{\pi n}s}{4\nu} + 2, \quad \varepsilon = \frac{3}{4}, \quad r = \log(1 + (s\sqrt{n})^{n-1}2\nu\sqrt{n}).$$

Proof. Let us fix a generator g of $\mathfrak{a}$ and its corresponding $(v_g, 1) \in G \subseteq \mathbb{R}^n \times \mathbb{Z}$. The main observation leading to the result is that $F(v, j) = f_c(v - jv_g)$, and therefore $|f_G(v, j)\rangle = |f_G(v - jv_g)\rangle$.

(a) Lipschitz condition. If $j_1 \neq j_2$, then $d_G((v_1, j_1), (v_2, j_2)) \geq 1$, while, at the same time,

$$\| |f_G(v_1, j_1)\rangle - |f_G(v_2, j_2)\rangle \| \leq 2.$$

So, in this case,

$$\| |f_G(v_1, j_1)\rangle - |f_G(v_2, j_2)\rangle \| \leq 2d_G((v_1, j_1), (v_2, j_2)).$$

On the other hand, if $j_1 = j_2 = j$, then

$$\begin{aligned} d_G((v_1, j_1), (v_2, j_2)) &= d_{\mathbb{R}^n}(v_1, v_2) \\ &= d_{\mathbb{R}^n}(v_1 - jv_g, v_2 - jv_g) \\ &= d_{\mathbb{R}^n}(v_1 - j_1v_g, v_2 - j_1v_g) \\ &\geq a\| |f_G(v_1 - j_1v_g)\rangle - |f_G(v_2 - j_2v_g)\rangle \| \\ &= a\| |f_G(v_1, j_1)\rangle - |f_G(v_2, j_2)\rangle \| \end{aligned}$$

for $a = \frac{\sqrt{\pi n}s}{4\nu} + 1$. Therefore, the Lipschitz condition is always satisfied for $a = \frac{\sqrt{\pi n}s}{4\nu} + 2$.

(b) The (r, ε) condition. We simply need to notice that $d_{G/\Lambda_{\alpha'}}((v_1, j_1), (v_2, j_2)) \leq d_{\mathbb{R}^n/U^+}(v_1 - j_1v_g, v_2 - j_2v_g)$, where $U^+ \subseteq \mathbb{R}^n$ denotes the vectors $u \in \mathbb{R}^n$ such that e^u is a totally positive unit of K. It is immediate that the periods of $f_{\mathbb{R}}$ are U^+, and according to Proposition 5.2, $f_{\mathbb{R}}$ is an (a, r, ε)-oracle for $a = \frac{\sqrt{\pi n}s}{4\nu}$, $\varepsilon = \frac{3}{4}$, and $r = \log(1 + (s\sqrt{n})^{n-1}2\nu\sqrt{n})$. We use the properties of $f_{\mathbb{R}}$ to analyze the behavior of $f_{\mathfrak{a}}$.

$$\begin{aligned} d_{G/\Lambda_{\alpha'}}((v_1, j_1), (v_2, j_2)) &= \inf_{\substack{u \in U^+ \\ j \in \mathbb{Z}}} \|(v_1, j_1) - (v_2, j_2) - (jv_g, j) - (u, 0)\| \\ &\leq \inf_{u \in U^+} \|(v_1 - j_1v_g, 0) - (v_2 - j_2v_g, 0) - (u, 0)\| \quad \text{(by choosing } j = j_1 + j_2) \\ &= d_{\mathbb{R}^n/U^+}(v_1 - i_1v_g, v_2 - i_2v_g). \end{aligned}$$

This means that, for $r = \log(1 + (s\sqrt{n})^{n-1}2\nu\sqrt{n})$ and $\varepsilon = \frac{3}{4}$, if

$$d_{G/\Lambda_{\alpha'}}((v_1, j_1), (v_2, j_2)) \geq r,$$

then $d_{\mathbb{R}^n/U^+}(v_1 - i_1v_g, v_2 - i_2v_g) \geq r$ as well, and then, necessarily,

$$\langle f_G(v_1, j_1)|f_G(v_2, j_2)\rangle = \langle f_{\mathbb{R}}(v_1 - j_1v_g)|f_{\mathbb{R}}(v_2 - j_2v_g)\rangle \leq \varepsilon. \qquad \square$$

Reduction to the case $G = \mathbb{R}^m$. We described an (a, r, ε)-oracle f_G on a hyperplane G of $\mathbb{R}^n \times \mathbb{Z}$ hiding the lattice $\Lambda_{\alpha'}$ for $a = \frac{\sqrt{\pi n}s}{4\nu} + 2$, $\varepsilon = \frac{3}{4}$, and $r = \log(1 + (s\sqrt{n})^{n-1}2\nu\sqrt{n})$. To apply [7, Theorem 6.1], we need a function $f_{\mathfrak{a}}$ on $\mathbb{R}^m$ for some m that hides the lattice $\Lambda_{\alpha'}$ and which is an $(\overline{a}, \overline{r}, \overline{\varepsilon})$-oracle in $\mathbb{R}^m$ for some $\overline{a}, \overline{r}, \overline{\varepsilon}$, not necessarily equal to a, r, ε. A general guideline for performing such a task is given in [7, Section 6.1]. By following it, we find such a function $f_{\mathfrak{a}}$, and we can apply the quantum algorithm of [7] to derive $\Lambda_{\alpha'}$, thus obtaining a totally positive generator for $\mathfrak{a}$.

