A Faster Lattice Reduction Method Using Quantum Search

Christoph Ludwig

Technische Universität Darmstadt, Fachbereich Informatik
Alexanderstr. 10, 64283 Darmstadt, Germany
cludwig@cdc.informatik.tu-darmstadt.de

Abstract. We propose a new lattice reduction method. Our algorithm approximates shortest lattice vectors up to a factor $\leq (k/6)^{n/2k}$ and makes use of Grover's quantum search algorithm. The proposed method has the expected running time $O(n^3(k/6)^{k/8}A + n^4A)$. That is about the square root of the running time $O(n^3(k/6)^{k/4}A + n^4A)$ of Schnorr's recent random sampling reduction which in turn improved the running time to the fourth root of previously known algorithms. Our result demonstrates that the availability of quantum computers will affect not only the security of cryptosystems based on integer factorization or discrete logarithms, but also of lattice based cryptosystems. Rough estimates based on our asymptotic improvements and experiments reported in [1] suggest that the NTRU security parameter needed to be increased from 503 to 1277 if sufficiently large quantum computer were available nowadays.

Keywords: Lattice Reduction, Quantum Computers, NTRU, GGH

1 Introduction

Impact of Quantum Computers on Classical Cryptology. It is well known that quantum computers will be able to break cryptosystems that are based on the integer factorization problem or some discrete logarithm problem in polynomial time [2]. In particular, this affects RSA and elliptic curve cryptosystems, but also number field cryptosystems [3].

However, quantum computers are not believed to be able to solve NP-hard problems in polynomial time. The closest and shortest lattice vector problems (CVP and SVP) are known to be NP-hard [4,5,6]. Up to now, there was no evidence that the security of cryptosystems of GGH-type [7,8], that are based on SVP or CVP in arbitrary lattices, will be affected by the future availability of quantum computers. Neither is such a result known for NTRU. In fact, Regev's quantum reduction [9] of the $\Theta(n^{2.5})$-unique shortest vector problem applies to a class of lattices only which NTRU lattices do not belong to.

Classical Lattice Reduction Methods. Kannan's algorithm [10] computes a shortest lattice vector but it has an exponential running time. The renowned LLL algorithm [11] and its many variants compute in polynomial time a vector at most

T. Ibaraki, N. Katoh, and H. Ono (Eds.): ISAAC 2003, LNCS 2906, pp. 199–208, 2003.

$(4/3+\varepsilon)^{(n-1)/2}$ times as long as the shortest vectors in a given n-dimensional lattice. By applying Kannan's algorithm to blocks of length $2k$ in the lattice basis, Schnorr [12] improved the approximation factor of LLL to $(k/3)^{n/k}$ for sufficiently large k at the cost of an additional running time $O(n^3 k^{k+o(k)} A)$. (A covers the number of bit operations for the arithmetic on $O(n^2)$-bit integers.) The so called primal-dual method by Koy is claimed [13] to reduce the additional running time to $O(n^3 k^{k/2+o(k)} A)$ and still achieve an approximation factor $\leq (k/6)^{n/k}$. A variant of the $2k$-algorithm called BKZ (for Block Korkine-Zolotarev) [14] is widely used in practice, even though it is not proven to run in time polynomial in n.

Whenever the $2k$-method replaces the first base vector it takes only the first $2k$ base vectors into account. Schnorr [13] recently proposed an algorithm that is kind of complementary. It searches a replacement for the first base vector in the span of an LLL-reduced basis, but only the contribution of the last base vectors can be varied. If many such vectors are sampled, a sufficiently short vector will be found with high probability. The expected additional running time of this random sampling reduction (RSR) is $O(n^3 (k/6)^{k/4} A)$ and it guarantees an approximation factor $(k/6)^{n/2k}$. If the alloted running time is fixed, RSR reduces the approximation factor to about its 4th root compared with the primal-dual method.

Schnorr also proposed to replace the random sampling by a birthday sampling method that exploits the birthday paradox. The additional running time of his simple birthday reduction (SBR) is only $O(n^3 (4/3)^{k/3} (k/6)^{k/8} A)$, $k \geq 60$ according to [13], but it requires the storage of $(4/3)^{k/3}(k/6)^{k/8}$ additional lattice vectors. Even if $k = 60$ and $n = 100$, almost 10^{12} integers need to be stored. The massive space requirements raise doubts about the practicability of SBR.

