Cryptanalysis of Compact-LWE Submitted to NIST PQC Project

Haoyu Li[1,2], Renzhang Liu[3], Yanbin Pan[1], Tianyuan Xie[1,2]

[1]Key Laboratory of Mathematics Mechanization, NCMIS,
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences,
Beijing 100049, China
[3] State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, China.

Abstract. Very recently, Liu, Li, Kim and Nepal submitted Compact-LWE, a new public key encryption scheme, to NIST as a candidate of the standard of post-quantum cryptography. About the security of Compact-LWE, the authors claimed that "*even if the hard problems in lattice, such as CVP and SIS, can be effciently solved, the secret values or private key in Compact-LWE still cannot be effciently recovered. This allows Compact-LWE to choose very small dimension parameters, such as $n = 8$ in our experiment*". However, in this paper, we show it is not true by proposing a ciphertext-only attack against Compact-LWE. More precisely, if we can solve CVP, we can decrypt any ciphertext without knowing the private keys. Since the dimension of the underlying lattice is very small (128) for the authors' parameter choice, (approximation-)CVP can be efficiently solved with lattice basis reduction algorithm. Hence, we can always break Compact-LWE with the authors' parameter choice in our experiments, which means that Compact-LWE with the recommended parameters is not secure.

Keywords: Ciphertext-only attack, lattice, LWE

1 Introduction

In December of 2017, NIST published the Round 1 submissions for the Post-Quantum Cryptography. Among all the candidate schemes, a new public key encryption scheme called Compact-LWE was proposed by Liu, Li, Kim and Nepal [7].

Compact-LWE has a similar structure with LWE [8], but with small samples and large errors. Based on the new features, Li, Kim and Nepal proposed the first version of Compact-LWE (we call Compact-LWE-α in this paper to distinguish it from the current version submitted to NIST) in an invited talk at ACISP 2017, see [5, 6]. However, Bootle and Tibouchi [1] showed that the particularly aggressive choice of parameters in Compact-LWE-α fails to achieve the stated security level.

Roughly speaking, to encrypt a message in Compact-LWE-α, the sender first computes a random subset sum of the Compact-LWE samples and then adds the message to the "error terms". Due to the special structure, Bootle and Tibouchi [1] proposed a very clever way to recover the message from the ciphertext by finding some lattice point close to the ciphertext (see [1] for more details).

In the current version of Compact-LWE submitted to NIST, Liu, Li, Kim and Nepal presented a much more complex method to generate the public key and private key. There are pare-wise dependent public keys instead of the independent public keys in Compact-LWE-α. The error terms are set to be $k_q^{-1}(sk \cdot u + r + ep)$, much more complex than $k_q^{-1}ep$ in Compact-LWE-α. Especially, they do not adds the message to the "error terms" to encrypt the message, but use a value related to u as an key to encrypt the message. Hence it seems we have to recover the key first instead of recovering the message directly as in Bootle and Tibouchi's attack.

They also increased the parameters, presented some analysis on the security of the improved Compact-LWE, and claimed that "*even if the hard problems in lattice, such as CVP and SIS, can be effciently solved, the secret values or private key in Compact-LWE still cannot be effciently recovered. This allows Compact-LWE to choose very small dimension parameters, such as $n = 8$ in our experiment*" [7]. Hence, Liu, Li, Kim and Nepal recommended small parameters, which make Compact-LWE efficient.

However, in this paper, we show that the claim is not true and it is insecure to choose small parameters. More precisely, we propose a ciphertext-only attack against Compact-LWE, which can decrypt any ciphertext without knowing the private keys. However, in the attack we need to find a short solution for a system of inhomogeneous linear equations, if we can solve CVP, we can find the short solution, hence we can break Compact-LWE, which contradicts the claim in [7].

For the small parameters recommended by the authors, it is easy to find the small solution by just lattice basis reduction. Hence, we can always break Compact-LWE with these small parameter choice in our experiments.

