Research Article

Nour-eddine Rahmani*, Taoufik Serraj, Moulay Chrif Ismaili, and Abdelmalek Azizi

On the quantum security of high-dimensional RSA protocol

https://doi.org/10.1515/jmc-2024-0007
received February 10, 2024; accepted September 13, 2024

Abstract: The idea of extending the classical RSA protocol using algebraic number fields was introduced by Takagi and Naito (Construction of RSA cryptosystem over the algebraic field using ideal theory and investigation of its security. Electron Commun Japan Part III Fund Electr Sci. 2000;83:19–29). Recently, Zheng et al. proposed the use of the ring of algebraic integers of an algebraic number field and the lattice theory to present a high-dimensional form of RSA. The authors claim that their proposal is post-quantum and is significant both from the theoretical and practical point of view. In this article, we prove that the security of Zheng et al.'s scheme is still based on the factorization problem, and we present a practical quantum attack on this proposed scheme, our attack is a quantum polynomial time algorithm that employs Shor's algorithm as a subroutine.

Keywords: RSA, lattices, algebraic number theory, cryptanalysis

MSC 2020: 11Y40, 11T71, 94A60

1 Introduction

Over the past three decades, information and communication technologies have changed our everyday life in different areas, and many services are provided online. In order to secure sensitive data exchanged or stored over public networks, many symmetric and asymmetric cryptographic techniques are used. Nowadays, Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA) encryption and Elliptic Curve Cryptography (ECC) are examples of schemes widely used for this purpose. Due to the rapid development of quantum technology in recent years and according to Grover's algorithm [1], the impact of quantum algorithms on symmetric cryptographic primitives is not expected to be as severe as Shor's algorithm [2] on number-theoretic-based public key constructions such as RSA and its underlying integer factorization problem. As a result, the current emphasis in post-quantum cryptography is on public-key cryptography. However, especially during the future standardization process, it is critical to consider the diversity of cryptographic primitives and the underlying hard mathematical problems. The approaches were studying new alternatives to public-key cryptosystems based on the integer factorization and discrete logarithm problems or extending the existing schemes to become post-quantum.

In 2005, Regev introduced the learning with error (LWE) problem [3] and showed that we could construct a public key scheme where its security is based on the LWE problem, but the scheme was not efficient for practical use. In 2010, Lyubashevsky et al. introduced the Ring-LWE [4], a variant of the LWE [3], and showed an efficiently practical scheme construction using elements of the ring of integers of an algebraic number field.

*** Corresponding author: Nour-eddine Rahmani,** ACSA Laboratory, Faculty of Sciences, Mohammed Premier University, Oujda, Morocco, e-mail: nour-eddine.rahmani@ump.ac.ma
Taoufik Serraj: ACSA Laboratory, Faculty of Sciences, Mohammed Premier University, Oujda, Morocco, e-mail: t.serraj@ump.ac.ma
Moulay Chrif Ismaili: ACSA Laboratory, Faculty of Sciences, Mohammed Premier University, Oujda, Morocco, e-mail: mc.ismaili@ump.ac.ma
Abdelmalek Azizi: ACSA Laboratory, Faculty of Sciences, Mohammed Premier University, Oujda, Morocco, e-mail: a.azizi@ump.ac.ma

The search version of the LWE (resp. Ring-LWE) is given (A, b) such that A is uniformly sampled from $\mathbb{Z}^{m\times n}$ (resp. A is uniformly sampled from R, where R is the ring of integers of a number field K) and $b = As + e \bmod q$, where $s \in \mathbb{Z}^n$ and $e \in \mathbb{Z}^m$ (resp. $s \in R$ and $e \in R$) following some specific distributions D_s, D_e, q an integer modulus, and the goal is to find s or e. We refer the reader to previous studies [4–8] for more details about algebraically structured variants in the literature.

Nowadays, lattice-based constructions are considered to be promising alternatives; indeed, there are two digital signatures (Falcon [9] and CRYSTALS-Dilithium [10]) and one key-encapsulation mechanism (CRYSTALS-Kyber [11][1]) based on lattice hard problems, which are selected to be standardized by National Institute of Systems and Technologies (NIST) in 2022 [12]. Analysing lattice constructions is still an open topic; many algorithms to solve lattice problems have been improved in the last 20 years, without any success to solve the problem in lattices of high dimensions [13–15], but lead to estimate more accurately the hardness of such a lattice instance. (For NTRU cryptanalysis, we refer the reader to [16,17].)

In 1986, introduced the idea to extend RSA to higher dimensions was introduced in the literature; later in 2015, Takagi and Naito [18] demonstrated a variant of the RSA in algebraic number fields; however, this necessitates that the ring of algebraic integers be a Euclidean ring, a requirement that is significantly more stringent than the class number one condition.

Zhiyong et al. [19] proposed a new cryptosystem in number fields similar to the RSA cryptosystem, claiming that the system's security is dependent on the problem of solving the factorization of ideals in the number field in question, which is, based on their claim, much more difficult than factorization of integers in the ring of integers $\mathbb{Z}$. As a result, they claim their construction as a new member of post-quantum construction. In this article, we prove that factorization in the ring of integer $\mathbb{Z}$ is as hard as factorization in the ring of integers O_K of any number field K is hard. This fact enables us to propose a quantum attack on high-dimensional RSA system.

The rest of the article is organized as follows: Section 2 recalls some notions on algebraic number theory and Euclidean lattices, and Section 3 briefly reviews the high-dimensional RSA encryption scheme. The proposed attack and the corresponding security and efficiency analysis are presented and discussed in Section 4. Finally, a conclusion is provided. An implementation using the PARI/GP system is given in the Appendix.

2 Preliminaries

This section recalls some notions and known results related to algebraic number fields and lattices. We denote by I_n the identity square matrix of n rows and n columns, by $0_{n,m}$ the zero matrix of n rows and m columns, we omit m when $m = n$, by $||\cdot||$ the euclidean norm of a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, $||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$, and by $\cdot^t$ the transpose for matrices and vectors.

2.1 Algebraic number theory

A field K that contains $\mathbb{Q}$ is called an **extension** of $\mathbb{Q}$. The dimension of K as $\mathbb{Q}$-vector space is called **the degree of the extension** and denoted by $[K : \mathbb{Q}]$. If the degree is finite, then we call K an **Algebraic Number Field**.

The set $\{y \in K | \exists P \text{ monic} \in \mathbb{Z}[X] : P(y) = 0\}$ is called the **ring of integers** of K and it is a ring under the induced operations of K and it is denoted by O_K. K is called **Galois number field** if for every irreducible $P \in \mathbb{Q}$, if P has a root in K, then all the other roots[2] are in K. (P splits into simple polynomials[3] in K.) Let us denote by $\mathrm{Gal}(K/\mathbb{Q})$ the Galois group of K which is the set of K-automorphisms that fix $\mathbb{Q}$. It is known that the number of

[1] Kyber now is called ML-KEM, abbreviation of Module-Lattice-Based Key-Encapsulation Mechanism, Dilithium is currently known as Module-Lattice-Based Digital Signature Standard, link for FIPS https://csrc.nist.gov/publications/fips.

[2] In fact, this is the definition of a normal extension, a Galois extension is a separable and normal extension. For separability, this is the case because $\mathbb{Q}$ is of characteristic 0, so each of its extensions, it is a separable extension. For more details, see Lang [20].

