Practical Lattice Basis Sampling Reduction

Johannes Buchmann and Christoph Ludwig

Technische Universität Darmstadt, Fachbereich Informatik
Hochschulstr. 10, 64289 Darmstadt, Germany
{buchmann,cludwig}@cdc.informatik.tu-darmstadt.de

Abstract. We propose a practical sampling reduction algorithm for lattice bases based on work by Schnorr [1] as well as two even more effective generalizations. We report the empirical behaviour of these algorithms. We describe how Sampling Reduction allows to stage lattice attacks against the NTRU cryptosystem with smaller BKZ parameters than before and conclude that therefore the recommeded NTRU security parameters offer ≤ 74 Bit security.
Keywords: lattice basis reduction, NTRU

1 Introduction

Lattice basis reduction, in particular the renowned LLL algorithm [2], has long been established as a powerful tool in cryptanalysis, e. g. [3, 4]. On the other hand, several cryptosystems were proposed over the last decade that are based on the hardness of certain lattice problems. Some of them are of mostly theoretical interest [5–7], but the NTRU cryptosystem [8, 9] is used in practice and actively marketed.

The key sizes that need to be selected in order for the system to be secure depend on the efficiency of the best algorithm for computing short vectors in lattices. Currently, that algorithm is the Block Korkine-Zolotarev (BKZ) reduction algorithm [10]. In 2003 Schnorr presented Random Sampling Reduction (RSR) [1], a new algorithm for computing short vectors in lattices. Assuming the so called geometric series assumption (GSA), RSR asymptotically outperforms BKZ. However, RSR is not a practical algorithm since the parameter choice in RSR depends on the GSA. But in general, GSA is not satisfied.

In this paper, we present *Sampling Reduction* (SR), a practical algorithm based on RSR. We report on experiments that demonstrate that the shortest vector found by SR is significantly shorter than the shortest vector found by BKZ. We also propose two generalizations of SR that generate lattice bases with even more short vectors. We describe successful attacks on low dimensional NTRU lattice bases that require smaller

BKZ parameters than previous attacks that used BKZ only. We conclude under reasonable assumptions that the recommended NTRU parameters [8] do not offer more than 74 Bit security.

2 Notation and Definitions

We assume the Euclidean metric on $\mathbb{R}^d$. A lattice L is a discrete subgroup of $\mathbb{R}^d$, its dimension is $\dim(L) := \dim(L \otimes_{\mathbb{R}} \mathbb{R})$. The first minimum of L is $\lambda_1(L) := \min\{\|x\| \,|\, 0 \neq x \in L\}$.

For any n-dimensional lattice L, $n \geq 1$, there are ordered bases $B = [b_1, \ldots, b_n] \in \mathbb{R}^{d \times n}$ such that $L = L(B) := \{v \,|\, v = Bx \text{ for some } x \in \mathbb{Z}^n\}$. Given an ordered basis B, the set of all bases of $L(B)$ is $\{BU \,|\, U \in \mathbb{Z}^{n \times n} \text{ and } \det U = \pm 1\}$. We consider integer coefficient lattices only whence $B \in \mathbb{Z}^{d \times n}$.

Let $B = \widehat{B}R$ be the Gram-Schmidt decomposition of B, i. e. the columns $\hat{b}_j$ of $\widehat{B} \in \mathbb{Q}^{d \times n}$ are pairwise perpendicular and $R = (\mu_{i,j}) \in \mathbb{Q}^{n \times n}$ is unit upper triangular. Let $\pi_i : \mathbb{R}^d \to \text{lin}\{b_1, \ldots, b_{i-1}\}^{\perp}$ be the orthogonal projection onto the orthogonal space of the first $i-1$ base vectors. Denote $L_{i,\beta}(B) = L([\pi_i(b_i), \ldots, \pi_i(b_{\min\{i+\beta-1,n\}})])$.