Due to our attack, it seems that Compact-LWE should enlarge its parameters to ensure the security, which will decrease the efficiency. Moreover, the security of Compact-LWE should be reevaluated more carefully.

2 Preliminaries

Notations Vectors are denoted by bold lowercase letters. Matrices are written in bold capital letters and row representation is used.

2.1 Lattices

Let $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\} \subset \mathbb{R}^m$ be a set of n linearly independent vectors. The lattice generated by the basis B is defined as

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

n and m are called the rank and dimension of the lattice respectively. The determinant of lattice $\mathcal{L}$ is defined as the volume of its parallelepiped, denoted by $\det(\mathcal{L})$. If $\boldsymbol{B}$ is a basis of $\mathcal{L}$, then $\det(\mathcal{L}) = \sqrt{\det(\boldsymbol{B}\boldsymbol{B}^T)}$. When $n = m$, $\det(\mathcal{L}) =\mid \det(\boldsymbol{B}) \mid$.

The parallelepiped spanned by $\boldsymbol{B}$ is defined as $\mathcal{P}(\boldsymbol{B}) = \{\boldsymbol{x}\boldsymbol{B} : \boldsymbol{x} \in [0,1)^n\}$. Nevertheless, we sometimes translate the domain to make it symmetric about the origin. The resulting domain is called the centered parallelepiped, denoted by $\mathcal{P}_c$. Mathematically, $\mathcal{P}_c(\boldsymbol{B}) = \{\boldsymbol{x}\boldsymbol{B} : \boldsymbol{x} \in [-1/2, 1/2)^n\}$.

For an ordered lattice basis $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$, the Gram-Schmidt orthogonalization $\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_n^\star$ can be efficiently computed by the recursion

$$\begin{aligned} \boldsymbol{b}_1^\star &= \boldsymbol{b}_1, \\ \boldsymbol{b}_i^\star &= \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\boldsymbol{b}_j^\star \text{ for } i = 2, \ldots, n, \end{aligned}$$

where $\mu_{i,j} = \langle \boldsymbol{b}_i, \boldsymbol{b}_j^\star \rangle / \langle \boldsymbol{b}_j^\star, \boldsymbol{b}_j^\star \rangle$. This procedure is dependent on the order of the basis. Hereafter, the Gram-Schmidt vectors corresponding to $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ are denoted by $\boldsymbol{B}^\star = \{\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_n^\star\}$. The determinant of $\mathcal{L}$ can be computed by the norm of the Gram-Schmidt vectors, namely, $\det(\mathcal{L}) = \prod_{i=1}^n \parallel \boldsymbol{b}_i^\star \parallel$.

2.2 SVP and CVP: Problems and Algorithms

SVP, the shortest vector problem, is one of the most famous hard problems in lattice. Given a lattice $\mathcal{L}$, the problem asks for a nonzero lattice point that is shortest.

SVP is known to be NP-hard under randomized reduction. However, there do exist polynomial-time algorithms to approximate SVP within exponential factor. The most famous algorithm is the LLL algorithm [4].

CVP, the closest vector problem, is one of the most famous hard problems in lattice. Given a lattice $\mathcal{L}$ and a target vector $\boldsymbol{t}$, the problem asks for a lattice point that is closest to $\boldsymbol{t}$.

The CVP is known to be NP-hard [10]. Even approximating CVP within almost polynomial factors is also proved to be NP-hard [2]. Therefore no polynomial-time algorithm is expected to be found. However, there do exist polynomial-time algorithms to approximate CVP within exponential factor. The most famous ones are Babai's rounding-off and nearest plane algorithms and the GPV algorithm [3], all of which start with an LLL-reduced [4] basis $\boldsymbol{B}$.