[3] Polynomials of degree 1.

automorphisms is equal to the degree of number field K, $\mathsf{Gal}(K/\mathbb{Q}) = \{\sigma : K \longrightarrow K | \sigma(x) = x, \forall x \in \mathbb{Q}\}$. Norm N and trace Tr of an element x of K are defined as follows: $\mathsf{N}(x) = \prod_{\sigma \in \mathsf{Gal}(K/\mathbb{Q})} \sigma(x)$ and $\mathsf{Tr}(x) = \sum_{\sigma \in \mathsf{Gal}(K/\mathbb{Q})} \sigma(x)$. Finally the discriminant of K is defined as $\Delta_K = \det((\mathsf{Tr}(b_i b_j))_{1 \le i,j \le n})$ for a basis $B = \{b_1, \ldots, b_n\}$.[4]

Example 1. The number field $K = \mathbb{Q}(\zeta_n)$ such that $\zeta_n = e^{\frac{2i\pi}{n}}$ is a primitive nth-root of unity, i.e., a root of $\Phi_n(x) = \prod_{\substack{1 \le k < n \\ gcd(k,n)=1}} (x - \zeta_i^k)$ is called the n**-th cyclotomic number field**. It is known that $K/\mathbb{Q}$ is a Galois number field of degree $\varphi(n)$ and φ is the Euler totient function.

We have the following facts from algebraic number theory:

Let K denote a number field of degree n, then the following properties hold:

1. There exists an α in K such that $K = \mathbb{Q}(\alpha)$, such one is called a primitive element of K, and K is isomorphic to $\mathbb{Q}[X]/\langle \phi_\alpha(X) \rangle$ where $\phi_\alpha(X) \in \mathbb{Q}[X]$ is the minimal polynomial of α.
2. Each ideal $p\mathbb{Z}$ for a prime number p of $\mathbb{Z}$ has a unique decomposition into product of prime ideals $\mathcal{P}_i$ in O_K:
$$(p) \coloneqq pO_K = \prod_{i=1}^{g} \mathcal{P}_i^{e_i},$$
e_i is called **index of ramification of** $\mathcal{P}_i$ **over** p. g is the **number of prime ideals** of K over p.
3. The index $[O_K/\mathcal{P}_i : \mathbb{Z}/p\mathbb{Z}]$ is finite and it is denoted by f_i and called **inertia degree of** $\mathcal{P}_i$ **over** p, and we have $\mathsf{N}(\mathcal{P}_i) = p^{f_i}$.
4. We have $n = \sum_{i=1}^{g} e_i f_i$, and if K is Galois, then there exist f and e from $\mathbb{N}^*$ such that $f_i = f$ and $e_i = e$ for every i, which implies $n = efg$. We simply call e and f by index of ramification of p and inertia degree of p, respectively.
5. It is known that for a prime ideal $\mathcal{P}$ of O_K above a prime number p in $\mathbb{Z}$, its norm is $\mathsf{N}_{K/\mathbb{Q}}(\mathcal{P}) = p^{f_\mathcal{P}}$ such that $f_\mathcal{P}$ is the residual degree of $\mathcal{P}$ above p. Also, for our work, we will use the following important property: For an ideal A in O_K such that there exist a set of prime ideals in O_K, $\mathcal{P}_1, \ldots, \mathcal{P}_k$ and $A = \prod_{i=1}^{k} \mathcal{P}_i$, we have
$$\mathsf{N}(A) = \prod_{i=1}^{k} \mathsf{N}(\mathcal{P}_i).$$
In particular, if $A = \mathcal{P}_1 \mathcal{P}_2$ for two prime ideals $\mathcal{P}_1$ and $\mathcal{P}_2$ in O_K above two integer primes p_1 and p_2 respectively, we have
$$\mathsf{N}(A) = p_1^{f_1} p_2^{f_2}$$
such that f_1 and f_2 are the inertia degree of $\mathcal{P}_1$ and $\mathcal{P}_2$ above p_1 and p_2, respectively.
6. The number of invertible elements of $(O_K/\mathcal{P}_i)/(\mathbb{Z}/p\mathbb{Z})$ is $\mathsf{N}(\mathcal{P}_i) - 1$, and in general for an ideal $A = \prod_i \mathcal{P}_i^{e_i}$ of O_K, we have $\mathsf{N}(A) = \prod_i \mathsf{N}(\mathcal{P}_i^{e_i})$ and the number of invertible elements of O_K/A is $\prod_i \mathsf{N}(\mathcal{P}_i)^{e_i - 1}(\mathsf{N}(\mathcal{P}_i) - 1)$. In particular for A as in point 5, the number of invertible elements of O_K/A is $(\mathsf{N}(\mathcal{P}_1) - 1)(\mathsf{N}(\mathcal{P}_2) - 1)$.

For proofs of the last propositions, we refer the reader to the study of Washington [21].

The following theorem tells us how to compute such a prime decomposition in K.

Theorem 1. [22] *Let K be a number field such that $K = \mathbb{Q}(\alpha)$ for $\alpha \in O_K$ defined by an irreducible polynomial $P(X) \in \mathbb{Q}[X]$, and let q be a prime number in $\mathbb{Z}$ such that*[5] *$q \nmid [O_K : \mathbb{Z}[\alpha]]$. Suppose that*
$$P(X) = \prod_{i=1}^{g} g_i(X)^{e_i} \bmod q,$$

[4] Remark that we omit to specify the basis, because the discriminant does not depend on the choice of the basis.

[5] If $O_K = \mathbb{Z}[\alpha]$, then the theorem holds for any prime number in $\mathbb{Z}$, see Murty and Esmonde [22] pp. 65–66.

then q splits in O_K as follows:

$$qO_K = \prod_{i=1}^{g} (g_i(\alpha), q)^{e_i},$$

and $f_i = \deg(g_i), \forall i$.

To decompose a prime integer in a number field K, it suffices to factor the minimal polynomial that defines K modulo p and this can be done in polynomial time (see Section 3.4, p. 124, in Cohen [23]).

2.2 Euclidean lattices

In this section, we provide the necessary preliminaries for a better understanding of the subsequent discussions.

Definition 1. An Euclidean lattice (or simply a lattice) $\mathcal{L}$ formally is a discrete subgroup of $\mathbb{R}^n$ for a norm $||\cdot||$, and equivalently it is a free $\mathbb{Z}$-module of free rank m contained in $\mathbb{R}^n$. A lattice can be represented by a basis $B = \{b_1, \ldots, b_m\}$, for $b_i \in \mathbb{R}^n$.

A basis is not unique if $1 < m \leq n$, so

$$\mathcal{L}(B) := \left\{ \sum_{i=0}^{m} \alpha_i b_i | \alpha_i \in \mathbb{Z} \right\},$$

and C is a basis of $\mathcal{L}$ if and only if there is a unimodular matrix U (i.e. $|\det(U)| = 1$ and $U \in \mathbb{Z}^{m\times m}$) such that $B = CU$, consequently for a lattice of rank greater than 2, there are infinite many different bases of that lattice. If $m = n$, then the lattice is said a full-rank lattice. From lattices theory, there is a shortest non-zero vector in $\mathcal{L}$, and finding such a vector is the well-known problem of the shortest vector problem (SVP).