Given a generating system of L and parameters (δ, β) with $1/2 < \delta < 1$ and $2 \leq \beta \in \mathbb{N}$, the BKZ algorithm [10] computes a (δ, β)-BKZ reduced basis of L. A (δ, β)-BKZ reduced basis B satisfies

$$|\mu_{i,j}| \leq 1/2 \qquad \text{for all } 1 \leq i < j \leq n, \qquad \text{(size condition)}$$

$$\delta \|\hat{b}_i\|^2 \leq \lambda_1(L_{i,\beta}(B)) \quad \text{for all } 1 \leq i \leq n. \qquad \text{(BKZ condition)}$$

In the course of BKZ reduction we also obtain the Gram-Schmidt coefficient matrix R as well as $\|\widehat{\boldsymbol{b}}_i\|^2$ for $i = 1, \ldots, n$. LLL reduction is the special case of BKZ reduction with $\beta = 2$.

Throughout this paper $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$ denotes a (δ, β)-BKZ reduced ordered lattice bases with Gram-Schmidt decomposition $B = \widehat{B}R$, $\widehat{B} = [\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_n]$, $R = (\mu_{i,j})$. All lattice points belong to the n-dimensional lattice $L = L(B)$. Even though B is updated in the course of the reduction, L stays always the same.

3 The Sampling Reduction

In this section we describe our Sampling Reduction Algorithm (SR) which is based on Schnorr's Random Sampling Reduction.

The idea of SR is as follows. SR operates on a generating system G of an n-dimensional lattice L. SR applies (δ, β)-BKZ reduction to G and

Algorithm 1 Sampling Reduction (SR)

Input: generating system G of L, reduction factor γ, search space parameter $u_{\max}$, BKZ parameters (δ, β)

Output: $(B, \textit{reason})$ where B is a (δ, β)-BKZ reduced basis of L and *reason* indicates why the algorithm terminated.

procedure SR$(G, \gamma, u_{\max}, \delta, \beta)$
  $(B, \mathfrak{b}, R) \leftarrow$ BKZ(G, δ, β) /* $B = \widehat{B}R$, $\mathfrak{b} = (\|\widehat{\boldsymbol{b}}_1\|^2, \ldots, \|\widehat{\boldsymbol{b}}_n\|^2)$ */
  if $-$BestBound$(\mathfrak{b}, u_{\max}, \gamma) > u_{\max}$ **then**
    return $(B$, "success probability too small")
  else
    for $l = 1, \ldots, 2^{u_{max}}$ **do**
      $\boldsymbol{v} \leftarrow$ Sample(B, R, l)
      if $\|\boldsymbol{v}\|^2 \leq \gamma \|\boldsymbol{b}_1\|^2$ **then**
        return SR$([\boldsymbol{v}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n], \gamma, u_{\max}, \delta, \beta)$
      end if
    end for
    return $(B$, "search space exhausted")
  end if
end procedure

obtains the BKZ reduced basis B. As long as the success probability – determined by the function BestBound – is sufficiently high, SR uses the sampling strategy Sample described below to find vectors that are significantly shorter than the first vector of B. Whenever such a vector $\boldsymbol{v}$ is found, SR is applied to the generating system that consists of $\boldsymbol{v}$ and the vectors in B. The input variable $u_{\max} \in \mathbb{N}$ limits the amount of work SR spends on sampling vectors.

SR always terminates due to the following lemma.

Lemma 1. *The recursion depth x of* SR$(G, \gamma, u_{\max}, \delta, \beta)$ *is bound by* $x \leq (n-1)\log_\gamma(\delta - 1/4)$.

Proof. SR operates on (δ, β)-BKZ reduced and thus δ-LLL reduced bases. Therefore, $\|\boldsymbol{b}_1\| \leq (\delta - 1/4)^{\frac{1-n}{2}} \lambda_1(L)$ [2]. BKZ reduction never increases the length of the first vector in the generating system. Each recursion decreases the length of the first base vector by a factor at most $\sqrt{\gamma} < 1$, and $\boldsymbol{b}_1$ cannot be shorter than $\lambda_1(L)$. Hence, $\gamma^x(\delta - 1/4)^{1-n} \geq 1$ which implies the lemma. □

3.1 The Sampling Algorithm

The actual sampling of vectors in SR is performed by the subalgorithm Sample described here. Sample is similar to Schnorr's algorithm SA [1]

except that the (pseudo-)random coin toss in the latter is replaced by an interger input argument l.