First of all, we can easily turn f_G defined on the hyperplane G into a function defined over $\mathbb{R}^{n-1} \times \mathbb{Z}$ with the intermediate operation

$$\begin{aligned} \mathbb{R}^n \times \mathbb{Z} &\xrightarrow{\phi} G, \\ (v, j) &\to \left(v_1, \ldots, v_{r_1+r_2-1}, -\sum_i v_i + j\log|\mathcal{N}(\mathfrak{a})|, j\right). \end{aligned}$$

Proposition 5.4. *Assume f_G is an (a, r, ε)-oracle hiding $\Lambda_{\alpha'}$ on G; then the function defined by $f_{G_1} := f_G \circ \phi$ is an (a_1, r, ε)-oracle hiding $\Lambda_{\alpha'}$ on $G_1 := \mathbb{R}^{n-1} \times \mathbb{Z}$, where $a_1 = a\sqrt{6(r_1 + r_2 - 1)}\log|\mathcal{N}(\mathfrak{a})|$.*

Proof. The fact that the (r, ε)-condition is preserved is obvious because we are dropping one coordinate. This means that if the distance in $G_1 = \mathbb{R}^{n-1} \times \mathbb{Z}$ (modulo $\Lambda_{\alpha'}$) is greater than r, then so is the distance in G (modulo $\Lambda_{\alpha'}$), and therefore the inner product of the two states has to be less than ε. The Lipschitz condition comes from the fact that $\||f_{G_1}(x)\rangle - |f_{G_1}(y)\rangle\|^2 = \||f_G(\phi(x))\rangle - |f_G(\phi(y))\rangle\|^2 \leq a^2 d^2(\phi(x), \phi(y))$ and that

$$\begin{aligned}
a^2 d^2(\phi(x), \phi(y)) &= a^2\Big(\sum_{k \leq r_1+r_2-1} v_k^2 + \Big(j\log|\mathcal{N}(\mathfrak{a})| - \sum_k v_k\Big)^2 + j^2\Big) \quad (\text{where } (v, j) := x - y)\\
&= a^2\Big(\sum_{k \leq r_1+r_2-1} v_k^2 + j^2\log^2|\mathcal{N}(\mathfrak{a})| + \sum_{k \leq r_1+r_2-1} v_k^2\\
&\qquad - 2j\log|\mathcal{N}(\mathfrak{a})|\Big(\sum_{k \leq r_1+r_2-1} v_k\Big) + 2\sum_{k \neq l \leq r_1+r_2-1} v_k v_l + j^2\Big)\\
&\leq 6a^2(r_1 + r_2 - 1)\log^2|\mathcal{N}(\mathfrak{a})|\Big(\sum_{k \leq r_1+r_2-1} v_k^2 + j^2\Big)\\
&= \big(6a^2(r_1 + r_2 - 1)\log^2|\mathcal{N}(\mathfrak{a})|\big)d_{G_1}^2(x, y).
\end{aligned}$$

□

We have now a function on $\mathbb{R}^k \times \mathbb{Z}^l$ for $k = n - 1$ and $l = 1$ that hides $\Lambda_{\alpha'}$ and that is an (a_1, r, ε)-oracle on $\mathbb{R}^k \times \mathbb{Z}^l$. Following the guidelines of [7, Section 6.1], we can turn it into an $(\overline{a}, \overline{r}, \overline{\varepsilon})$-oracle $f_{\mathfrak{a}}$ on $\mathbb{R}^{k+l}$ that hides $\Lambda_{\alpha'}$. To do so, we define

$$|f_{\mathfrak{a}}(\mathbf{x}, x_1, \ldots, x_l)\rangle := \sum_{z_1,\ldots,z_l \in \{0,1\}} \Big(\bigotimes_{j=1}^{l} |\psi(x_j, z_j)\rangle\Big) \otimes |f_{G_1}(\mathbf{x}, s(x_1, z_1), \ldots, s(x_l, z_l))\rangle,$$

where $s(x, z) = \lfloor \frac{x}{\lambda} \rfloor + z$, $|\psi(x, z)\rangle = \cos(\frac{\pi}{2})\mathrm{str}_\nu(t)$ with $t = \frac{x}{\lambda} - s(x, z)$ for a lower bound λ on the shortest vector of $\Lambda_{\alpha'}$.

Proposition 5.5. *The function $f_{\mathfrak{a}}$ hides the lattice $\Lambda_{\alpha'}$ and satisfies conditions (i), (ii) and (iii) for $G = \mathbb{R}^n$ and the parameters $\overline{a}, \overline{r}, \overline{\varepsilon}$ defined by*

$$\begin{aligned}
\overline{a}^2 &= a_1^2 + l\Big(\frac{\pi}{2\nu\lambda}(1 + \nu)\Big)^2\\
&= 6(r_1 + r_2 - 1)\log^2|\mathcal{N}(\mathfrak{a})|\Big(\frac{\sqrt{\pi n}s}{4\nu} + 2\Big)^2 + l\Big(\frac{\pi}{2\nu\lambda}(1 + \nu)\Big)^2,\\
\overline{r}^2 &= \big(\log(1 + (s\sqrt{n})^{n-1}2\nu\sqrt{n})\big)^2 + l(2\nu\lambda)^2,\\
\overline{\varepsilon} &= \frac{3}{4}.
\end{aligned}$$

Proof. See [7, Section 6.1]. It shows that, by the transformation above, the new function is a valid HSP instance with

$$\overline{a}^2 = a^2 + l\Big(\frac{\pi}{2\nu\lambda}(1 + \nu)\Big)^2, \quad \overline{r}^2 = r^2 + l(2\nu\lambda)^2, \quad \overline{\varepsilon} = \varepsilon.$$

□

6 Computing a short generator of a principal ideal in $\mathbb{Q}(\zeta_{2^s})$

In this section, we show how to reduce the search for a small generator of an input ideal I in $K = \mathbb{Q}(\zeta_{2^s})$ to the computation of a totally positive generator of a principal ideal $\mathfrak{a}$ (which depends on I) in $K^+ = \mathbb{Q}(\zeta_{2^s} + \zeta_{2^s}^{-1})$. The main ingredient of this reduction is the norm equation resolution of the Howgrave-Graham–Szydlo algorithm [13]. This reduction seems natural, but no formal procedure (and analysis) was available [16]. The main steps of the whole attack are:

(i) Create the ideal $\mathfrak{a} = (\mathcal{N}_{K/K^+}(g)) \subseteq K^+$, where g is a short generator of I.
(ii) Find a generator α' of $\mathfrak{a}$ with the quantum algorithm of Section 5.2.
(iii) Find a short generator g' of I by using α' and a $\mathbb{Z}$-basis of I.

Input: $K = \mathbb{Q}(\zeta_{2^s})$ and an ideal $I \subseteq \mathbb{Z}[\zeta_{2^s}]$.
Output: $\mathfrak{a} = I\bar{I} \cap K^+$.