Definition 2. (SVP)
- **Given**: $B \in \mathbb{Z}^{n\times m}$,
- **Find**: $z \in \mathbb{Z}^m \backslash \{0\}$, such that $||Bz|| \leq ||By||$ for every y in $\mathbb{Z}^m$.

Another important lattice problem is the closest vector problem (CVP), given a non-lattice target vector, find the closest lattice vector to the latter.

Definition 3. (CVP)
- **Given**: $B \in \mathbb{Z}^{n\times m}$, and $t \in \mathbb{R}^n \backslash \mathcal{L}$
- **Find**: $z \in \mathbb{Z}^m \backslash \{0\}$, such that $||Bz - t|| \leq ||By - t||$ for every y in $\mathbb{Z}^m$.

If $\mathcal{L}$ is a subset of $\mathbb{Z}^n$, then it is called an integral lattice, and for a given q if $\mathcal{L}$ contains $q\mathbb{Z}^n$, then $\mathcal{L}$ is called a q-ary lattice. For more details and discussion of the hardness of the above problems, we refer the reader to [7,24].

Since there are infinitely many bases for a given lattice, it is natural to ask which of basis is better to work with. This is a well-studied topic in lattices theory, and there is no precise definition for a good basis and bad basis. A good basis in general is a basis with short vectors; in contrast, a bad basis is constituted by long vectors and form a skewed parallelepiped. The operation to find a good basis from a given bad basis is known as lattice reduction [13,25], and there are many strategies to find such one, and they differ by running time and the quality of the output basis, precisely the LLL algorithm runs in polynomial time but it produces a base with exponential approximate short vector, in contrast HKZ-reduction runs in exponential time and produces a basis containing a shortest vector. We refer the reader to the survey [7] for more details. Finally, good to mention that improving lattice reductions is still an open research [14] and most known efficient implementation is known as the general sieve kernel [15].

3 High-dimensional RSA

In this section, we state the ideal factorization problem in number fields, and we recall the high-dimensional RSA as it is described in the original article. Then, we provide some remarks on the vulnerabilities that we found.

3.1 Coefficient embedding and rotation matrix

In the study of Zhiyong et al. [19], the multiplication of elements in K is defined as the matrix-vector product which is known in the literature of lattice-based cryptography (e.g. see [7,8]), by using the rotation matrix of an element a in K and multiply it by the vector corresponding to the coefficients of an element b of K. We respect the same notation of the paper for clarity.

Every degree n number field $K = \mathbb{Q}(\zeta)$ defines an n-dimensional vector space over $\mathbb{Q}$ with basis $1, \zeta, \ldots, \zeta^{n-1}$. As a result, any element $a \in K$ may be expressed as $a = \sum_{j=0}^{n-1} a_j \zeta^j$, where $a_j \in \mathbb{Q}$.

The isomorphism that sends every element a in K to its coefficient vector $\tau(a) = (a_0, \ldots, a_{n-1})^t$ is the coefficient embedding $\tau : K \to \mathbb{Q}^n$ and denoted by $\tau(x) = \bar{x}$. By the coefficient embedding, multiplication by x can be represented by a matrix multiplication, with the associated matrix denoted by $\mathrm{Rot}(x) \in \mathbb{R}^{n\times n}$. More specifically, it returns $\tau(a) = \mathrm{Rot}(b) \cdot \tau(c)$ for every a, b, c in K with $a = bc$. It is worth noting that the matrix $\mathrm{Rot}(a)$ is invertible in K for all $a \neq 0$, and that its concrete form is determined by the number field K.

Definition 4. Let K be a number field. We define $\otimes$ to be the operation between coefficients vectors of α and β, for α and β from K such that the result is in the number field K. Explicitly

$$\forall \alpha, \beta \in K, \bar{\alpha} \otimes \bar{\beta} \coloneqq \tau^{-1}(\mathrm{Rot}(\alpha) \cdot \tau(\beta)).$$

We remark that $\bar{\alpha} \otimes \bar{\beta}$ is an element of K. Clearly, if $a = \sum_{i=0}^{n-1} a_i \zeta^i \in O_K$ and $O_K = \mathbb{Z}[\zeta]$, then $\tau(a) = (a_0, a_1, \ldots, a_{n-1})^t \in \mathbb{Z}^n$. The property $O_K = \mathbb{Z}[\zeta]$ is called NC-property in the paper of [19]. They defined and denoted the rotation matrix by the following matrix:

$$\mathrm{Rot}(a) = H^*(a) = [\tau(a), H\tau(a), \ldots, H^{n-1}\tau(a)],$$

where H depends on the number field defined by the polynomial $\phi(x) = x^n - \sum_{i=0}^{n-1} \phi_i x^i$, and it equals

$$H = \begin{bmatrix} 0 & \ldots & 0 & \phi_0 \\ & & & \phi_1 \\ & I_{n-1} & & \vdots \\ & & & \phi_{n-1} \end{bmatrix}.$$

The product of two elements a and b in K can be computed by $a \cdot b = \tau^{-1}(\mathrm{Rot}(a)\tau(b))$.

3.2 Description of the high-dimensional RSA

Let $n \geq 1$ be a positive integer, K be an algebraic number field with the NC-property of degree n, $R = O_K \subset K$ be the ring of algebraic integers of K, $\alpha \in R$, $\beta \in R$ be two distinct prime elements of R, $A = \alpha\beta R$ be a principal ideal of R, $H^*(\bar{\alpha} \otimes \bar{\beta})$ be the ideal matrix corresponding to A, $L_{\alpha,\beta} = L_{H^*}(\bar{\alpha} \otimes \bar{\beta})$ be the lattice generated by

$H^*(\bar{\alpha} \otimes \bar{\beta})$, $B_{\alpha,\beta} = \mathrm{HNF}(L_{\alpha,\beta})$ be the basis of $L_{\alpha,\beta}$ in Hermite normal form, and $B^*_{\alpha,\beta} = \mathrm{diag}\{b_1, b_2, \ldots, b_n\}$ be the elements in the diagonal of the $B_{\alpha,\beta}$ matrix.

Parameters:

$$\phi(\alpha,\beta) = (|\det(H^*(\alpha))| - 1) \cdot (|\det(H^*(\beta))| - 1),$$
$$S_{\alpha,\beta} = \{x = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n \mid 0 \leq x_i < b_i\},$$
$$1 \leq e < \phi(\alpha,\beta) \text{ such that } e \text{ coprime with } \phi(\alpha,\beta),$$
$$1 \leq d < \phi(\alpha,\beta) \text{ such that } ed \equiv 1 \pmod{\phi(\alpha,\beta)}.$$

Public keys: The rotation matrix H, the lattice $L(B_{\alpha,\beta}) = L_{\alpha,\beta}$, and the positive integer e are public keys.

Private keys: Ideal matrices $H(\bar{\alpha})$ and $H(\bar{\beta})$, the basis $H^*(\bar{\alpha} \otimes \bar{\beta})$ of $L_{\alpha,\beta}$, and the positive integer d are private keys.

Encryption: For any input message $a \in S_{\alpha,\beta}$, the ciphertext c is given by $c \equiv a^e \pmod{L_{\alpha,\beta}}$.