The rounding-off algorithm first computes the coefficients of $\boldsymbol{t}$ when represented by the basis $\boldsymbol{B}$, and then, as the name indicates, rounds the coefficients off to get a lattice point $\boldsymbol{v}_{\text{roa}}$ as output, that is, $\boldsymbol{v}_{\text{roa}} = \lceil \boldsymbol{t}\boldsymbol{B}^{-1} \rfloor \boldsymbol{B}$. It can be easily seen that $\boldsymbol{v}_{\text{roa}} - \boldsymbol{t} \in \mathcal{P}_c(\boldsymbol{B})$.

The nearest plane algorithm uses a more complicated method and can be described as Algorithm 1.

Algorithm 1 Babai's Nearest Plane Algorithm.

```
Input:    A good lattice basis $\boldsymbol{B}$ and a target vector $\boldsymbol{t}$;
Output:   A lattice vector close to $\boldsymbol{t}$;
1: $\boldsymbol{b} = \boldsymbol{t}$;
2: for $j = n$ to 1 do
3:     $\boldsymbol{b} = \boldsymbol{b} - c_j \boldsymbol{b}_j$ where $c_j = \lceil \frac{<\boldsymbol{b}, \boldsymbol{b}_j^\star>}{\boldsymbol{b}_j^\star, \boldsymbol{b}_j^\star} \rfloor$;
4: return $\boldsymbol{t} - \boldsymbol{b}$;
```

If we write $\boldsymbol{t} = \sum_{i=1}^n \alpha_i \boldsymbol{b}_i^\star$, it can be easily concluded that the algorithm outputs a lattice vector $\boldsymbol{v}_{npa} = \sum_{i=1}^n \beta_i \boldsymbol{b}_i = \sum_{i=1}^n \beta_i' \boldsymbol{b}_i^\star$ such that $\mid \beta_i' - \alpha_i \mid \leq 1/2$, where $\beta_i \in \mathbb{Z}$. Hence, $\boldsymbol{v}_{\mathrm{npa}} - \boldsymbol{t} \in \mathcal{P}_c(\boldsymbol{B}^\star)$.

The GPV algorithm is a randomized version of Babai's nearest plane algorithm. For Step 3 in Algorithm 1, GPV algorithm chooses a random c_j from a "discrete Gaussian distribution" and produces a random lattice vector which is close to the target. The readers are referred to [3] for more details.

2.3 Solving Short Solution by Lattice Basis Reduction

Given a matrix $A \in \mathbb{Z}_q^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Z}_q^n$, we show how to use algorithm for (approximation-)CVP to find a short solution $\boldsymbol{x} \in \mathbb{Z}^m$ such that $\boldsymbol{x}A = \boldsymbol{b} \mod q$.

Note that all the integer solutions for $\boldsymbol{x}A = \boldsymbol{0} \mod q$ form a lattice Λ. Given any solution $\boldsymbol{x}_0$ such that $\boldsymbol{x}_0 A = \boldsymbol{b} \mod q$, the set of all the integer solutions for $\boldsymbol{x}A = \boldsymbol{b} \mod q$ is obviously $\boldsymbol{x}_0 - \Lambda$. So we can find some lattice vector $\boldsymbol{v}$ in Λ, such that $\boldsymbol{v}$ is close to $\boldsymbol{x}_0$. Then $\boldsymbol{x}_0 - \boldsymbol{v}$ is a short solution.

Algorithm 2 Finding Small Solution with Algorithm for (Approximation-)CVP.

```
Input:    A matrix $A \in \mathbb{Z}_q^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Z}_q^n$;
Output:   A short solution $\boldsymbol{x}$ such that $\boldsymbol{x}A = \boldsymbol{b} \mod q$;
1: Find any solution $\boldsymbol{x}_0$ such that $\boldsymbol{x}_0 A = \boldsymbol{b} \mod q$;
2: Find a lattice vector $\boldsymbol{v}$ in $\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^m | \boldsymbol{x}A = \boldsymbol{0} \mod q\}$ close to $\boldsymbol{x}_0$ by the
   Algorithm for (approximation-)CVP.
3: return $\boldsymbol{x}_0 - \boldsymbol{v}$;
```

3 The Compact-LWE public key encryption scheme

3.1 The description of basic Compact-LWE

Public Parameters: The paramters are all positive integers:

- Moduli : q, t.
- Dimension: n
- Bounds: w, w', b, b'.