Contribution of this Paper and Outline. We propose to replace the random sampling of vectors in Schnorr's algorithm by the technique of Grover's quantum search. We show that a quantum computer finds a sufficiently short vector with only $O((k/6)^{k/8})$ evaluations of a predicate that is as expensive to evaluate as one call to Schnorr's sampling algorithm. This leads to a quantum search reduction algorithm (QSR) that performs $O(n^3 (k^6)^{k/8} A)$ operations and achieves an approximation factor $\leq (k/6)^{n/2k}$. Hence, QSR improves in fixed time the approximation factor to about the square root compared with RSR and to about the 8th root compared with the primal-dual method.

Our result has an immediate effect on the security offered by all lattice based cryptosystems, including systems of GGH-type. But of particular interest is the impact of QSR on NTRU. If we transfer our improved running time bounds on the experimental results reported in [1] and require the security thresholds from the Lenstra-Verheul heuristic [15], we find that NTRU's security parameter need to be more than doubled. More precisely, the security parameter for NTRU's "highest security" parameter set had to be raised from 503 up to 1277 if quantum computers were available in 2005.

After some technical preliminaries and notations in section 2 we will outline Schnorr's random sampling reduction in section 3. Section 4 presents our proposed quantum reduction method. In section 5 we study the possible impact of the availability of sufficiently large quantum computers on NTRU. Section 6 points to possible further improvements of our algorithm and open research questions.

2 Preliminaries and Notation

All vector norms are Euclidean. A d-dimensional integral lattice $L = L(B)$ is the $\mathbb{Z}$-span of some linear independent lattice basis $B = \{b_1, \ldots, b_d\} \subset \mathbb{Z}^n$, i. e. $L = \{\sum_{i=1}^{d} a_i b_i : a_1, \ldots, a_d \in \mathbb{Z}\}$. By abuse of notation, we identify the basis B with the $n \times d$ matrix $B = [b_1, \ldots, b_d]$. For simplicity, we assume $d = n$.

We also assume $\max_j\{\|b_j\|\} = 2^{O(n)}$. Then the LLL algorithm operates on integers of bitlength $O(n^2)$. A denotes the the number of bit operations required for an arithmetic step on such integers.

Let $B = \widehat{B}R$ be the Gram-Schmidt decomposition of B, i. e. the columns $\hat{b}_j$ of $\widehat{B} \in \mathbb{Q}^{n\times n}$ are pairwise perpendicular and $R = (\mu_{i,j}) \in \mathbb{Q}^{n\times n}$ is unit upper triangular. In the following, whenever we pass B to an algorithm, we implicitly also pass $\widehat{B}$ and R.

B is δ-LLL reduced ($1/4 \leq \delta < 1$) if and only if

$$\begin{aligned} |\mu_{i,j}| &\leq 1/2 && \text{for all } 1 \leq i < j \leq n \quad \text{and} \\ \delta\|\hat{b}_j\|^2 &\leq \|\mu_{j,j+1}\hat{b}_j + \hat{b}_{j+1}\|^2 && \text{for all } 1 \leq j < n\,. \end{aligned}$$

Then the first basis vector b_1 satisfies $\|b_1\| \leq \left(\delta - \frac{1}{4}\right)^{-\frac{n-1}{2}} \lambda_1$, where $\lambda_1 = \min\{\|u\| : 0 \neq u \in L(B)\}$ is the length of the shortest nonzero lattice vectors.

For many applications of lattice theory an approximate solution of a *Shortest Vector Problem* (SVP) is required for some approximation factor α: Given a basis B, find a nonzero lattice vector $v \in L$ such that $\|v\| \leq \alpha\lambda_1$. In high dimensional lattices, this is infeasible for very small approximation factors; in fact, the problem is NP-hard for randomized reductions if $\alpha < \sqrt{2}$ [6]. The LLL algorithm computes solutions to SVP with approximation factor $\alpha = 2^{(n-1)/2}$, though.

3 Schnorr's Random Sampling Reduction

In 2001, Schnorr proposed a novel algorithm for approximate solutions of the SVP. We present here only the essential parts of the algorithm; for a detailed description as well as proofs, cf. [13].