Decryption: $c^d \equiv a^{de} \equiv a^{k\phi(\alpha,\beta)+1} \equiv a \pmod{L_{\alpha,\beta}}$. One can find the plaintext a from c in $S_{\alpha,\beta}$.

Decryption success probability: The authors proved that decryption success probability depends on the norm of $A = \mathcal{P}\mathcal{Q}$ for prime ideals $\mathcal{P}, \mathcal{Q}$ in O_K and the splitting behaviour of the integer primes p, q in $\mathcal{P}, \mathcal{Q}$ respectively, showing that the decryption success probability is

$$s = \frac{p^{f_{\mathcal{P}}} q^{f_{\mathcal{Q}}}}{p^n q^n},$$

where $f_{\mathcal{P}}, f_{\mathcal{Q}}$ the residual degree of $\mathcal{P}, \mathcal{Q}$ above p, q, respectively, and n is the degree of K over $\mathbb{Q}$.

4 Attacking high-dimensional RSA

While claiming that the proposed scheme has post-quantum security, we do not find any proof for this claim in the original paper; in this section, we provide remarks on the security of the scheme that lie on the hardness of factorization in number fields.

4.1 Hardness of factorization in number fields

We prove the following fact:

Theorem 2. *Let K be a number field with degree d. There is a polynomial time algorithm that factors elements in the ring of integers $\mathbb{Z}$ if and only if there is a polynomial time algorithm that factors ideals in the ring of integers O_K of the number field K.*

Proof. It is clear, if there is a polynomial time algorithm to factor an ideal A of O_K into a product of prime ideals in O_K in polynomial time leads to factor an integer $n = \prod_{i=1}^{r} p_i^{e_i}$ in $\mathbb{Z}$ in polynomial time. Considering the ideal in O_K generated by n and since the ring of integers is a Dedikind domain A factors uniquely into prime ideals (up to permutation) as follows:

$$A = nO_K = \prod_{i=1}^{r} \prod_{j=1}^{g_i} \mathcal{P}_{i,j}^{e_{i,j}}$$

in O_K where each of $\mathcal{P}_{i,j}$ is a prime ideal of O_K above p_i, $e_{i,j}$ is the ramification index of $\mathcal{P}_{i,j}$'s above p_i, and g_i is the number of prime ideals in O_K above p_i for all i, j. Then, computing $\mathcal{P}_{i,j} \cap \mathbb{Z} = p_i\mathbb{Z}$ and $e_i = \max\{e : n \equiv 0 \bmod p_i^e\}$ and return (p_i, e_i) for every i.

Now, we prove the other direction. For a fixed number field $K/\mathbb{Q}$, given an ideal $A = \prod_{i=1}^{r} \mathcal{P}_i^{t_i}$ of O_K with $\mathcal{P}_i$ are prime ideals of O_K, and our goal is to find $\mathcal{P}_i$ and t_i for each i. For simplicity, we assume that the prime ideals are ordered by the prime integers they contain, for each $1 \le i < j \le r$, there is p_i and p_j prime integers in $\mathbb{N}$ such that: $p_i \le p_j$, $p_i \in \mathcal{P}_i$, and $p_j \in \mathcal{P}_j$. Assume that there is an algorithm that solves the problem of integer factorization in polynomial time. First, we compute the algebraic norm $N = \mathrm{N}(A)$, which is an integer number in $\mathbb{Z}$, the algebraic norm of an ideal is known to be computable in polynomial time by computing the determinant of its representative matrix (since determinants can be computed in polynomial time using, for example, Gaussian elimination). Using the integer factorization algorithm to factor N over $\mathbb{Z}$, the algorithm will output $(p_i, t_i f_i)$ for $1 \le i \le r$ such that p_i is in increasing order, and we have

$$N = \prod_{i=1}^{r} p_i^{t_i f_i},$$

where, for each $1 \le i \le r$, f_i is the residual degree of $\mathcal{P}_i$ above p_i, which is less than d. Now, for each i, the procedure to find $\mathcal{P}_i$ and t_i is starts by decomposing the prime integers p_i in O_K, which returns a set of prime ideals $\mathcal{P}_{i,j}$ of O_K that lie above p_i, since the prime decomposition is unique in a O_K, only one of the $\mathcal{P}_{i,j}$ is equal to $\mathcal{P}_i$, which divides A and $\mathcal{P}_i$ lies above p_i in O_K, thus, running – for example – an exhaustive search for the right index j and the maximum exponent t_i of the prime ideal $\mathcal{P}_{i,j}$ above p_i such that $\mathcal{P}_{i,j}^{t_i}$ divides A for each i and j. By the proposition from the fact 2.1, we have less than d many prime ideals above p_i. Thus, this process clearly is polynomially bounded in the number field degree d and number of primes that divide N. □

The previous proof has no efficiency concerns, in fact it is only to show that the two problems are equivalent computationally for any number field K. As a consequence, we conclude that if there is a polynomial time algorithm that solves one of them in polynomial time, then necessarily there is an algorithm that solves the other equivalent problem in polynomial time, which implies that the problem is solvable (e.g. using Shor's algorithm) in quantum polynomial time. One may wonder if there is a number field K in which it is possible to perform ideal factorization in polynomial time (may be classical), thus by the theorem we know that the existence of such a field implies that we can factor integers n of $\mathbb{Z}$ in polynomial time by factoring nO_K using the known algorithm and computing $\mathcal{P}_i \cap \mathbb{Z} = p\mathbb{Z}$ which can also be done in polynomial time as described in the proof of Theorem 2.

4.2 Parameter restriction

Assume that p, q are b-bits prime numbers that lie in the prime ideals $\mathcal{P}$, Q with inertia degree $f_{\mathcal{P}}, f_Q$ respectively, then we have the following inequalities:

$$2^{b(f_{\mathcal{P}}+f_Q-2n)} \le s \le 2^{(b-1)(f_{\mathcal{P}}+f_Q-2n)}.$$

Since $f_{\mathcal{P}}$ and f_Q are less than or equal to n, the decryption success probability s is negligible in b and n unless $f_{\mathcal{P}} = f_Q = n$. This is another inconvenience of the proposed scheme, on the one hand, factorization of integers is achievable for product of small primes, on the other hand, for larger bit-size prime numbers, if one of the selected primes does not have inertia degree equals to n in the number field K, then we have a negligible success probability which makes the scheme useless for. This restriction also makes key generation harder. The decryption success probability in question is related to the construction, and the proposed scheme does not provide any additional procedures to make the scheme randomized for public key purpose.

In the next section, we show our last observation using the fact from Theorem 2 and the restriction above, which make the proposed scheme insecure.

4.3 Attack description

We describe in this section our key recovery attack against the described construction in Algorithm 1, which we will prove that it runs in (at most quantum) polynomial time.

We have seen in the previous section that to obtain a negligible decryption failure is to choose α, β such that $\mathcal{P} = \alpha R, \mathcal{Q} = \beta R$ are inert prime ideals of $\mathbb{Z}$ in O_K. So, we have to focus on prime ideals of form $\mathcal{P} = pR$ for p in $\mathbb{Z}$. The latter have norm p^n and for $A = \mathcal{P}\mathcal{Q} = pqR$ the norm of A is clearly $\mathrm{N}(A) = p^n q^n$, and hence, we have decryption failure probability equal to 0. It is known that we can compute the norm of A from the given $B_{\alpha,\beta}$ by computing the determinant of $B_{\alpha,\beta}$. Using the following algorithm, we could retrieve the private key d. We do not need to retrieve α and β since $d = e^{-1} \bmod \phi(A)$, where $\phi(A) = (p^{f_p} - 1)(q^{f_q} - 1)$ is computable if we know p and q. Thus, only p and q are sufficient to retrieve and then makes, the construction insecure.