– Number of samples: m.

Private Parameters: The parameters p_size, s_max, e_min, e_max are all positive parameters and used to bound other parameters.

For the sake of correctness of the basic decryption algorithm and the security of the cryptosystem, [7] requires that

– $m > n + 2$
– $w > w'$, $(w - w') \cdot b' > t$
– q is the largest parameter
– t is a power of 2
– $sk\_max \cdot b' + p + e\_max \cdot p < q/(w + w')$

Key Generation: We use the notation $a \xleftarrow{\$} A$ to represent the random sampling of an element a from the set A.

The private key is generated in the following steps.

– $\boldsymbol{s}, \boldsymbol{s'} \xleftarrow{\$} \mathbb{Z}_q^n$
– $k, k' \xleftarrow{\$} \mathbb{Z}_q$ with k, k' coprime with q.
– $p \xleftarrow{\$} \{(w + w')b', \ldots, (w + w')b' + p\_size\}$ such that p is coprime with q.
– $ck, ck' \xleftarrow{\$} \mathbb{Z}_p$ with ck' is coprime with p.
– $sk, sk' \xleftarrow{\$} \mathbb{Z}_{sk\_max}$ such that $sk \cdot ck + sk' \cdot ck'$ is coprime with p.

The private key is

$$SK = (\boldsymbol{s}, \boldsymbol{s'}, k, k', p, ck, ck', sk, sk').$$

For $i \in \{1, 2, \ldots, m\}$, the generation of the public key is to collect m random compact-LWE samples.

– $\boldsymbol{a_i} \xleftarrow{\$} \mathbb{Z}_b^n$
– Random integers $e_i, e'_i \xleftarrow{\$} [e\_min, e\_max]$
– Random integer $u_i \xleftarrow{\$} \mathbb{Z}_{b'}$
– $r_i, r'_i \xleftarrow{\$} \mathbb{Z}_p$ satisfying $ck \cdot r_i + ck' \cdot r'_i = 0 \mod p$
– Compute k_q^{-1}, $k'^{-1}_q \in \mathbb{Z}_p$ such that $k_q^{-1} \cdot k = 1 \mod q$ and $k'^{-1}_q \cdot k' = 1 \mod q$
– Compute

$$pk_i = < \boldsymbol{a_i}, \boldsymbol{s} > + k_q^{-1} \cdot (sk \cdot u_i + r_i + e_i \cdot p) \mod q$$

$$pk'_i = < \boldsymbol{a_i}, \boldsymbol{s'} > + k'^{-1}_q \cdot (sk' \cdot u_i + r'_i + e'_i \cdot p) \mod q$$

Let $\boldsymbol{u} = (u_i)_{i=1}^m, \boldsymbol{pk} = (pk_i)_{i=1}^m, \boldsymbol{pk'} = (pk'_i)_{i=1}^m$ The public key is

$$PK = (\boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_m}, \boldsymbol{u}, \boldsymbol{pk}, \boldsymbol{pk'}).$$

Encryprion

To encrypt a message $v \in \mathbb{Z}_t$, we compute the ciphertext $\boldsymbol{c}$ as follows:

– Generate the m-dimensional random vector $\boldsymbol{l}$ such that the sum of all positive entries of $\boldsymbol{l}$ lies between w and $w + w'$, the sum of all negative entries of $\boldsymbol{l}$ between $-w'$ and 0.
– Compute $lu = \sum_{i=1}^{m} l_i \cdot u_i$, if $lu \leq 0$, then regenerate the $\boldsymbol{l}$, otherwise calculate $u_t = lu \mod t$, take u'_t the smallest integer equal or greater than lu/t coprime with t, encrypt v by computing

$$d = f(v, \sum_{i=1}^{m} l_i \cdot u_i)$$

where f is an efficiently computable function including two basic operations: XOR operation and bit shift operation. For more details about f see [7].
– Compute $la = \sum_{i=1}^{m} l_i \cdot \boldsymbol{a_i}$, $lpk = \sum_{i=1}^{m} l_i \cdot pk_i$ and $lpk' = \sum_{i=1}^{m} l_i \cdot pk'_i$, the ciphertext is $\boldsymbol{c} = (\boldsymbol{la}, d, lpk, lpk')$.

Decrption

Given the ciphertext $\boldsymbol{c} = (\boldsymbol{la}, d, lpk, lpk')$, the decryption algorithm recovers the message v in the following steps.

– Compute $d_1 = k \cdot (lpk - < \boldsymbol{a}, \boldsymbol{s} >) \mod q$, $d'_1 = k' \cdot (lpk' - < \boldsymbol{a}, \boldsymbol{s'} >) \mod q$ ($0 < d_1, d'_1 < q$).
– Compute $d_2 = ck \cdot d_1 + ck' \cdot d'_1 \mod p$ and $sckInv$ satisfying $sckInv \cdot (sk \cdot ck + sk' \cdot ck') = 1 \mod p$.
– Compute $d_3 = sckInv \cdot d_2 \mod p$.
– Compute $v = f^{-1}(d, d_3)$.

Remark 1. Liu, Li, Kim and Nepal [7] also proposed a general encryption algorithm, which encodes first and then encrypts messages. More precisely, after encoding the original message, the general encryption algorithm divides a long message into blocks and encrypts each block with the basic encryption algorithm above.

3.2 The choice of the parameters

In [7], Liu, Li, Kim and Nepal recommend the selections of parameters as follows for efficiency and evaluation.

Table 1. Public Parameters

q	t	n	m	w	w'	b	b'	l
2^{64}	2^{32}	8	128	224	32	16	2^{36}	8

4 Our Attack against Compact-LWE

It is enough to break the basic encryption algorithm in Compact-LWE.

Table 2. Private Parameters

sk_max	p_size	e_min	e_max
229119	2^{24}	457	3200

4.1 The key observation

We have the following key observations.

Lemma 1 (Informal). *Given a Compact-LWE ciphertext* $\boldsymbol{c} = (\boldsymbol{la}, d, lpk, lpk')$ *where*

$$\boldsymbol{la} = \sum_{i=0}^{m} \boldsymbol{l}_i \cdot \boldsymbol{a}_i$$
$$lpk = \sum_{i=0}^{m} \boldsymbol{l}_i \cdot pk_i \mod q$$
$$lpk' = \sum_{i=0}^{m} \boldsymbol{l}_i \cdot pk'_i \mod q,$$

if we could find a short enough vector $\boldsymbol{l}' = (\boldsymbol{l}'_1, \boldsymbol{l}'_2, \cdots, \boldsymbol{l}'_m)$*, such that*

$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot \boldsymbol{a}_i = \boldsymbol{la}$$
$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot pk_i = lpk \mod q$$
$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot pk'_i = lpk' \mod q,$$

then

$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot u_i = \sum_{i=0}^{m} \boldsymbol{l}_i \cdot u_i,$$

Proof (Sketch). By the process of decryption, with $\mathbf{la}, lpk, lpk'$, we can determine a unique $d_3 = \sum_{i=1}^{m} \boldsymbol{l}_i \cdot u_i$, which is used to recover the message from d.

If $\boldsymbol{l}'$ is short enough such that $0 < \sum_{i=1}^{m} \boldsymbol{l}'_i(sk \cdot u_i + r_i + e_i \cdot p) < q$ and $0 < \sum_{i=1}^{m} \boldsymbol{l}'_i(sk' \cdot u_i + r'_i + e'_i \cdot p) < q$, then by the same process of decryption with the same input $\mathbf{la}, lpk, lpk'$, we will obtain the exactly same value $d_3 = \sum_{i=1}^{m} \boldsymbol{l}'_i \cdot u_i$. Then the lemma follows.