1: $I^+ \leftarrow I\bar{I}$.
2: Compute the intersection $\mathfrak{a}$ of I^+ and $\mathbb{Z}[\zeta_{2^n} + \zeta_{2^n}^{-1}]$ with [5, Algorithm 1.4.5].
3: **return** $\mathfrak{a}$.

Algorithm 1: Creation of $\mathfrak{a} = I\bar{I} \cap K^+$.

The first step consists in finding a $\mathbb{Z}$-basis of the ideal $I\bar{I} \cap K^+$. We can easily find a basis of the ideal $I\bar{I}$ of K, and we intersect it with the ring of integers of the subfield K^+ of K by using [5, Algorithm 1.4.5]. When I is principal and generated by g, then $\mathfrak{a}$ is principal as well and generated by $\mathcal{N}_{K/K^+}(g) = g\bar{g}$.

The ideal $\mathfrak{a}$ of the totally real field K^+ is principal and generated by a totally positive generator $g\bar{g}$. Therefore, it satisfies the conditions of the quantum polynomial-time algorithm for computing a totally positive generator of an ideal in a totally real field described in Section 5.2. The output of this procedure is a rational approximation of the real vector $\mathrm{Log}(\alpha')$, where α' is a totally positive generator of $\mathfrak{a}$. We want to lift this generator to obtain a generator of I. We need to assert two important properties:

- α' is of the form $\mathcal{N}_{K/K^+}(g')$, where g' is a generator of I.
- α' is short enough to be written on the integral basis of K^+ in polynomial time.

The surjectivity of the relative norm map does not necessarily hold true. As a matter of fact, we can only prove it under Conjecture 3.1 (Weber conjecture) which states that the class number of K^+ is 1.

Proposition 6.1 (under Conjecture 3.1). *Let $K = \mathbb{Q}(\zeta_{2^s})$, $I = (g)$ be an ideal of $\mathbb{Z}[\zeta_{2^s}]$ and $\mathfrak{a} = I\bar{I} \cap K^+$. Then every totally positive generator of $\mathfrak{a}$ is of the form $\mathcal{N}_{K/K^+}(g')$ for g' a generator of I.*

Proof. The ideal $\mathfrak{a}$ is generated by at least one totally positive number (i.e., the image $\mathcal{N}_{K/K^+}(g)$ of a generator g of I by the relative norm map). Then, from [20], we know that the totally positive units are exactly the squares of units (see also [14, Intro]), which are also the norms of the units of $\mathbb{Z}[\zeta_{2^s}]$ that are in K^+. Let α' be a totally positive generator of $\mathfrak{a}$. Then the two totally positive generators $\alpha', \mathcal{N}_{K/K^+}(g)$ of $\mathfrak{a}$ differ by a totally positive unit, hence a square, and hence the image of a unit u of $\mathbb{Z}[\zeta_{2^s}] \cap K^+$ by the norm map, i.e., $\alpha' = \mathcal{N}_{K/K^+}(u)\mathcal{N}_{K/K^+}(g) = \mathcal{N}_{K/K^+}(ug)$, which is the image of a generator ug of I by the relative norm map. □

The vector $\mathrm{Log}(\alpha')$ returned by the quantum algorithm of Section 5.2 has polynomial size, but the representation of α' over an integral basis of K^+ may have exponential size. Therefore, the resolution of the norm equation by the method of Howgrave-Graham and Szydlo [13] with input α' may take exponential time. We need to find another totally positive generator α of $\mathfrak{a}$ with reasonable size. We know that $\alpha := \mathcal{N}_{K/K^+}(g)$ where g is the secret short generator of I has a poly-size representation on the integral basis of K^+. Therefore, we use the method of Cramer et al. [6] to derive a short generator of $\mathfrak{a}$ before applying the algorithm of Howgrave-Graham and Szydlo [13].

Proposition 6.2 (under Conjecture 3.1). *The element α computed in step 7 of Algorithm 2 has a poly-size representation on the integral basis of K^+.*

Proof. This directly follows from the analysis of Babai's round-off method (used in steps 3–4) by Cramer et al. [6]. The only difference is that we work with the lattice of the $\mathrm{Log}(\mathcal{N}_{K/K^+}(u_i))$ instead of that of $\mathrm{Log}(u_i)$. Let g' be the generator of I such that $\mathcal{N}_{K/K^+}(g') = \alpha'$. According to the analysis of [6], we know that if we find $(x_i)_{i \le n/2} \in \mathbb{R}^{n/2}$ such that $\mathrm{Log}(g') = \sum_i x_i \mathrm{Log}(u_i)$ and then perform the operation $x_i \leftarrow -\lfloor x_i \rceil$, then $g = g' \prod_i u_i^{x_i}$ is a generator of I that satisfies $\|g\| = e^{n^{1/2+o(1)}} \mathcal{N}(I)$. In particular, its representation on an integral basis has polynomial size. The $(x_i)_{i \le n/2}$ calculated in step 3 are such $\mathrm{Log}(\alpha') = 2\,\mathrm{Log}(g') = \sum_{i \le n/2} 2x_i \mathrm{Log}(u_i)$. These ensure that $g = g' \prod_i u_i^{x_i}$ is a short generator of I, and

$$\alpha = \alpha' \prod_i \mathcal{N}_{K/K^+}(u_i)^{x_i} = \mathcal{N}_{K/K^+}\left(g' \prod_i u_i^{x_i}\right) = \mathcal{N}_{K/K^+}(g)$$

is the relative norm of the small generator g of I. It is therefore a short generator of $\mathfrak{a}$. □

Input: $K = \mathbb{Q}(\zeta_{2^s})$, an ideal $I \subseteq \mathbb{Z}[\zeta_{2^s}]$, $\mathfrak{a} = I\bar{I} \cap K^+$ and $\mathrm{Log}(\alpha')$ for a totally positive generator α' of $\mathfrak{a}$.
Output: A short generator g of I.