RSR is built around the sampling algorithm (SA). SA randomly chooses lattice vectors with Gram-Schmidt coefficients $\nu_1, \ldots, \nu_n$ that satisfy

$$\begin{aligned} &\nu_j \in (-\tfrac{1}{2}, \tfrac{1}{2}] \text{for } 1 \leq j \leq n - k', \\ &\nu_j \in (-1, 1] \text{ for } n - k' < j < n, \qquad (1) \\ &\nu_n \in \{1, 2\} \end{aligned}$$

for some integer k'. Denote $D_{n,k'} := (-\frac{1}{2}, \frac{1}{2}]^{n-k'} \times (-1, 1]^{k'-1} \times \{1, 2\}$.

Algorithm 1 (Sampling Algorithm (SA)). *Given a lattice basis B and an integer $1 \leq k' < n$, SA returns in $O(n^2)$ arithmetic steps a uniformly chosen $b = \widehat{B}\nu \in L(B)$ such that $\nu \in D_{n,k'}$.*

Based on empiric data, Schnorr makes two assumptions:

Assumption 1 (Randomness Assumption (RA)). *The coefficient vector $\nu = (\nu_1, \ldots, \nu_n)^t$ sampled by SA satisfies the following conditions:*

1. *The random variables $\nu_1, \ldots, \nu_{n-1}$ are uniformly distributed in the intervals $(-\frac{1}{2}, \frac{1}{2}]$ and $(-1, 1]$, respectively.*
2. *The random variables $\nu_1, \ldots, \nu_{n-1}$ are pairwise statistically independent.*

Note that (RA) is crucial only for coefficients ν_j with small index j.

Assumption 2 (Geometric Series Assumption (GSA)). *There is $0 < q < 1$ such that $\|\hat{b}_j\|^2 = q^{i-1}\|b_1\|^2$ for $1 \leq j \leq n$.*

In practice, of course, (GSA) holds approximately only, but the analysis remains valid as long the approximation is good enough. Schnorr [13] outlines how to "repair" bases that do not approximate (GSA) by reducing subbases.

Under these assumptions, SA will eventually yield a short lattice vector after expected $O((k/6)^{(k-1)/4})$ iterations:

Algorithm 2 (Sample Short Vector (SHORT)). *Let B be a δ-LLL reduced basis and let $k \geq 24$ be an integer subject to*

$$n \geq 3(k+1) + \tfrac{k-1}{4} \log_2\left(\tfrac{k}{6}\right). \tag{2}$$

Assume (RA) and (GSA) with $q < (6/k)^{1/k}$. On input k and B, SHORT computes in average $O(n^2(k/6)^{(k-1)/4})$ arithmetic steps a vector $b \in L(B)$ satisfying $\|b\|^2 \leq 0.99\|b_1\|^2$.

Once we found a short lattice vector b, an LLL update (LLLU) replaces b_1 by b and LLL reduces the resulting bases again. Since it is merely an update, this algorithm requires only $O(n^3)$ arithmetic steps.

Algorithm 3 (Random Sampling Reduction (RSR)). *Let B be a δ-LLL reduced basis and let $k \geq 24$ be an integer subject to (2). On input k and B, RSR computes under (RA) and (GSA) in average*

$$O\Big(n^3\left(\tfrac{k}{6}\right)^{\frac{k-1}{4}} A + n^4 A\Big)$$

bit operations a still δ-LLL reduced basis $B' = [b'_1, \ldots, b'_n]$ satisfying

$$\|b'_1\| \leq \left(\tfrac{k}{6}\right)^{\frac{n}{2k}} \lambda_1 .$$

1: ***while*** $\|b_1\| > (k/6)^{(n-1)/2k}\|\hat{b}_n\|$ ***do*** `/*` $O(n)$ `iterations */`

2: $b \leftarrow \mathrm{SHORT}(B, k)$
3: $B \leftarrow \mathrm{LLLU}(B, b)$
4: ***end while***
5: ***return*** $B' \leftarrow B$

The loop condition implies $q < (6/k)^{1/k}$ whence the preconditions of SHORT are met. Since the input of RSR is already δ-LLL reduced, the approximation factor α after the i-th iteration satisfies $1 \leq \alpha \leq 0.99^i 2^{(n-1)/2}$. Therefore, RSR returns after $O(n)$ iterations.