Remark 2. In [7], the authors also considered the attack to ciphertexts. However, they thought that one need to guess the exact $\boldsymbol{l}$ to recover the message. Due to our attack, we do not need to guess the correct $\boldsymbol{l}$, another short $\boldsymbol{l}'$ can also help recover the message very well.

4.2 Our Ciphertext-Only Attack

Based on the lemma above, we present our attack.

Step 1 Given a Compact-LWE ciphertext $\mathbf{c} = (\mathbf{la}, d, lpk, lpk')$, find a short enough vector $\boldsymbol{l}' = (\boldsymbol{l}'_1, \boldsymbol{l}'_2, \cdots, \boldsymbol{l}'_m)$ by lattice basis reduction algorithm, such that

$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot \boldsymbol{a}_i = \mathbf{la}$$
$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot pk_i = lpk \mod q$$
$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot pk'_i = lpk' \mod q.$$

Step 2 Compute $\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot u_i$, and use it to recover the message by computing $f^{-1}(d, \sum_{i=0}^{m} \boldsymbol{l}'_i \cdot u_i)$.

4.3 Experimental Result

The most time-consuming computation in our attack is to find a short enough vector $\boldsymbol{l}' = (\boldsymbol{l}'_1, \boldsymbol{l}'_2, \cdots, \boldsymbol{l}'_m)$. However, since in the recommended parameters [7], $m = 128$, it is easy to find a short solution with Algorithm 2.

We randomly generated some Compact-LWE instances, and employed the LatticeSolve function in NTL [9] to compute $\boldsymbol{l}'$. In our experiments, all the $\boldsymbol{l}'$ returned by the LatticeSolve function satisfied

$$\sum_{i=0}^{m} \boldsymbol{l}'_i \cdot u_i = \sum_{i=0}^{m} \boldsymbol{l}_i \cdot u_i.$$

5 Conclusion

In this paper, we show that Compact-LWE with the recommended parameters is not secure.

Acknowledgement. We would like to thank Prof. Tancrède Lepoint for telling us the results about the former attack [1] against the former CompactLWE [5, 6].

References

1. J. Bootle and M. Tibouchi: Cryptanalysis of Compact-LWE. Available at https://eprint.iacr.org/2017/742.pdf

2. I. Dinur, G. Kindler, R. Raz, and S. Safra, Approximating CVP to within almost-polynomial factors is NP-hard. Combinatorica, 23:205 - 243, 2003.
3. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proc. of STOC, pages 197 - 206. ACM, 2008.
4. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients, Math. Ann. 261:513 - 534, 1982.
5. Dongxi Liu. Compact-LWE for lightweight public key encryption and leveled IoT authentication. In Josef Pierpzyk and Suriadi Suriadi, editors, ACISP 17, Part I, volume 10342 of LNCS, page xvi. Springer, Heidelberg, July 2017.
6. Dongxi Liu, Nan Li, Jongkil Kim, and Surya Nepal. Compact-LWE: Enabling practically lightweight public key encryption for leveled IoT device authentication. Cryptology ePrint Archive, Report 2017/685, 2017. http://eprint.iacr.org/2017/685.
7. Dongxi Liu, Nan Li, Jongkil Kim, and Surya Nepal: Compact-LWE: a Public Key Encryption Scheme. Available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Compact_LWE.zip
8. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005. pp. 84–93. ACM (2005)
9. V. Shoup. NTL: A library for doing number theory. Available at http://www.shoup.net/ntl/
10. P. van Emde Boas, Another NP-complete Problem and the Complexity of Computing Short Vectors in a Lattice, Tech. Report 81 - 04, Mathematische Instituut, University of Amsterdam, 1981.