1: Compute $\alpha_0, \ldots, \alpha_l$, where l is polynomial and each α_i has a polynomial size representation on the integral basis of K^+ such that $\alpha' = \alpha_0\alpha_1^2 \ldots \alpha_k^{2^k}$ by using the compact representation algorithm of [2, Section 5].
2: $u_i \leftarrow (\frac{\zeta_{2^s}^i - 1}{\zeta_{2^s} - 1})$ for $i \in \mathbb{Z}_n^*$.
3: Find $(x_i)_{i \leq n/2} \in \mathbb{R}^{n/2}$ such that $\mathrm{Log}(\alpha') = \sum_{i \leq n/2} 2x_i \mathrm{Log}(u_i)$.
4: $x_i \leftarrow -\lfloor x_i \rceil$ for $i \leq \frac{n}{2}$.
5: Compute primes $p_1, \ldots, p_m$ prime ideals such that $\log(p_i) \in \mathrm{Poly}(n)$ and $\prod_j p_j > \mathcal{N}(I)$.
6: Compute $\alpha_0\alpha_1^2 \ldots \alpha_k^{2^k} \prod_i \mathcal{N}_{K/K^+}(u_i)^{x_i} \bmod (p_j)$ for all $j \leq m$.
7: Reconstruct $\alpha := \alpha' \prod_i \mathcal{N}_{K/K^+}(u_i)$ on the integral basis of K^+ by the Chinese remainder theorem.
8: Compute g such that $\mathcal{N}_{K/K^+}(g) = \alpha$ with the algorithm of Howgrave-Graham and Szydlo [13].
9: **return** g.

Algorithm 2: Lift of the solution in K^+.

Corollary 6.3. *Algorithm 2 runs in polynomial time and returns a generator g of I such that $\|g\| = e^{n^{1/2+o(1)}}\mathcal{N}(I)$.*

Proof. All steps run in polynomial time. In addition, according to the proof of Proposition 6.2, we have the guarantee that step 7 produces the relative norm α of a small generator g of I. Then the solution to the relative norm equation in step 8 yields the desired short element. □

7 Conclusion and significance

We described a quantum polynomial time algorithm to recover a short generator of an ideal in $\mathbb{Q}(\zeta_{2^s})$. We showed that it derives from the results of [7] in a rather straightforward way. It is a significant result for post-quantum cryptography. Indeed, together with the reduction from the short-PIP to the PIP originally observed by CGS and later proved by Cramer et al. [6], it is enough to attack cryptosystems based on the hardness of finding a short generator of a principal ideal in $\mathbb{Q}(\zeta_{2^s})$ in quantum polynomial time. These include the multilinear maps of Garg, Gentry and Halevi [8] and the fully homomorphic encryption scheme of Smart and Vercauteren [18].

Strictly speaking, the algorithm we discussed in Section 5 does not solve the standard principal ideal problem with absolute certainty since the algorithm cannot decide if an input ideal is principal (it rather takes as promise that it is principal). Further generalizations of the methods of [7] will lead to the resolution of related problems in number theory in arbitrary fields including the PIP, the computation of the ideal class group, the computation of S-units, or the resolution of norm equations.

A Previous algorithms for solving the HSP

A.1 Shor's factoring algorithm

Post-quantum cryptography really became a concern when Shor proposed a quantum algorithm to factor integers [17]. Moreover (as we see in the next section), this algorithm extends to the discrete logarithm problem in any group. An RSA integer N is an integer satisfying $N = pq$ where p, q are distinct prime numbers. The problem of factoring an RSA integer reduces to an instance of the so-called *hidden subgroup problem* (HSP).

Definition A.1 (Hidden subgroup problem over $\mathbb{Z}$). Given $f\colon \mathbb{Z} \mapsto X$ for a finite set X such that there exists a subgroup $H \leq \mathbb{Z}$ with

$$f(x+g) = f(x) \quad \text{for all } x \in \mathbb{Z} \quad \text{if and only if} \quad g \in H,$$

the hidden subgroup problem is the task of finding H given oracle access to f. This means finding r such that $H = r\mathbb{Z}$.

We want to factor an RSA integer $N = pq$. Let a be coprime with N (if $a \mid N$, the factorization problem is solved) and

$$\begin{aligned} \mathbb{Z} &\xrightarrow{f} \mathbb{Z}/N\mathbb{Z}, \\ x &\to a^x \bmod N. \end{aligned}$$

A solution to the HSP with f yields r, the order of $a \bmod N$, and if a is a square, we get

$$(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N.$$

This means that $N \mid (a^{r/2} - 1)(a^{r/2} + 1)$, and $\gcd(N, a^{r/2} - 1)$ may yield a non-trivial factor of N.

Let us sketch the resolution of this instance of the HSP. The first step relies on the fact that if f is efficiently computable classically, one can create an efficient quantum algorithm to evaluate f in superposition. This yields a circuit for

$$\frac{1}{\sqrt{M}} \sum_{x \in \mathbb{Z}_M} |0\rangle|x\rangle \xrightarrow{f} \frac{1}{\sqrt{M}} \sum_{x \in \mathbb{Z}_M} |f(x)\rangle|x\rangle.$$

The other main ingredient we need to use in Shor's algorithm is the so-called quantum Fourier transform (QFT) over $\mathbb{Z}_M$ (for a large enough M). Let $\omega_M = e^{2\pi i/M}$, the QFT is the quantum algorithm realizing

$$\mathrm{QFT}_M\colon |x\rangle \mapsto \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M} \omega_M^{x \cdot y} |y\rangle.$$

If we apply the QFT to the second register of the previous state, we obtain

$$\begin{aligned} \frac{1}{\sqrt{M}} \sum_{x \in \mathbb{Z}_M} |f(x)\rangle|x\rangle \xrightarrow{\mathrm{QFT}_N} & \frac{1}{\sqrt{M}} \sum_{x \in \mathbb{Z}_M} |f(x)\rangle \Big(\frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M} \omega_M^{x \cdot y} |y\rangle \Big) \\ &= \frac{1}{M} \sum_{y \in \mathbb{Z}_M} \Big(\sum_{x \in \mathbb{Z}_M} \omega_M^{x \cdot y} |f(x)\rangle \Big) |y\rangle := \frac{1}{M} \sum_{y \in \mathbb{Z}_M} |\phi_y\rangle \otimes |y\rangle. \end{aligned}$$