The task of SAMPLE is to generate lattice points that are likely to be short. Because of $\|\boldsymbol{v}\|^2 = \sum_{i=1}^{n} \nu_i^2 \|\widehat{\boldsymbol{b}}_i\|^2$, it is plausible to expect that a lattice point $\boldsymbol{v}$ is short if all Gram-Schmidt coefficients ν_i are small. Therefore, SAMPLE enumerates lattice points with all $|\nu_i| \leq 1$.

To be precise, let $2^{u-1} < l \leq 2^u$. Then $\boldsymbol{v} = \text{SAMPLE}(B, R, l) = \sum_{i=1}^{n} \nu_i \widehat{\boldsymbol{b}}_i$ satisfies

$$\nu_i \in \begin{cases} (-\frac{1}{2}, \frac{1}{2}] & \text{for } 1 \leq i < n-u, \\ (-1, 1] & \text{for } n-u \leq i < n, \\ \{1\} & \text{for } i = n. \end{cases} \tag{SC}$$

Algorithm 2 SAMPLE

Input: unit upper triangular matrix $R = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n] \in \mathbb{Q}^{n \times n}$, lattice basis $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n] \in \mathbb{Z}^{n \times n}$ with Gram-Schmidt decomposition $B = \widehat{B}R$, $1 \leq l \leq 2^{n-1}$
Output: $\boldsymbol{v} \in L(B)$ subject to (SC)

```
procedure SAMPLE(B, R, l)
    v ← b_n, ν = (ν_1, ..., ν_n)^t ← r_n
    for i = n − 1, n − 2, ..., 1 do
        x ← ⌈ν_i − 1/2⌉                          /* −1/2 < ν_i − x ≤ 1/2 */
        if l mod 2 = 1 then
            if ν_i − x ≤ 0 then
                x ← x − 1                        /* 1/2 < ν_i − x ≤ 1 */
            else
                x ← x + 1                        /* −1 < ν_i − x ≤ −1/2 */
            end if
        end if
        l ← l div 2
        v ← v − x b_i, ν ← ν − x r_i             /* ν_i ← ν_i − x */
    end for
    return v
end procedure
```

Let $i \in \{1, \ldots, n\}$. The choice of ν_i in $\boldsymbol{v} = \sum x_j \boldsymbol{b}_j = \sum \nu_j \widehat{\boldsymbol{b}}_j$ does not affect $\nu_{i+1}, \ldots, \nu_n$ since R is unit upper triangular. Therefore, SAMPLE computes (x_i, ν_i) by iteration based on $x_n = \nu_n = 1$. Assume the coefficients $(x_{i+1}, \nu_{i+1}), \ldots, (x_n, \nu_n)$ are already fixed. Then SAMPLE determines the unique $x' \in \mathbb{Z}$ with $\pi_i(x'\boldsymbol{b}_i + \sum_{j=i+1} x_j \boldsymbol{b}_j) = \nu' \widehat{\boldsymbol{b}}_i + \sum_{j=i+1}^{n} \nu_j \widehat{\boldsymbol{b}}_j$ and $\nu' \in (-\frac{1}{2}, \frac{1}{2}]$. SAMPLE chooses $(x_i, \nu_i) = (x', \nu')$ if $l \operatorname{div} 2^{n-i-1}$ is

even. Else (x_i, ν_i) becomes the also unique $(x' \pm 1, \nu' \pm 1)$ s. t. $\nu_i \in (-1, -\frac{1}{2}] \cup (\frac{1}{2}, 1]$.

Thus, $\{1, \ldots, 2^{u_{\max}}\} \to L(B) : l \mapsto \textsc{Sample}(B, R, l)$ is an enumeration of all points in $L(B)$ subject to (SC) with $u = u_{\max}$. Inspection of Alg. 2 shows the computation of Sample requires $2n$ vector updates and assignments, i. e. $O(n^2)$ arithmetic operations.

3.2 The Best Bound Algorithm

The vectors computed by Sample are likely to be short but they are of course not necessarily shorter than $\boldsymbol{b}_1$. The algorithm BestBound (Alg. 3) yields an estimate how many samples are required in the search space $V_t := \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{2^{-t}}\}$ if we want to guarantee a success probability $\Pr\left[\min\{\|\boldsymbol{v}\|^2 \,|\, \boldsymbol{v} \in V_t\} \leq \gamma \|\boldsymbol{b}_1\|^2\right] \geq 1/2$.