All arithmetic steps operate on integers of length $O(n^2)$. Combining the complexity of SHORT and LLLU with the number of iterations in RSR, we get the average bit complexity $O(n^3(k/6)^{(k-1)/4}A + n^4 A)$.

4 Quantum Search Reduction

Algorithm SHORT searches in the unsorted finite set of coefficient vectors $\nu \in D_{n,k'}$ with $\widehat{B}\nu \in L(B)$ for an element such that $\|\widehat{B}\nu\|^2 \leq 0.99\|b_1\|^2$. This is a setup where Grover's quantum search algorithm outperforms all classical search algorithms. For details on quantum computing in general and Grover's quantum search in particular we refer to [16] and [17,18], respectively.

Grover's quantum search QS makes use of a quantum operator called *black box oracle*. Let $S = \{0, \ldots, N-1\}$, $N = 2^n$. Given a (classical) algorithm that evaluates the predicate $f : S \to \{0, 1\}$, we can easily construct a black box oracle O_f such that QS finds some $s \in S$ with $f(s) = 1$. More precisely, if there is a uniform circuit family evaluating f with $O(T_f(n))$ gates then O_f requires $O(T_f(n))$ quantum operations and ancilla qubits.

Algorithm 4 (Quantum Search (QS)). *Assume $M := |f^{-1}(\{1\})| > 0$. On input a black box algorithm O_f the quantum algorithm QS returns some $s \in S$ such that $f(s) = 1$.*

QS makes expected $\Theta((N/M)^{1/2})$ queries to O_f (even if M is unknown) and applies additional expected $\Theta((N/M)^{1/2})$ quantum operations on its $n+1$ qubit register.

Remark 1. QS can be easily modified to handle the case $M = 0$ at the cost of a small error probability [18]. However, in our particular application we know $M \geq 1$ whence we do not have to deal with sporadic errors

The idea underlying the quantum search reduction is to replace algorithm SHORT by a quantum search for a vector b satisfying $\|b\|^2 \leq 0.99\|b_1\|^2$. More precisely, we look for some sufficiently short b in

$$V_{B,k} = \left\{ v \in L(B) : v = \widehat{B}\nu,\ \nu \in D_{n,k'} \text{ with } k' = 1 + \left\lceil \tfrac{k-1}{4} \log_2\left(\tfrac{k}{6}\right) \right\rceil \right\} .$$

Let $N = 2^{k'} = \min\left\{2^t : 2^t \geq 2\left(\tfrac{k}{6}\right)^{(k-1)/4}\right\}$. There is a (classical) $O(n^2 A)$-time algorithm that enumerates $V_{B,k}$. In particular, $|V_{B,k}| = N$. The algorithm mimics Schnorr's algorithm SA; only the random bits are replaced by the input index.

Algorithm 5 (Enumerate $V_{B,k}$ (ENUM)). *Let B be a δ-LLL reduced basis with Gram-Schmidt decomposition $\widehat{B}R$, $R = [\mu_1, \ldots, \mu_n]$ and $k \geq 24$ be an integer subject to (2). On input B, k, and an index $0 \leq i < N$, ENUM computes in $O(n^2)$ arithmetic steps the vector v_i for some enumeration of $V_{B,k}$.*

$i_0 \leftarrow i \bmod 2$, $i \leftarrow \lfloor i/2 \rfloor$
$\nu = (\nu_1, \ldots, \nu_n)^t \leftarrow (i_0 + 1)\mu_n$, $b \leftarrow \nu_n b_n$
for $j = n-1$ ***downto*** 1 ***do***
    if $j \leq n - 1 - \left\lceil \frac{k-1}{4} \log_2\left(\frac{k}{6}\right) \right\rceil$ ***then***
        $c \leftarrow \lceil \nu_j \rfloor$
    else
        $i_0 \leftarrow i \bmod 2$, $i \leftarrow \lfloor i/2 \rfloor$
        $c \leftarrow \lceil \nu_j \rceil - i_0$
    end if
    $b \leftarrow b - cb_j$, $\nu \leftarrow \nu - c\mu_j$
end for
return b

The vector returned by ENUM satisfies (1) because R is unit upper triangular. Since we restricted the coefficients in (1) to half-open intervals, the enumeration of $V_{B,k}$ is exhaustive.