We can easily verify that the $|\phi_y\rangle$ are orthogonal vectors satisfying $\frac{1}{M} \sum_y \langle \phi_y | \phi_y \rangle = 1$. We perform a measurement on the second register, which yields the value y with probability $\frac{1}{M^2} \langle \phi_y, \phi_y \rangle \approx \frac{1}{M} \sum_{k \leq M/r} (\omega_M^{y \cdot r})^k$.

$$\begin{aligned} \Pr[\text{measure } y] &= \frac{1}{M^2} \Big(\sum_{x_1 \in \mathbb{Z}_M} \langle f(x_1) | \omega_M^{-x_1 \cdot y} \Big) \Big(\sum_{x_2 \in \mathbb{Z}_M} \omega_M^{x_2 \cdot y} | f(x_2) \rangle \Big) \\ &= \frac{1}{M^2} \sum_{x_1, x_2 \in \mathbb{Z}_M} \omega_M^{y(x_2 - x_1)} \langle f(x_1) | f(x_2) \rangle \\ &= \frac{1}{M^2} \sum_{x_1, x_2 \in \mathbb{Z}_M, f(x_1) = f(x_2)} \omega_M^{y(x_2 - x_1)} \\ &\approx \frac{1}{M} \sum_{k \leq M/r} (\omega_M^{y \cdot r})^k. \end{aligned}$$

Then if $\frac{y}{M}$ is close to an element of the form $\frac{l}{r}$, then the above probability is high, and if not, then the probability of measuring y is low. If $\frac{y}{M}$ is a good enough approximation of an element of the form $\frac{l}{r}$, then $\frac{l}{r}$ belongs to the list of convergents of the continued fraction expansion of $\frac{y}{M}$, which is computed in classical polynomial time. Then we recover the period r and thus solve the problem. The probability of successfully recovering r is in $\frac{1}{\Omega(\log(\log(N)))}$ (there is a constant probability variant consisting in repeating this procedure twice). This is not the only variant of Shor's algorithm for factoring algorithms. Alternatively, a partial measurement is performed on the $f(x)$ register before applying the quantum Fourier transform. The quantum algorithm for solving the PIP sketched by CGS follows closely the HSP variant that we described.

A.2 The hidden subgroup problem in higher dimension

The hidden subgroup problem has a straightforward generalization in higher dimension. Many problems in algebraic number theory can be reduced to an instance of the HSP.

Definition A.2 (Hidden subgroup problem over $\mathbb{Z}^m$). Given $f\colon \mathbb{Z}^m \mapsto X$ for a set X such that there exists a subgroup $H \leq \mathbb{Z}^n$ with

$$f(x+g) = f(x) \quad \text{if and only if} \quad g \in H,$$

the hidden subgroup problem is the task of finding H given oracle access to f.

The discrete logarithm problem is the search for $h \in \mathbb{Z}$ such that $b = a^h$, where a, b are given elements of a finite group $\mathcal{G}$. This can be reduced to an instance of the hidden subgroup problem in $\mathbb{Z}^2$. We define the function

$$\begin{aligned} \mathbb{Z}\times\mathbb{Z} &\xrightarrow{f} \mathcal{G},\\ (x,y) &\to a^x b^{-y}. \end{aligned}$$

The periods of this function are the subgroup $G = \mathbb{Z}(1,h) + \mathbb{Z}(r,0)$, where r is the order of a. Finding the subgroup G hidden by f solves our problem. The analysis we carried on to solve the HSP in $\mathbb{Z}$ generalizes in higher dimension by using the tensor product of the QFT

$$\mathrm{QFT}_M^{\otimes m}\colon |\mathbf{x}\rangle \mapsto \frac{1}{\sqrt{M^m}} \sum_{\mathbf{y}\in\mathbb{Z}_M^m} \omega_M^{\mathbf{x}\cdot\mathbf{y}} |\mathbf{y}\rangle,$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_M^m$, and $|\mathbf{x}\rangle$ is an encoding of the vector $\mathbf{x}$. Note that, here again, M has to be chosen large enough with respect to the typical values we are calculating. As for factoring, applying the QFT yields a state of the form $\frac{1}{M^m}\sum_{\mathbf{y}\in\mathbb{Z}_M^m} |\phi_{\mathbf{y}}\rangle \otimes |\mathbf{y}\rangle$, and we measure the vector $\mathbf{y} \in \mathbb{Z}_M^m$ with probability

$$\frac{1}{M^{2m}} \sum_{\mathbf{x}_1,\mathbf{x}_2\in\mathbb{Z}_M^m, f(x_1)=f(x_2)} \omega_M^{\mathbf{y}\cdot(\mathbf{x}_2-\mathbf{x}_1)} = \frac{1}{M^m} \sum_{\mathbf{u}\in\mathcal{L}\cap\mathbb{Z}_M^m} \omega_M^{\mathbf{y}\cdot\mathbf{u}},$$

where $\mathcal{L} \subseteq \mathbb{Z}^m$ is the hidden subgroup (a lattice) we are looking for. This sum is larger when $\mathbf{y}\cdot\mathbf{x}$ is an integer, that is, when $\frac{\mathbf{y}}{M} \in \mathcal{L}^*$. It can be shown that, when $\frac{\mathbf{y}}{M}$ is close enough to a point in the dual of $\mathcal{L}$, then it has a high probability of being sampled. This generalizes the factoring algorithm presented in the previous section which relies on the sampling of elements in the dual of the lattice $\mathcal{L} = r\mathbb{Z}$. After finding a good approximation of the dual lattice $\mathcal{L}^*$, we use classical linear algebra methods to compute $\mathcal{L}$.