Schnorr [1] gives a formula that implies a bound on t but his formula requires the so called Geometric Series Assumption (GSA). In practice, we usually encounter bases that do not satisfy GSA. That poses two questions Schnorr does not answer: How good (and in which sense) must a basis approximate GSA for the given formula still to be meaningful? How can the formula be evaluated if the GSA coefficient it involves is not evident?

The way BestBound achieves its estimate is explained in detail in Sect. 4. Here we point out that BestBound makes no assumption on the input basis, in particular not GSA. However, if the basis B happens to satisfy GSA then the bound that BestBound computes is at least as sharp as the one given by Schnorr's formula.

4 BestBound

We explain in the following the idea behind BestBound and the numerical solution of the resulting problems.

The Randomness Assumption. Empirically, the coefficients ν_i of the vectors sampled by SR behave like uniform (pseudo-)random variables except for i near to n. We thus can assess the expected values of $\nu_i^2 \|\widehat{\boldsymbol{b}}_i\|^2$ if $n - i$ is not too small. Typically, the last columns of $\widehat{B}$ are much shorter than the leading columns whence the error we incur if we treat all ν_i as uniform random variables is insignificant. That justifies the following assumption:

Algorithm 3 BESTBOUND

Input: $\mathfrak{b} = (\|\widehat{\boldsymbol{b}}_1\|^2, \ldots, \|\widehat{\boldsymbol{b}}_n\|^2)$, base 2 logarithm $u_{\max}$ of maximum number of samples, reduction factor γ

Output: $t_{\max} = \max\{t \in \mathbb{Z} \mid \Pr\left[\min\{\|\boldsymbol{v}\|^2 \leq \gamma\|\widehat{\boldsymbol{b}}_1\|^2 \mid \boldsymbol{v} \in V_t\}\right] \geq 1/2\} \cup \{-\infty\}$

procedure EXPLENGTH(ℓ, k, u, q) /* $\ell = (l_1, \ldots, l_n)$ */
  return $\frac{1}{12}\sum_{i=1}^{k-1} q^{k-i} l_i + \frac{1}{12}\sum_{i=k}^{n-u-1} l_i + \frac{1}{3}\sum_{i=n-u}^{n-1} l_i + l_n$
end procedure

procedure LOGSUCCESSPROBBOUND(ℓ, k, u, γ)
  if EXPLENGTH($\ell, k, u, 1$) $\leq \gamma\|\widehat{\boldsymbol{b}}_1\|^2$ **then**
    return -1
  else if EXPLENGTH($\ell, k, u, 0$) $\geq \gamma\|\widehat{\boldsymbol{b}}_1\|^2$ **then**
    return $-\infty$
  end if
  $q_\gamma \leftarrow$ REGULAFALSI(EXPLENGTH(ℓ, k, u, q) $= \gamma\|\boldsymbol{b}_1\|^2, q \in [0, 1]$)
  return $\lfloor \frac{k(k-1)}{4} \log_2(q_\gamma) - 1 \rfloor$
end procedure

procedure BESTBOUND($\mathfrak{b}, u_{\max}, \gamma$)
  $\ell \leftarrow (\|\widehat{\boldsymbol{b}}_{\sigma(1)}\|^2, \ldots, \|\widehat{\boldsymbol{b}}_{\sigma(n)}\|^2)$ /* permutation σ defined by (1), page 7 */
  return $\max\{$LOGSUCCESSPROBBOUND$(\ell, k, u_{\max}, \gamma) \mid k = 1, \ldots, n - u_{\max}\}$
end procedure

Assumption 1 (Randomness Assumption) *Let $l \in_R \{1, \ldots, 2^u\}$ be a uniform random variable, $1 < u < n$. Then the ν_i, $1 \leq i < n$, defined by* SAMPLE$(B, R, l) = \sum_{i=1} \nu_i \widehat{\boldsymbol{b}}_i$ *are statistically indistinguishable from independent random variables with uniform distribution on the intervals defined by (SC).*