Algorithm 1: Compute private key d

```
    Input: Public key as lattice basis $B_{\alpha,\beta}$
    Output: Private key $d$
1   Compute $N \leftarrow \det B_{\alpha,\beta}$;
2   Compute $N \leftarrow N^{\frac{1}{n}}$;
3   F($(p, e_p), (q, e_q)$)$\leftarrow$ Factor ($N$) its prime factors;
4   Compute $d \leftarrow e^{-1} \bmod (p^n - 1)(q^n - 1)$;
5   return $d$
```

4.4 Running time and algorithm correctness

Since $\mathcal{P} = \alpha R$ and $\mathcal{Q} = \beta R$, by algebraic number theory, we know that $\mathcal{P} \cap \mathbb{Z} = p\mathbb{Z}$ and $\mathcal{Q} \cap \mathbb{Z} = q\mathbb{Z}$ for prime numbers $p, q \in \mathbb{Z}$. Therefore $\det(L_{\alpha,\beta}) = \mathrm{N}(L_{\alpha,\beta}) = p^{f_P} q^{f_Q}$. The following remarks justify why the algorithm runs in polynomial time:

- The lattice volume is an invariant of the lattice, which leads to computing the norm from the bad basis $B_{\alpha,\beta}$.
- Step 2 can be done efficiently using sufficient precision, and the result is surely an integer since we restricted the choice of parameters as in the previous discussion.
- Step 3 can be done by using any factoring algorithm. Using Shor's algorithm makes this step computable in quantum polynomial time.
- Step 4 can be done in polynomial time even in a classical computer.

This proves that our suggested algorithm runs at most in quantum polynomial time; therefore, our recovery attack is efficient, and the proposed scheme is not a post-quantum construction.

Basing the proposed scheme on number fields of large degree may help to resist Shor's algorithm (e.g. extensions of degree ≥ 100), but since the proposed construction work with principal maximal ideal of O_K this is not always secure because for example if the primes contained in the ideals splits completely then the norm of the public key does not get increased sufficiently to make the Shor's factoring algorithm costly, thus, one should avoid primes that splits completely in K in the key generation process, and also for inert primes since we can compute n-th root of the determinant of the lattice which is the result of the norm of the ideals multiplication. Conversely, increasing the number field degree makes the computations too slow, which is not favourable in practice.

Remark 1. We stress that we do not see any role of the lattice structure in the proposed design security, nor the author of the proposed scheme has presented a security guaranty based on a lattice problem. Our attack does not exploit any problem related to lattices, and hence, the design has no security guarantee based on any of the lattice problems (e.g. the SVP).

Example 2. Let $K = \mathbb{Q}(\sqrt[3]{2})$ be the number field defined by the polynomial $x^3 - 2 = X^3 - 0 \cdot x^2 - 0 \cdot x - 2$. Its ring of integers verifies $O_K = \mathbb{Z}[\sqrt[3]{2}]$. The matrix H related to this field is

$$H = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Let $a = 3 + 2\sqrt[3]{2^2}$ and $b = 3 + \sqrt[3]{2} - \sqrt[3]{2^2}$, then their coefficients vectors are, respectively, $\bar{a} = (3, 0, 2)^t$ and $\bar{b} = (3, 1, -1)$. Now, to compute the product ab, we need to compute $\tau^{-1}(H^*(a)\bar{b})$. Computing $H^*(a)$, we find

$$H^*(a) = [\bar{a} \quad H\bar{a} \quad H^2\bar{a}] = \begin{bmatrix} 3 & 4 & 0 \\ 0 & 3 & 4 \\ 2 & 0 & 3 \end{bmatrix}$$

and therefore

$$H^*(a)\bar{b} = [\bar{a} \quad H\bar{a} \quad H^2\bar{a}] \cdot \bar{b} = \begin{bmatrix} 3 & 4 & 0 \\ 0 & 3 & 4 \\ 2 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 13 \\ -1 \\ 3 \end{bmatrix},$$

hence,

$$ab = 13 - \sqrt[3]{2} + 3\sqrt[3]{2^2}.$$

A simple verification

$$\begin{aligned} xy &= (3 + 2\sqrt[3]{2^2}) \cdot (3 + \sqrt[3]{2} - \sqrt[3]{2^2}) \\ &= 9 + 3\sqrt[3]{2} - 3\sqrt[3]{2^2} + 6\sqrt[3]{2^2} + 4 - 4\sqrt[3]{2} \\ &= 13 - \sqrt[3]{2} + 3\sqrt[3]{2^2} \end{aligned}$$

is needed.

We can verify that x and y are irreducible elements of O_K of prime norms 59 and 43, respectively.[6] The public key is the product $xy = 13 - \sqrt[3]{2} + 3\sqrt[3]{4}$, which corresponds to its rotation matrix in Example 2. The totient in this case is $\phi = (59^1 - 1)(43^1 - 1) = 2{,}436$.

Let the public key be $e = 5$, and $d = 5^{-1} = 1{,}949 \mod 2{,}436$. Encrypting the message $m = 1 + \sqrt[3]{2} + \sqrt[3]{4}$, we obtain $c = m^{\otimes e} \equiv -2 - 8\sqrt[3]{2} - 3\sqrt[3]{4} \mod A$. Decrypting c gives $D = c^{\otimes d} \equiv 191 + 97\sqrt[3]{2} + 209\sqrt[3]{4} \mod A$, which is clearly different from the starting message m, and for this example, the success probability is $s = \frac{1}{64{,}36{,}369}$. This example shows the weakness of parameters that leads to observable decryption failure.

Example 3. Now, we encrypt the same message using prime ideals of K, which contains an integer prime that inerts in K. Let $x = 49 + 14\sqrt[3]{2} - 42\sqrt[3]{4}$ and $y = 31 + 93\sqrt[3]{2} - 93\sqrt[3]{4}$, $e = 11$, we obtain $\phi = 1{,}01{,}88{,}180$ and $d = 46{,}30{,}991 \mod 1{,}01{,}88{,}180$. The encryption then is $c = m^{\otimes e} \equiv 36 + 191\sqrt[3]{2} + 67\sqrt[3]{4} \mod A$ and the decryption can also be verified to be correct. But the matter here is that given the Hermite normal form of this parameters, one can compute the norm as the determinant of the matrix

$$B = \begin{bmatrix} 217 & 0 & 0 \\ 0 & 217 & 0 \\ 0 & 0 & 217 \end{bmatrix},$$

which is 217^3 and we have 217 = 7 × 31.

In order to give a numerical example of the proposed attack, we used Pari/GP software. In this example, we work with cyclotomic fields, one of the suggested families of number fields by Zhiyong et al. [19] that satisfies the NC-property.

[6] In this example, 59, 43 are not inert primes in K just to show the decryption failing.