To solve other number theoretic problems, we need to work with approximations of real numbers. This occurs for example in Hallgren's method [12] to solve the Pell equation in quantum polynomial time. The discretization method used by Hallgren was generalized by Hales [10] to derive a solution to the hidden subgroup problem over (approximations of) the reals. To compute the ideal class group, the unit group and to solve instances of the principal ideal problem in number fields of higher degree, the usual approach is to first reduce the problem to the task of finding the periods of a function f defined over $\mathbb{R}^m$ for some m, and then find these periods with an algorithm for solving the HSP. For example, Hallgren [11, Section 3.1] described a unit group algorithm in a field K consisting of finding the periods of the function

$$\begin{aligned} \mathbb{R}^r &\xrightarrow{f} \mathcal{I}\times\mathbb{R}^r,\\ x &\to \left(\frac{1}{\mu}\mathcal{O}, x - \mathrm{Log}(\mu)\right), \quad \text{where} \quad \mu\in\mathcal{O} \text{ minimizes } \|\mathrm{Log}(\mu) - x\|. \end{aligned}$$

Here r is the rank of the unit group and $\mathrm{Log}(\mu) = (\log|\sigma_1(\mu)|, \ldots, \log|\sigma_r(\mu)|)$ is the vector of the first r logarithms of the Archimedean embeddings of μ. Since this function relies on the search for a minimum in $\mathcal{O}$, its evaluation costs exponential time in the degree, thus restricting its use for classes of number field with fixed degree. In the same paper, Hallgren [11] described quantum polynomial-time algorithms for the unit group, the class group and the principal ideal problem in classes of fixed-degree number fields.

A necessary condition to ensure that these problems can be solved in polynomial time is that they reduce to the search for the periods of a function that is efficiently computable. The evaluation of the function described above is not polynomial in the degree of the extension, which is one reason why the overall algorithm does not run in polynomial time in k. The other obstruction lies within the resolution of the subsequent instance of the HSP. Indeed, the method used in [11] to solve the hidden subgroup problem in $\mathbb{R}^m$ does not seem to run in polynomial time with respect to m. It relies on the creation and the measurement of the state

$$|\psi\rangle = \frac{1}{\sqrt{|\mathcal{L}_q|}} \frac{1}{\sqrt{M}} \sum_{\mathbf{x}\in\mathbb{Z}_M^m} \sum_{\mathbf{u}\in\mathcal{L}_q} \omega_M^{\mathbf{x}\cdot\lceil N\mathbf{u}\rfloor} |\mathbf{x}\rangle, \quad \text{where} \quad \mathcal{L}_q = \mathcal{L} \cap [0,q]^m.$$

Hallgren showed that the probability of measuring $\mathbf{x}$ such that $\frac{\mathbf{x}}{q}$ was $\frac{1}{q}$-close to $\mathcal{L}^*$ was at least $\frac{1}{8\log(\mathrm{disc}(\mathcal{O}))^m}$ (a term corresponding to the zero-filling was omitted, i.e., an artificial enlargement of the size of the bounded region where we perform the QFT to facilitate the analysis). In classes of fixed degree (i.e., when m is fixed), this gives a polynomial time algorithm to solve the HSP. The case of $m \to \infty$ was solved 10 years later by EHKS.

B Towards an analysis of the algorithm of CGS

In this section, we show that if we discretize G at a precision $\frac{1}{N}$ as it is done in [11], then the quantum algorithm of CGS cannot return, in better complexity than 2^n, a vector that is ε-close to $\Lambda_{\alpha'}$ for $\varepsilon = \frac{1}{q}$ and $q \geq n^2\lambda$, where λ is a bound on the size of the vectors in a reduced basis of $\Lambda_{\alpha'}$.

Proposition B.1 (Sampling probability). *Let $N > 0$ be the precision of the discretization of G. We assume that the fingerprint encoding behaves as conjectured in* [4, Section 3.6], *that is,*

- $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = 1$ *if* $\mathbf{x}_2 - \mathbf{x}_1$ *is ε-close to $\mathcal{L}$ for some $\varepsilon < \frac{1}{N}$,*
- $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = 0$ *otherwise.*

The probability of drawing a rational approximation that is $\frac{1}{q}$-close to a vector in $\Lambda^*_{\alpha'}$ for $q \geq (n)^2\lambda$, where $n = \deg(K^+)$, $\Delta = \mathrm{disc}(K^+)$ and where λ is a bound on the size of the vectors in a reduced basis of $\Lambda_{\alpha'}$, is at least

$$P \geq \frac{1}{8(\log(|\Delta|)t)^n} \quad \textit{for any } t \geq 8n.$$

Proof. To bound and discretize G, we need three parameters that were not explicitly given in [4]. The grid has precision $\frac{1}{N}$ for some $N > 0$, and we choose to restrict the QFT to $G \cap [0,q]^n$ for a large enough integer q. We also enlarge the grid by a factor t that will be used to analyze the complexity (this is the so-called zero-filling technique). Let the normalization factor be $M = qtN$. We can identify the discretized and bounded G' with $\mathbb{Z}_M^n$. Then the algorithm is the same as for factoring,

$$\frac{1}{\sqrt{M^n}} \sum_{\mathbf{x}\in\mathbb{Z}_M^n} |0\rangle|x\rangle \xrightarrow{F} \frac{1}{\sqrt{M^n}} \sum_{\mathbf{x}\in\mathbb{Z}_M} |\psi_{\mathbf{x}}\rangle|\mathbf{x}\rangle \xrightarrow{\mathrm{QFT}_M^{\otimes n}} \frac{1}{M^n} \sum_{\mathbf{y}\in\mathbb{Z}_M^n} |\phi_{\mathbf{y}}\rangle \otimes |\mathbf{y}\rangle.$$

We measure $\mathbf{y}$ and hope that it is close enough to a vector in $\Lambda_{\alpha'}$. To analyze this technique, we use the same approach as Hallgren's 2005 paper [11]. As for Shor's factoring algorithm, the probability of drawing $\mathbf{y} \in G'$ (regardless of its properties) is

$$\frac{1}{M^{2n}}\langle\phi_{\mathbf{y}}, \phi_{\mathbf{y}}\rangle = \frac{1}{M^{2n}} \sum_{\mathbf{x}_1,\mathbf{x}_2\in\mathbb{Z}_M^n} \omega_M^{\mathbf{y}\cdot(\mathbf{x}_2-\mathbf{x}_1)} \langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle.$$