4.1 The Approach Taken by BESTBOUND

BESTBOUND is supposed to return a lower bound for (the $\log_2$ of) the probability that SAMPLE returns a vector shorter than $\sqrt{\gamma}\|\boldsymbol{b}_1\|$. The algorithm is based on the following idea: The sampling of a lattice point $\boldsymbol{v}$ is a random experiment. We consider some event $(\mathrm{S}_{q,k})$ parameterized by $q \in [0, 1]$ and $1 \leq k < n - u_{\max} < n$. The probability of $(\mathrm{S}_{q,k})$ is strictly increasing in q. Let $0 \leq q_\gamma \leq 1$ be maximal s. t. the conditional expected length $E\left[\|\boldsymbol{v}\|^2 \mid (\mathrm{S}_{q,k})\right] \leq \gamma\|\boldsymbol{b}_1\|^2$. Then the success probability is

$$\begin{aligned}\Pr\left[\|\boldsymbol{v}\|^2 \leq \gamma\|\boldsymbol{b}_1\|^2\right] &\geq \Pr\left[\|\boldsymbol{v}\|^2 \leq E\left[\|\boldsymbol{v}\|^2 \mid (\mathrm{S}_{q_\gamma,k})\right] \mid (\mathrm{S}_{q_\gamma,k})\right] \Pr\left[(\mathrm{S}_{q_\gamma,k})\right] \\ &= \frac{1}{2}\Pr\left[(\mathrm{S}_{q_\gamma,k})\right] .\end{aligned}$$

BestBound computes $\max\{\lfloor \log_2 \left(\frac{1}{2}\Pr\left[(\mathrm{S}_{q_\gamma,k})\right]\right)\rfloor \mid k = 1,\ldots,n-u_{\max}\}$. Consequently, if SR samples at least $2^{-\textsc{BestBound}(\mathfrak{b},u_{\max},\gamma)}$ lattice points then the probability to find a sufficiently short vector is at least $1/2$.

The event $(S_{q,k})$. Consider the random experiment $\boldsymbol{v} = \textsc{Sample}(B,R,l) = \sum_{i=1}^{n} \nu_i \widehat{\boldsymbol{b}}_i$, $l \in_R \{1,\ldots,2^{u_{\max}}\}$. $\sigma \in \mathrm{Sym}(\{1,\ldots,n\})$ describes the sorting of the first $n-u_{\max}-1$ elements of $\mathfrak{b}$ in non-increasing order, i. e.

$$\|\widehat{\boldsymbol{b}}_{\sigma(1)}\|^2 \geq \cdots \geq \|\widehat{\boldsymbol{b}}_{\sigma(n-u_{\max}-1)}\|^2 \quad \text{and} \quad \sigma(i) = i \text{ for } i \geq n-u_{\max}. \quad (1)$$

Let $q \in [0,1]$ and $1 \leq k < n-u_{\max}$. $(\mathrm{S}_{q,k})$ denotes the event

$$\nu_{\sigma(i)}^2 \leq \frac{1}{4} q^{k-i} \quad \text{for } i = 1,\ldots,k-1. \qquad (\mathrm{S}_{q,k})$$

The randomness assumption on ν_i yields

$$\Pr\left[(\mathrm{S}_{q,k})\right] = \prod_{i=1}^{k-1} \Pr\left[|\nu_{\sigma(i)}| \leq \frac{1}{2} q^{\frac{k-i}{2}}\right] = q^{\frac{k(k-1)}{4}}.$$

The expected length of $\boldsymbol{v}$. Assume $(\mathrm{S}_{q,k})$. For any uniform random variable $x \in (-t,t]$ the expected value of x^2 is $E\left[x^2\right] = \frac{1}{3}t^2$. The ν_i are independent random variables uniformly distributed on intervals defined by (SC) and $(\mathrm{S}_{q,k})$ whence