The oracle black box of the quantum search is based upon the predicate $f_{B,k} : \{0, \ldots, N-1\} \to \{0, 1\}$ with $f_{B,k}(i) = 1$ if and only if $\|\mathrm{ENUM}(B, k, i)\|^2 \leq 0.99\|b_1\|^2$. The evaluation of $f_{B,k}$ requires $O(n^2)$ arithmetic steps on integers of length $O(n^2)$, whence $O_{f_{B,k}}$ requires $O(n^2 A)$ quantum operations and $O(n^2 A)$ ancilla qubits.

We then have the following trivial algorithm to find sufficiently short vectors:

Algorithm 6 (Quantum Short Vector Search (QSHORT)). *Let B be a δ-LLL reduced basis and $k \geq 24$ be an integer subject to (2). On input B and k, QSHORT computes under (RA) and (GSA) with expected $O(n^2 (k/6)^{k/8} A)$ operations on $O(n^2 A)$ qubits a lattice vector $b \in L(B)$ satisfying $\|b\|^2 \leq 0.99\|b_1\|^2$.*

$i \leftarrow \mathrm{QS}(O_{f_{B,k}})$
$b \leftarrow \mathrm{ENUM}(B, k, i)$
return b

Schnorr [13] shows $\Pr[\|b\|^2 \leq 0.99\|b_1\|^2] \geq \frac{1}{2}\left(\frac{k}{6}\right)^{(1-k)/4}$ under (RA) and (GSA) provided b is a vector sampled by SA. Since SA returns elements uniformly chosen from $V_{B,k}$, we have

$$M = N \Pr[\|b\|^2 \leq 0.99\|b_1\|^2] \geq 2\left(\frac{k}{6}\right)^{\frac{k-1}{4}} \frac{1}{2}\left(\frac{k}{6}\right)^{\frac{1-k}{4}} = 1\,.$$

Therefore, QSHORT makes under (RA) and (GSA) expected

$$\Theta((N/M)^{1/2}) = O(N^{1/2}) = O((\tfrac{k}{6})^{(k-1)/8})$$

queries to the black box $O_{f_{B,k}}$. The total number of expected elementary operations is $O(n^2(k/6)^{(k-1)/8}A)$. The space requirements of QSHORT are dominated by the black box $O_{f_{B,k}}$.

Replacing SHORT by QSHORT, RSR becomes a quantum algorithm QSR that achieves the same approximation factor with significantly less elementary operations.

Algorithm 7 (Quantum Search Reduction (QSR)). *Let $B = [b_1, \dots, b_n]$ be a δ-LLL reduced basis and let $k \geq 24$ be an integer subject to (2). On input B and k, QSR computes under (RA) and (GSA) a still δ-LLL reduced basis $B' = [b'_1, \dots, b'_n]$ satisfying*

$$\|b'_1\| \leq \left(\tfrac{k}{6}\right)^{\frac{n}{2k}} \lambda_1 \,.$$

QSR performs on average

$$O\left(n^3 \left(\tfrac{k}{6}\right)^{(k-1)/8} A + n^4 A\right)$$

operations.

1: **while** $\|b_1\| > (k/6)^{(n-1)/2k} \|\hat{b}_n\|$ **do** `/*` $O(n)$ `iterations */`
2: $b \leftarrow \mathrm{QSHORT}(B, k)$
3: $B \leftarrow \mathrm{LLLU}(B, b)$
4: **end while**
5: **return** B

Like RSR, QSR executes the loop body $O(n)$ times and each iteration requires $O(n^2(k/6)^{(k-1)/8}A + n^3 A)$ operations, yielding the stated operation bound.

5 Impact on NTRU

We discuss the impact of our proposed reduction algorithm QSR on NTRU if quantum computers were available. The NTRU cryptosystem attracted a lot of attention since it is very efficient. It is being standardized by the IEEE P1363 workgroup; another standard has already been published by the Consortium for Efficient Embedded Security [19].