Example 4. We fix $n = 128$ for the 256th cyclotomic number field with the following parameters:

1. the defining polynomial of the number field is $X^{128} + 1$,
2. the private keys are
   - $\alpha = 2x^{126} - x^{125} + x^{124} - 5x^{123} - x^{122} - 4x^{121} + x^{120} + 6x^{119} - 9x^{118} + 2x^{115} - x^{114} + 4x^{112} - x^{110} - 2x^{109} -$
     $2x^{107} - x^{106} - x^{105} - 2x^{104} - x^{103} - x^{102} + 2x^{101} + x^{99} - 2x^{98} - x^{97} + x^{96} - x^{95} + 3x^{94} - 3x^{93} - 3x^{92} - x^{91} +$
     $5x^{89} - x^{88} + x^{86} - x^{85} + x^{84} + 3x^{81} - x^{80} - 2x^{78} - x^{77} + x^{76} - 3x^{75} + 5x^{74} + 2x^{73} + 21x^{72} - x^{71} - x^{70} +$
     $11x^{66} + x^{65} - x^{64} + 2x^{63} - 5x^{61} - 6x^{60} + 2x^{59} + 3x^{58} + 7x^{57} + x^{56} - x^{55} - 4x^{54} - 8x^{53} - 2x^{52} - x^{51} - 2x^{49} -$
     $2x^{48} - x^{46} + 2x^{44} + x^{43} - 4x^{42} - 3x^{40} + 2x^{39} + 3x^{38} - x^{37} + 13x^{36} + x^{35} + 2x^{34} + 3x^{33} - x^{32} - x^{31} +$
     $x^{30} + x^{29} + 2x^{28} + x^{27} + x^{26} + x^{25} - 2x^{24} - 33x^{23} - 2x^{21} + x^{20} + 2x^{19} - x^{16} - 2x^{15} + x^{14} - x^{12} - x^{10} + 2x^{9} +$
     $3x^{8} - 17x^{5} - 6x^{4} + x^{3} + x^{2} + 8x - 1$
   - $\beta = 3x^{127} - x^{126} + x^{125} + x^{124} - x^{123} + 5x^{122} + x^{121} + x^{120} - x^{119} - x^{118} + x^{117} + 5x^{116} - x^{115} + x^{114} - x^{113} +$
     $2x^{111} - 5x^{110} - 99x^{109} + x^{108} - 2x^{105} + x^{103} + 2x^{102} + x^{101} + 11x^{99} - 3x^{98} + x^{97} - x^{96} + x^{93} - 2x^{92} + 4x^{91} -$
     $29x^{90} - 2x^{88} + x^{87} - 36x^{86} + x^{85} - 13x^{84} - x^{82} - 12x^{81} - x^{79} - x^{78} + x^{77} - 7x^{76} - x^{75} + 47x^{74} - 5x^{73} + 52x^{72}$
     $+ x^{71} - x^{69} - 3x^{67} - x^{66} - 13x^{65} - x^{63} + x^{62} - x^{61} + x^{60} - x^{59} + 8x^{58} - x^{57} - 3x^{56} - x^{54} + 2x^{53} - 3x^{52} -$
     $42x^{51} - 3x^{50} + 3x^{49} - x^{48} + x^{46} + 2x^{45} + x^{43} + x^{42} - x^{39} + x^{38} - 7x^{37} - 44x^{36} - x^{35} - 3x^{34} + x^{33} - 3x^{31} -$
     $x^{30} - 19x^{28} - 5x^{27} - x^{26} + 2x^{25} + x^{23} + 2x^{22} + x^{21} + x^{20} - x^{18} - 6x^{17} - 5x^{16} + 4x^{15} + 19x^{14} + 2x^{13} + x^{12} - x^{11}$
     $+ 2x^{9} - x^{8} - x^{7} + x^{5} + x^{4} - x^{3} + 4x^{2} + 2x.$
   - Computing $\phi(\alpha, \beta)$ we obtain (in hexadecimal):
     $\phi(\alpha, \beta) = 33bf97da2fddb268db31c6bb7a846d06d223f0faf9b3baee105a8152e1413a9f6d5da04bc33f$
     $794dfcc815ee61338c2f52ff51a05f34c583b84bb02a18b05d0505017dea0ea5936a75869717d49f15883$
     $366fa4fc4a12ba022882b0cda052e6d2de90c6108a117c55d57a88baf02a3a2d25a075fea12bcf3a5d31d$
     $3c53ea7428fd90dbad14340b40fae247133a296697c490cce35784648d5492f5ec18c06faf26ce3f647b5f$
     $c4f27961af8dbf9f5b825c3e10845f0508df70e5642d871d357ec00000.$
3. The public key which is the given ideal:
   $A = (-382x^{127} - 291x^{126} - 817x^{125} - 160x^{124} - 1203x^{123} - 1148x^{122} + 297x^{121} - 544x^{120} + 14x^{119} - 199x^{118} - 952x^{117} -$
   $201x^{116} + 309x^{115} + 1172x^{114} + 1634x^{113} - 105x^{112} + 235x^{111} - 231x^{110} + 717x^{109} - 419x^{108} + 789x^{107} - 128x^{106} + 429x^{105}$
   $+ 14x^{104} - 106x^{103} - 685x^{102} + 99x^{101} + 332x^{100} - 747x^{99} - 81x^{98} - 1142x^{97} + 631x^{96} - 1726x^{95} - 112x^{94} + 77x^{93} +$
   $46x^{92} + 545x^{91} + 67x^{90} + 356x^{89} + 577x^{88} - 852x^{87} + 524x^{86} - 493x^{85} - 240x^{84} - 236x^{83} + 371x^{82} + 229x^{81} + 242x^{80} -$
   $1103x^{79} - 85x^{78} - 375x^{77} - 652x^{76} + 660x^{75} + 795x^{74} + 209x^{73} - 722x^{72} + 10x^{71} + 581x^{70} + 29x^{69} - 82x^{68} + 351x^{67} -$
   $479x^{66} - 288x^{65} - 11x^{64} - 537x^{63} + 384x^{62} - 212x^{61} - 116x^{60} + 1215x^{59} - 210x^{58} + 289x^{57} + 152x^{56} + 415x^{55} + 181x^{54} +$
   $2150x^{53} - 223x^{52} + 760x^{51} + 379x^{50} + 276x^{49} + 315x^{48} + 1238x^{47} - 215x^{46} - 591x^{45} - 41x^{44} - 142x^{43} + 52x^{42} - 4x^{41} +$
   $464x^{40} + 847x^{39} + 544x^{38} - 927x^{37} + 150x^{36} - 576x^{35} + 160x^{34} - 366x^{33} + 163x^{32} - 95x^{31} + 168x^{30} - 699x^{29} + 416x^{28} -$
   $101x^{27} - 283x^{26} + 140x^{25} + 598x^{24} - 309x^{23} + 249x^{22} - 14x^{21} + 274x^{20} + 325x^{19} - 1825x^{18} + 1444x^{17} - 1069x^{16} +$
   $296x^{15} + 160x^{14} - 137x^{13} - 570x^{12} + 22x^{11} - 594x^{10} + 323x^{9} + 37x^{8} - x^{7} + 25x^{6} - 270x^{5} - 2982x^{4} - 262x^{3} - 499x^{2} -$
   $421x + 111)\mathcal{O}_K$
4. If we choose $e = 65537$ we obtain (in hexadecimal):
   $d = e^{-1} \bmod \phi(\alpha, \beta) = c4e8a6ca769901e83d2a8b2b98678ef60568fdf0037904341bc6479337cf1d62c8a51$
   $aef2f64b811296fb304009a45334e7e79fbe44ac9c90dddf3d83b59dc9add11702ed0ccfac47db045ff47f08$
   $fde54b9e480e394e48bd2da91f5b5254223f02e7a0c40e9bbaa262508137d3e6dab8301504e884204fc9a615$
   $93dd592adafcc2d09baeec59a8d081b3e98cb17fe6a426d3b35582f4eeefa82bb84482af8cf62ba2c3e0db7bd$
   $936a8d6b95d5326ce2fe5cac3d05d4028f38e7a6ea16d7ee570001.$