Unlike in the exact case where $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle$ is either 1 when $\mathbf{x}_2 - \mathbf{x}_1 \in \mathcal{L}$ and 0 otherwise (here $\mathcal{L} = \Lambda_{\alpha'}$), we are dealing with approximations. We assume that the fingerprint behaves as conjectured in [4, Section 3.6]. We formalize this by $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = 1$ if $\mathbf{x}_2 - \mathbf{x}_1$ is ε-close to $\mathcal{L}$ for some $\varepsilon < \frac{1}{N}$ and $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = 0$ otherwise. For each lattice vector $\mathbf{u} \in \mathcal{L}$, we have $\langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = 1$ for all the $\mathbf{x}_1, \mathbf{x}_2$ such that $\mathbf{x}_2 - \mathbf{x}_1$ is in a ball of radius ε centered around $\mathbf{u}$. So the probability of measuring $\mathbf{y}$ is

$$\frac{1}{M^{2n}} \sum_{\mathbf{x}_1,\mathbf{x}_2\in\mathbb{Z}_M^n} \omega_M^{\mathbf{y}\cdot(\mathbf{x}_2-\mathbf{x}_1)} \langle\psi_{\mathbf{x}_1}|\psi_{\mathbf{x}_2}\rangle = \frac{1}{M^{2n}} \sum_{\mathbf{x}_1\in\mathbb{Z}_M^n} \sum_{\substack{\mathbf{x}_2\in\mathbb{Z}_M^n \\ \mathbf{x}_1-\mathbf{x}_2\in\mathcal{L}+(0,\varepsilon)^n}} \omega_M^{\mathbf{y}\cdot(\mathbf{x}_2-\mathbf{x}_1)}.$$

To bound this probability from below, we show that the phases corresponding to an element $\mathbf{y}$ close to a dual lattice vector are small. Each term $\mathbf{x}_2 - \mathbf{x}_1$ is of the form $Nv + \varepsilon_v$, where $v \in \mathcal{L}$ and $|\varepsilon_v| < 1$. The $\mathbf{y}$ that we hope to measure are of the form $\lfloor tqw \rceil$ for $w \in \mathcal{L}^*$. Moreover, to make sure that the phase terms remain bounded, we restrict ourselves to vectors with entries satisfying $|y_i| \leq \frac{qNt}{\log|\Delta|}$. This means that we are measuring approximations of $w \in \mathcal{L}^*$ with $|w_i| \leq \frac{qNt}{\log|\Delta|} + 1$ and that N has to be chosen large enough so that we measure a significant portion of $\mathcal{L}^*$. So $\mathbf{y} = qtw + \delta_w$ for $\|\delta_w\| < \frac{1}{2}$, and

$$\mathbf{y} \cdot (\mathbf{x}_2 - \mathbf{x}_1) = (qtw + \delta_w) \cdot (Nv + \varepsilon_v) = qNt(w \cdot v) + qt(w \cdot \varepsilon_v) + \delta_w \cdot (Nv + \varepsilon_v).$$

The first term of the sum vanishes from the phase because it equals zero modulo qtN. Indeed, $v \cdot w \in \mathbb{Z}$. The second term satisfies

$$\left|\frac{qt(w \cdot \varepsilon_v)}{qtN}\right| \leq \frac{n \max_i |w_i|}{N} \leq \frac{n}{\log|\Delta|} \approx \frac{1}{\log(n)}.$$

Finally, the third term of the phase satisfies

$$\left|\frac{\delta_w \cdot (Nv + \varepsilon_v)}{qtN}\right| \leq \frac{|\delta_w \cdot v|}{qt} + \frac{|\delta_w \cdot \varepsilon_v|}{qtN} \leq \frac{n \max|v_i|}{qt} \leq \frac{1}{8}$$

if we choose $t \geq 8n$. So, for large enough n, we have $|\frac{\mathbf{y} \cdot (\mathbf{x}_2 - \mathbf{x}_1)}{qtN}| < \frac{1}{6}$, and the probability $P_{\mathbf{z}}$ of measuring $\mathbf{z}$ satisfies

$$\begin{aligned} P_{\mathbf{z}} &= \frac{1}{M^{2n}} \sum_{\mathbf{x}_1 \in \mathbb{Z}_M^n} \sum_{\substack{\mathbf{x}_2 \in \mathbb{Z}_M^n \\ \mathbf{x}_1 - \mathbf{x}_2 \in \mathcal{L} + (0,\varepsilon)^n}} \omega_M^{\mathbf{y} \cdot (\mathbf{x}_2 - \mathbf{x}_1)} = \frac{1}{M^n} \sum_{\mathbf{u} \in \mathcal{L} \cap [0,q]^n} \omega_M^{\mathbf{y} \cdot [N\mathbf{u}]} \\ &\geq \frac{1}{2M^n} \sum_{\mathbf{u} \in \mathcal{L} \cap [0,q]^n} (e^{2i\pi/3} + e^{-2i\pi/3}) = \frac{|\mathcal{L} \cap [0,q]^n|}{2M^n}. \end{aligned}$$

The above probability holds for all $\mathbf{z} \in \mathcal{L}^*$ with entries bounded by $\frac{N}{\log|\Delta|}$. As in [11], we need to relate the number of points in $\mathcal{L}_q = \mathcal{L} \cap [0,q]^n$ to the number of points of $\mathcal{L}^*_{N/\log|\Delta|} = \mathcal{L}^* \cap [0, \frac{N}{\log|\Delta|}]^n$. Let λ be a bound on the length of the vectors in a reduced basis of $\mathcal{L}$; by [15, Proposition 8.7], we have $|\mathcal{L}_q| \geq \frac{q^n}{2\det(\mathcal{L})}$ if $q \geq n^2\lambda$ and $|\mathcal{L}^*_{N/\log|\Delta|}| \geq \frac{(N/\log|\Delta|)^n}{2\det(\mathcal{L}^*)}$ if $N \geq \log|\Delta|^n n^2 \lambda$. Therefore,

$$|\mathcal{L}_q||\mathcal{L}^*_{N/\log|\Delta|}| \geq \frac{q^n(\frac{N}{\log|\Delta|})^n}{4\det(\mathcal{L})\det(\mathcal{L}^*)} = \frac{q^n(\frac{N}{\log|\Delta|})^n}{4},$$

and the probability of drawing $\mathbf{z}$ such that $qt\mathbf{z}$ is $\frac{1}{q}$-close to $w \in \mathcal{L}^*_{N/\log|\Delta|}$ satisfies

$$P_{\mathbf{z}} \geq \frac{|\mathcal{L}_q|}{2M^n} \geq \frac{1}{8(\log|\Delta|t)^n} \frac{1}{|\mathcal{L}^*_{N/\log|\Delta|}|}.$$

As pointed out in [11], such $\mathbf{z}$ are the points of our grid such that $\frac{\mathbf{y}}{qt}$ is $\frac{1}{q}$-close to a $w \in \mathcal{L}^*_{N/\log|\Delta|}$. As there are $|\mathcal{L}^*_{N/\log|\Delta|}|$ vectors $\mathbf{y}$ associated to such a w, the probability of measuring one is at least $\frac{1}{8(\log|\Delta|t)^n}$. □

The above statement gives a lower bound on the probability of drawing points that are approximations of elements in $\Lambda^*_{\alpha'}$. This, in turn, gives an upper bound on the run time to obtain enough approximations of lattice points before being able to find a basis of $\Lambda_{\alpha'}$. Still assuming that the same techniques are used, we can also derive an upper bound on the probability of sampling an approximation of a dual lattice point, which, in turn, gives a lower bound on the run time of the algorithm.