The one-wayness of NTRU is based on the hardness of SVP in a certain class of lattices generated by convolution matrices. The resistance of NTRU against lattice reduction attacks has been studied in [1, §4.2 and Appendix] and [20]. The authors of these papers report experiments on a 200 MHz PC with the BKZ implementation found in Shoup's NTL library [21]. They tried to recover private keys in lattice dimension $2N$, $75 \leq N \leq 108$, for parameter sets relating to "moderate", "high", and "highest" security. We are not aware of any more recent data on lattice attacks against the one-wayness of NTRU.

It is noticeable that in the experiments, the block size had to be increased very quickly. For $N = 75$ a block size k between 4 and 6 sufficed to approximate the corresponding SVP well enough, but for $N = 108$ the required block size

Table 1. Estimated running time for recovering private NTRU keys (in MIPS-years)

N	$t_{\mathrm{BKZ}}(N)$	$t_{\mathrm{RSR}}(N)$	$t_{\mathrm{QSR}}(N)$
503	$1.2 * 10^{30}$	$3.3 * 10^{8}$	$5.8 * 10^{3}$
709	$6.5 * 10^{46}$	$\mathbf{2.8 * 10^{11}}$	$5.3 * 10^{5}$
809	$2.7 * 10^{53}$	$\mathbf{2.3 * 10^{13}}$	$4.8 * 10^{6}$
1277	$1.4 * 10^{89}$	$1.9 * 10^{22}$	$\mathbf{1.4 * 10^{11}}$
1511	$9.7 * 10^{106}$	$5.6 * 10^{26}$	$\mathbf{2.4 * 10^{13}}$

[15] considers $1.02 * 10^{11}$ and $2.07 * 10^{13}$ MIPS-years infeasible in 2005 and 2015, respectively.

was already $k = 22$. From their experiments the authors extrapolated the running time t necessary to recover an NTRU key generated for highest security. Assuming a second on a 200 MHz PC is equivalent to 200 MIPS-seconds, they found

$$t_{\mathrm{BKZ}}(N) \geq e^{0.17564N-19.04795} \text{ MIPS-years}.$$

The estimated cost of their attack for $N = 503$ is about 10^{30} MIPS-years. According to the Lenstra-Verheul heuristic [15], even $3*10^{21}$ MIPS-years are infeasible until 2050, i. e. NTRU's security margin with respect to this attack seemed plenty.

However, recall that QSR reduces the running time to about the 8th root compared with Koy's primal-dual method. The primal-dual method is already supposed to perform better than the BKZ reduction used in [1]. As a first approximation, we therefore estimate the running time of an attack with QSR as

$$t_{\mathrm{QSR}}(N) \geq e^{(0.17564N-19.04795)/8} \text{ MIPS-years}.$$

Therefore, keys generated for NTRU-503 will be recovered after $\approx 10^{30/8} = 10^{3.75}$ MIPS-years and NTRU-503 cannot be considered secure anymore once QSR can be implemented.

But the speedup by QSR is only polynomial, whence the NTRU scheme itself won't be broken by QSR. It is sufficient to multiply NTRU's security parameter with a constant factor. Lenstra and Verheul claim that a running time of $1.02 * 10^{11}$ MIPS-years will be infeasible in 2005, $2.07 * 10^{13}$ will be infeasible in 2015. By our rough estimate, it would only be infeasible to recover an NTRU key in 2005 if $N \geq 1277$. Tab. (1) gives an overview of the estimated running times for recovering a private NTRU key generated with the parameters proposed for NTRU-503 if the attacker uses the BKZ implementation from NTL, Schnorr's random sampling reduction, and the proposed quantum search reduction, respectively. The shown values of N are minimal primes that can be considered secure against attacks with the RSR and QSR algorithm in 2005 and 2015, respectively.

6 Further Improvements and Research

Schnorr [13] reports that a variant of RSR that replaces any one of the first ten base vectors and updates the basis by BKZ rather than LLL is very effective. His extended sampling algorithm ESHORT returns a pair (b, i) such that $\|\pi_i(b)\|^2 = \sum_{j=i}^{n} \nu_j \|\hat{b}_j\|^2 \leq 0.99 \|\hat{b}_j\|^2$. We can implement an analog quantum search algorithm QESHORT by straightforward modifications to our predicate f. The time bounds for QESHORT do not change. Of course, we cannot bound the overall running time of the resulting reduction algorithm since we have no proven time bound for the BKZ algorithm.