Now, we retrieve the secret decryption key d, given A and e:

1. Computing the norm of A gives (in hexadecimal):
   $\mathrm{N}(A) = 33bf97da2fddb268db31c6bb7a846d06d223f0faf9b3baee105a8152e1413a9f6d5da04bc33f794$
   $dfcc815ee61338c2f52ff51a05f34c583b84bb02a18b05d0505017dea0ea5936a75869717d49f15883366fa$
   $4fc4a42c4dfe92db3c15c034fb7b07a71477aa68d81c24bf5d9ab8f16c65b9257d355b29eabf38388da239$
   $a9d9492281bb44e87a8a41e2e134140ffdffa0ca5ad437b62a2a0128977a0f4e21de46de92f07b1f7c1e3fa$
   $77c663bb58d433114458551f469925568a474ed84263178101.$

2. The factorization into prime numbers gives (in hexadecimal):
   $p = 113bf6817372776d88f58cc40efc6434929b12638ad7f24fa63e8d61155ddce4b68b40459ac5eb56ed51d$
   $bf63ec3a6bf0479222441c655fef2012e9bd3e49d49b2b5546f81afd6c6227c0b7a5929fb8e5c032455fc01$
   and
   $q = 300addc0ab02f3bbb068e4d1e9ab36f095112becd16d1eaa28e617c27f6a672b9142ad875550e0525848a$
   $12e426e93650867335aabc528e7de363828d2f3b715ca7dd48a967220646200cdbf3ee2c109fa4420907830$
   $3e0eadc79dcbb379490c79ffb3ef0a08e80a6be6d58501.$
3. We compute the valuation of N at the prime integer p and q, and obtain $f_p = f_q = 1$.
4. We compute $\phi(A) = (p - 1)(q - 1)$ and then retrieve $d = e^{-1} \bmod \phi(A)$ and obtain the same value of d as above.

In the previous example, the primes p, q are not inert in the cyclotomic number field $K = \mathbb{Q}(\zeta_{256})$, and our goal in this example was to show the ability of retrieving the decryption d even for higher degree number field with NC property. Also, good to mention that our attack is applicable on general number fields not necessarily have the NC property.

5 Conclusion

In summary, this study provides insights on the high-dimensional RSA scheme's vulnerabilities to quantum polynomial attacks. By disproving the claim that the suggested construction provides post-quantum security that is more robust than NTRU [26], we highlight the significance of thorough study in the assessment of cryptographic primitives. Despite having a lattice structure, the suggested design is vulnerable to quantum attacks, since its hardness is not based on a lattice hard problem. Moreover, our results serve as a warning story, emphasizing that security against sophisticated attacks is not guaranteed simply by relying on a lattice structure.

Funding information: None declared.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors declare no conflict of interest/competing interest.

References

[1] Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96. New York, NY, USA: Association for Computing Machinery; 1996. p. 212–9. doi: 10.1145/237814.237866.

[2] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science; 1994. p. 124–34.

[3] Regev O. On lattices, learning with errors, random linear codes, and cryptography. New York, NY, USA: Association for Computing Machinery; 2005. doi: 10.1145/1060590.1060603.

[4] Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings. In: Gilbert H, editor. Advances in cryptology - EUROCRYPT 2010. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 1–23.

[5] Stehlé D, Steinfeld R, Tanaka K, Xagawa K. Efficient public key encryption based on ideal lattices. In: Matsui M, editor. Advances in cryptology - ASIACRYPT 2009. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. p. 617–35.

[6] Lindner R, Peikert C. Better key sizes (and attacks) for LWE-based encryption. In: Kiayias A, editor. Topics in Cryptology - CT-RSA 2011. Berlin, Heidelberg: Springer; 2011. p. 319–39.

[7] Peikert C. A decade of lattice cryptography; 2016. doi: http://dx.doi.org/10.1561/0400000074.

[8] Peikert C, Pepin Z. Algebraically structured LWE, Revisited. In: Hofheinz D, Rosen A, editors. Theory of Cryptography. Cham: Springer International Publishing; 2019. p. 1–23.

[9] Fouque PA, Hoffstein J, Kirchner P, Lyubashevsky V, Pornin T, Prest T, et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST's post-quantum cryptography standardization process. 2018;36(5):1–75.

[10] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, et al. CRYSTALS-Dilithium: A lattice-based digital signature scheme. Transactions on cryptographic hardware and embedded systems. 2018;2018(1):238–68.

[11] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, et al. CRYSTALS - Kyber: A CCA-Secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P); 2018. p. 353–67.

[12] Alagic G, Apon D, Cooper D, Dang Q, Dang T, Kelsey J, et al. Status report on the third round of the NIST post-quantum cryptography standardization process. USA: US Department of Commerce, NIST. 2022.

[13] Lenstra AK, Lenstra HW, Lovász LM. Factoring polynomials with rational coefficients. Mathematische Annalen. 1982;261:515–34. https://api.semanticscholar.org/CorpusID:5701340.

[14] Zhao Z, Ding J. Practical improvements on BKZ algorithm. In: Dolev S, Gudes E, Paillier P, editors. Cyber Security, Cryptology, and Machine Learning. Cham: Springer Nature Switzerland; 2023. p. 273–84.

[15] Albrecht M, Ducas L, Herold G, Kirshanova E, Postlethwaite E, Stevens M. The General Sieve Kernel and New Records in Lattice Reduction. In: EUROCRYPT 2019. Lecture Notes in Computer Science. Springer; 2019. p. 717–46.

[16] Kirchner P, Fouque PA. Revisiting lattice attacks on overstretched NTRU parameters. In: Coron JS, Nielsen JB, editors. Advances in cryptology - EUROCRYPT 2017. Cham: Springer International Publishing; 2017. p. 3–26.

[17] Micheli GD, Heninger N, Shani B. Characterizing overstretched NTRU attacks. J Math Cryptol. 2020;14(1):110–9. doi: 10.1515/jmc-2015-0055 [cited 2024-07-09].

[18] Takagi T, Naito S. Construction of RSA cryptosystem over the algebraic field using ideal theory and investigation of its security. Electron Commun Japan Part III Fund Electr Sci. 2000;83:19–29. https://api.semanticscholar.org/CorpusID:119513671.

[19] Zhiyong Z, Fengxia L, Man C. On the high-dimensional RSA algorithm–a public key cryptosystem based on lattice and algebraic number theory. In: Zheng Z, editor. Proceedings of the Second International Forum on Financial Mathematics and Financial Technology. Singapore: Springer Nature Singapore; 2023. p. 169–89.