Proposition B.2 (Exponential run time). *Under the same assumptions as Proposition B.1, the run time of the overall algorithm is at least* 2^n.

Proof. With the same choice of parameters as in the proof of the previous proposition, the probability of drawing $\mathbf{z}$ satisfies

$$P_{\mathbf{z}} = \frac{1}{M^n} \sum_{\mathbf{u} \in \mathcal{L} \cap [0,q]^n} \omega_M^{\mathbf{y} \cdot [N\mathbf{u}]} \leq \frac{|\mathcal{L} \cap [0,q]^n|}{M^n} \approx \frac{q^n}{M^n \det(\mathcal{L})}.$$

There are $|\mathcal{L}^*_{N/\log|\Delta|}| \approx \frac{(N/\log|\Delta|)^n}{\det(\mathcal{L}^*)}$ such points, which means that the probability of drawing a rational approxi-

mation that is $\frac{1}{q}$-close to a point in $\mathcal{L}^*_{N/\log|\Delta|}$ is no more than

$$P \leq \frac{\left(\frac{N}{\log|\Delta|}\right)^n}{\det(\mathcal{L}^*)} \frac{q^n}{M^n \det(\mathcal{L})} = \frac{1}{(\log|\Delta|t)^n} \leq \frac{1}{2^n}.$$

The total run time is at least as much as the time taken to draw a single approximation of a dual lattice point, which is at least 2^n. □

Remark. The above analysis shows that if we only assume that the quantum fingerprint has the property (called "fidelity") that

- $\langle \psi_{\mathbf{x}_1} | \psi_{\mathbf{x}_2} \rangle = 1$ if $\mathbf{x}_2 - \mathbf{x}_1$ is ε-close to $\mathcal{L}$ for some $\varepsilon < \frac{1}{N}$,
- $\langle \psi_{\mathbf{x}_1} | \psi_{\mathbf{x}_2} \rangle = 0$ otherwise,

then the techniques mentioned by CGS relying on the discretization of $\mathbb{R}^m$ and the QFT do not allow to prove that the procedure has a polynomial run time.

Funding: This work was supported by the U.S. National Science Foundation under grants 1839805 and 1846166, by NIST under grant 60NANB17D184, and by a Seed Award of the Florida Center for Cybersecurity.

References

[1] J.-F. Biasse, Subexponential time relations in the class group of large degree number fields, *Adv. Math. Commun.* **8** (2014), no. 4, 407–425.
[2] J.-F. Biasse and C. Fieker, Subexponential class group and unit group computation in large degree number fields, *LMS J. Comput. Math.* **17** (2014), 385–403.
[3] J.-F. Biasse and 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*, ACM, New York (2016), 893–902.
[4] P. Campbell, M. Groves and D. Shepherd, SOLILOQUY, a cautionary tale.
[5] H. Cohen, *Advanced Topics in Computational Number Theory*, Grad. Texts in Math. 193, Springer, New York, 2000.
[6] R. Cramer, L. Ducas, C. Peikert and O. Regev, Recovering short generators of principal ideals in cyclotomic rings, IACR Cryptology ePrint Archive (2015), https://eprint.iacr.org/2015/313.
[7] 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 2014 ACM Symposium on Theory of Computing—STOC'14*, ACM, New York (2014), 293–302.
[8] S. Garg, C. Gentry and S. Halevi, Candidate multilinear maps from ideal lattices, in: *Advances in Cryptology—EUROCRYPT 2013*, Lecture Notes in Comput. Sci. 7881, Springer, Heidelberg (2013), 1–17.
[9] C. Gentry and M. Szydlo, Cryptanalysis of the revised NTRU signature scheme, in: *Advances in Cryptology—EUROCRYPT 2002*, Lecture Notes in Comput. Sci. 2332, Springer, Berlin (2002), 299–320.
[10] L. Hales, *The quantum fourier transform and extensions of the abelian hidden subgroup problem*, PhD thesis, University of California Berkeley, 2002.
[11] S. Hallgren, Fast quantum algorithms for computing the unit group and class group of a number field, in: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing—STOC'05*, ACM, New York (2005), 468–474.
[12] S. Hallgren, Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem, *J. ACM* **54** (2007), no. 1, Article ID 4.
[13] N. Howgrave-Graham and M. Szydlo, A method to solve cyclotomic norm equations $f * \overline{f}$, in: *Algorithmic Number Theory*, Lecture Notes in Comput. Sci. 3076, Springer, Berlin (2004), 272–279.
[14] M.-H. Kim and S.-G. Lim, Square classes of totally positive units, *J. Number Theory* **125** (2007), no. 1, 1–6.
[15] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems. A Cryptographic Perspective*, Kluwer Int. Ser. Eng. Comp. Sci. 671, Kluwer Academic, Boston, 2002.
[16] O. Regev, Private communication, 2015.
[17] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26** (1997), no. 5, 1484–1509.
[18] N. P. Smart and F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes, in: *Public Key Cryptography—PKC 2010*, Lecture Notes in Comput. Sci. 6056, Springer, Berlin (2010), 420–443.
[19] L. C. Washington, *Introduction to Cyclotomic Fields*, Grad. Texts in Math. 83, Springer, New York, 1982.
[20] H. Weber, *Lehrbuch der Algebra. Vol. II*, Vieweg, Braunschweig, 1899.