As mentioned before, Schnorr also proposes a sampling reduction that exploits the birthday paradox. Unfortunately, he has to trade very much space for the additional speedup whence it is doubtful whether the simple birthday reduction (SBR) is practical. Anyway, the birthday paradox has also been used to accelerate Grover's search algorithm. Brassard, Høyer and Tapp [22] proposed an quantum algorithm that finds a collision in a hash function $h : X \to Y$ with at most $O(N^{1/3})$ evaluations of h, $N = |X|$. Thus, on the first glance, it seemed possible to construct a quantum variant of SBR that performs estimated $O(n^3 (4/3)^{k/3} (k/6)^{k/12} A + n^4 A)$ operations. Unfortunately, our attempt failed since [22] requires $N \geq 2|Y|$ which does not hold if we follow the construction of SBR. It therefore stays an open question whether QSR allows an additional speedup by a time-space trade-off.

7 Conclusion

We presented a quantum algorithm QSR that approximates shortest lattices vectors up to a factor $\leq (k/6)^{n/2k}$ where n is the lattice dimension and $k \geq 24$ is an almost arbitrary parameter. The expected running time of our algorithm is $O(n^3 (k/6)^{k/8} A + n^4 A)$ which is roughly the square root of the running time of the fastest known classical algorithm RSR. We reconsidered the security analysis of NTRU and found that an attack against NTRU-503 with our algorithm required only (roughly) estimated $5.8 * 10^3$ MIPS-years. An attack with QSR against NTRU would be infeasible only if NTRU's security parameter was raised up to 1277.

References

1. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In Buhler, J.P., ed.: Algorithmic Number Theory (ANTS III). Volume 1423 of LNCS., Springer-Verlag (1998)
2. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26** (1997) 1484–1509
3. Hallgren, S.: Polynomial-time quantum algorithm for Pell's equation and the principal ideal problem. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, ACM Press (2002)

4. Emde Boas, P.v.: Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, University of Amsterdam, Department of Mathematics, Netherlands (1981)
5. Ajtai, M.: The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ACM Press (1998) 10–19
6. Micciancio, D.: The shortest vector in a lattice is hard to approximate to within some constant. In: IEEE Symposium on Foundations of Computer Science. (1998) 92–98
7. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In Kaliski, Jr., B.S., ed.: Advances in Cryptology – Crypto'97. Volume 1294 of LNCS., Springer-Verlag (1997) 112–131
8. Micciancio, D.: Improving lattice based cryptosystems using the Hermite normal form. In Silverman, J.H., ed.: Cryptography and Lattices. Volume 2146 of LNCS., Springer-Verlag (2001) 126–145
9. Regev, O.: Quantum computations and lattice problems. In: The 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02), IEEE (2002) 520–529
10. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Research **12** (1987) 415–440
11. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261** (1982) 515–534
12. Schnorr, C.P.: A hierachy of polynomial lattice basis reduction algorithms. Theoretical Computer Science **53** (1987) 201–224
13. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In Alt, H., Habib, M., eds.: STACS 2003: 20th Annual Symposium on Theoretical Aspects of Computer Science. Volume 2607 of LNCS., Springer (2003) 146–156
14. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Math. Programming **66** (1994) 181–199
15. Lenstra, A.K., Verheul, E.R.: Selecting cryptographic key sizes. J. Cryptology **14** (2001) 255–293
16. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC), ACM Press (1996) 212–219
18. Boyer, M., Brassard, G., Høyer, P., Tapp, A.: Tight bounds on quantum searching. arXiv e-print quant-ph/9605034 (1996)
19. Consortium for Efficient Embedded Security: EESS #1: Implementation aspects of NTRUEncrypt and NTRUSign. `http://www.ceesstandards.org/documents/EESS1_11122002_v2.pdf` (2002) Version 1.0.
20. Silverman, J.: Estimated breaking times for NTRU lattices. Technical Report 12, NTRU Cryptosystems, Inc. (1999)
21. Shoup, V.: NTL – a library for doing number theory. URL `http://www.shoup.net/ntl/index.html` (2001) Release 5.2.
22. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In Lucchesi, C., Moura, A., eds.: LATIN'98: Theoretical Informatics. Volume 1380 of LNCS., Springer-Verlag (1998)