[20] Lang S. Algebraic number theory. Graduate texts in mathematics. Springer-Verlag; 1994.

[21] Washington LC. Introduction to Cyclotomic Fields. Graduate Texts in Mathematics. New York: Springer; 1997.

[22] Murty MR, Esmonde J. Problems in algebraic number theory. vol. 190. Springer Science & Business Media; 2005.

[23] Cohen H. A course in computational algebraic number theory. Graduate Texts in Mathematics. Berlin Heidelberg: Springer; 2000. https://books.google.co.ma/books?id=hXGr-9l1DXcC.

[24] Regev O. On the complexity of lattice problems with polynomial approximation factors. In: Nguyen PQ, Vallée B, editors. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 475–96. doi: 10.1007/978-3-642-02295-1_15.

[25] Schnorr CP. A hierarchy of polynomial time lattice basis reduction algorithms. Theoretic Comput Sci. 1987;53(2):201–24. https://www.sciencedirect.com/science/article/pii/0304397587900648.

[26] Hoffstein J, Pipher J, Silverman JH. NTRU: A ring-based public key cryptosystem. In: International Workshop on Ant Colony Optimization and Swarm Intelligence; 1998.

[27] PARI/GP version 2.15.4. Univ. Bordeaux; 2023. http://pari.math.u-bordeaux.fr/.

Appendix

A Pari/GP demonstration

To obtain an idea of the comportment of our attack, we have used the Pari/GP software [27] to implement our attack on ideals which are the product of two prime ideals over power of two cyclotomic number fields, and summarized the result in Table A1.

Table A1: Average CPU time and wall time for different degrees of cyclotomic number fields

Degree	**1**	**1**	**2**	**4**	**8**
Average CPU time	143 ms	135 ms	163 ms	311 ms	424 ms
Average wall time	0 ms	0 ms	1 ms	1 ms	1 ms
Degree	**16**	**32**	**64**	**128**	**256**
Average CPU time	643 ms	1,007 ms	1,007 ms	1,427 ms	3,459 ms
Average wall time	1 ms	4 ms	1 ms	2 ms	9 ms
Degree	**512**	**1,024**			
Average CPU time	16,013 ms	1,56,591 ms			
Average wall time	72 ms	1,819 ms			

This research was supported through computational resources of HPC-MARWAN (hpc.marwan.ma) provided by the National Center for Scientific and Technical Research (CNRST), Rabat, Morocco. We run the code on HPC-Marwan Cluster, using a node of 2 cpus and 64 Go of memory, the running time to complete took 5 hours 12 minutes and 34 seconds. Our demonstrative code can be found in the following link: https://github.com/ndrahmani/Attack_code.

```
default("parisize", 64G);
default("timer", 1);
bit = 1024 ; /* Primes bit size */
ns = powers(2, 11);
verbose = 0 ; /* Set to 1 for printing messages, 0 to hide them */
NUM_TEST = 100 ; /* Number of tests */
NUM_PRIM = 2 ; /* Number of primes */


/* Function to safely read a text file if it exists */
read_field(filename) = {
    my (result);
    iferr(
        result = read(filename),
        E, /* If an error occurs (e.g., file not found) */
        result = 0;
    );
    result;
}
```

```
/* Function to write number field to a text file */
write_field(filename, field) = {
    write(filename, field); /* Create and write the number field to the file */
}

{
    for (n_i = 1, #ns -1,
      n = ns[n_i];
      filename = Str("cyclotomic_field_", n, ".txt");

      /* Attempt to read the number field from the file */
      K = read_field(filename);

      if (!K,
        /* File does not exist, compute the cyclotomic polynomial and number field */
        P = polcyclo(n);
        print1("Computing the ", n, "-th cyclotomic number field defined by the irreducible polynomial: ", P, " of degree: ", poldegree(P));
        K = nfinit(P);

        /* Store the number field in the file */
        write_field(filename, K);
        print("... Completed and stored.");
      , P = K.pol;
        print("Using Stored number field...nThe ", n, "-th cyclotomic number field\ndefined by the irreducible polynomial: ", P, " of degree: ", poldegree(P));
);

      if (verbose,
        print("/*********************************");
        print("Generating starting primes since ");
        print("we don’t have a quantum computer");
        print("**********************************/");
);

[S, F] = [0, 0];
if (verbose,   print("***************************");
   print("Number of tests: ", NUM_TEST);
   print("***************************");
);

t0 = getwalltime();
t0_ = gettime();

for (test = 1, NUM_TEST,
   /* Generate two random primes */
   kill(ps);
   ps = vector(NUM_PRIM, i, randomprime([2^(bit-1), 2^bit]));

   /* Decompose primes in the number field K */
```

```
  Ps = vector(#ps, i, idealprimedec(K, ps[i]));

  /* Generate a random ideal in K */
  A = idealmul(K, 1, 1);
  for (i = 1, #Ps,
    P_above_p_i = Ps[i][random([1, #Ps[i]])];
    P_exp_p_i =1 ;/* random([1, 32]);*/;
    A = idealmul(K, A, P_above_p_i);/*idealpow(K, P_above_p_i, P_exp_p_i));*/
  );

  /* STARTING THE ATTACK */
  /************************/
  /* Calculate the norm of the ideal A and find prime divisors */
  t0 = getwalltime();
  normA = idealnorm(K, A);
  p_div_normA = [];
  for (i = 1, #ps,
    if (normA % ps [i] == 0,
      p_div_normA = concat(p_div_normA, ps[i])
    )
  );

/* Verify all primes were used to construct A */
if (verbose, print("Assuring that we got all the primes that we constructed A from: ", ps ==
p_div_normA));

/* Decompose the primes that divide the norm of A */
if (verbose, print("Assume now that we factored the norm of the ideal A then we
decompose the primes and check for each prime divide A and get its valuation: "));
ind_pow = [];
for (i = 1, #p_div_normA,
  P_above_p = idealprimedec(K, p_div_normA[i]);
  for (j = 1, #P_above_p,
    e = idealval(K, A, P_above_p[j]);
    if (e != 0,
      ind_pow = concat(ind_pow, [[j, e]])
    )
  )
);

  /* Reconstruct the ideal A */
  new_A = idealmul(K, 1, 1);
  for (i = 1, #Ps,
    [pi, e_pi] = ind_pow[i];
    P_above_p_i = Ps[i][pi];
    P_exp_p_i = e_pi;
    new_A = idealmul(K, new_A, idealpow(K, P_above_p_i, P_exp_p_i));
  );
  t1 = getwalltime();
  /* Check if reconstruction was successful */
  if (new_A == A,
    S += 1;
    ,
    F += 1;
  );
  kill(new_A);
  kill(Ps);
  kill(ps);
  kill(A)
);

t1_ = gettime();
t1 = getwalltime();

print("NUMBER OF TESTS : ", NUM_TEST);
print("SUCCESS RATE : ", S/NUM_TEST * 100, " %");
print("AVERAGE WALL TIME: ", strtime(ceil((t1-t0)/NUM_TEST)));
print("AVERAGE CPU TIME : ", strtime(ceil((t1_-t0_)/NUM_TEST)));
print("=======================================================================");
print();
print();
);

print("DONE");
}
\q
```