$$\begin{aligned} E(\mathfrak{b},k,q) &:= E\left[\|\boldsymbol{v}\|^2 \mid (\mathrm{S}_{q,k})\right] = \sum_{i=1}^{n} E\left[\nu_i^2 \mid (\mathrm{S}_{q,k})\right] \|\widehat{\boldsymbol{b}}_i\|^2 \\ &= \sum_{i=1}^{k-1} q^{k-i} \frac{\|\widehat{\boldsymbol{b}}_{\sigma(i)}\|^2}{12} + \sum_{i=k}^{n-u_{\max}-1} \frac{\|\widehat{\boldsymbol{b}}_{\sigma(i)}\|^2}{12} + \sum_{i=n-u_{\max}}^{n-1} \frac{\|\widehat{\boldsymbol{b}}_i\|^2}{3} + \|\widehat{\boldsymbol{b}}_n\|^2. \end{aligned}$$

4.2 Numerical Solution

Schnorr gives a closed formula for $E(\mathfrak{b},k,q)$. We cannot derive such a formula without the GSA since we do not have a priori knowledge about $\mathfrak{b}$. However, given a basis B we can easily evaluate $E(\mathfrak{b},k,q)$, provided one takes some care to avoid loss of numeric precision. Since all summands are positive it is sufficient to add them in non-decreasing order.

Computation of q_γ. The expected length $E(\mathfrak{b}, k, q)$ is a polynomial in q with non-negative coefficients. Therefore, $f : [0,1] \to \mathbb{R} : q \mapsto E(\mathfrak{b}, k, q) - \gamma\|\boldsymbol{b}_1\|^2$ is a strictly increasing continuous function that has a root if and only if $f(0) \leq 0 \leq f(1)$. The unique root q_γ can be efficiently determined with the textbook Regula Falsi algorithm [11] provided such a root exists.

If $f(1) < 0$ then the (unconditional) mean value $E\left[\|\boldsymbol{v}\|^2\right]$ is already short enough and we have $\Pr\left[\|\boldsymbol{v}\|^2 \leq \gamma\|\boldsymbol{b}_1\|^2\right] \geq \frac{1}{2}$. On the other hand, if $f(0) > 0$ then our approach does not yield a positive lower bound on $\Pr\left[\|\boldsymbol{v}\|^2 \leq \gamma\|\boldsymbol{b}_1\|^2\right]$ for this particular choice of k.

The optimal bound t. BestBound computes the maximum success probability for all k. The computation of $\Pr\left[(\mathrm{S}_{q_\gamma,k})\right]$ is in our experience fast enough that the cost for computing the probability for all $k = 1, \ldots, n - u_{\max} - 1$ is negligible.

5 Experimental Results

All experiments were performed on a Linux system with a 2.4 GHz Pentium 4 processor and 1 GByte RAM. We used a lattice reduction library that is derived from Shoup's NTL [12]. We tested our algorithm with bases in Hermite normal form as proposed by Micciancio [7] for the public keys in his variant of the GGH cryptosystem. They are derived from base vectors uniformly chosen from a cube whence the generated lattices do not have any special structure. The HNF bases were $(0.99, \beta)$-BKZ reduced for various values of β. The resulting bases were input to the Sampling Reduction. We stick here to an example in dimension 180 (Fig. 1) in order to simplify the presentation. Our experiments showed comparable results in other dimensions, though.

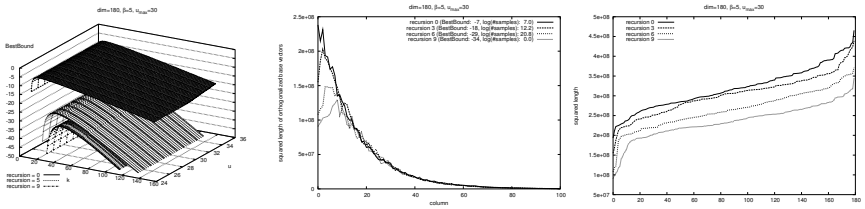

Fig. 1. Sampling Reduction of HNF Bases
Values of BestBound, $\|\widehat{\boldsymbol{b}}_i\|^2$, and $\|\boldsymbol{b}_i\|^2$ (sorted in non decreasing order)

With the sampling parameter $u_{\max} = 30$ the Sampling Reduction improved the squared length of the shortest base vector by a factor < 0.6.

A large part of this improvement is gained in the first iterations. With $\beta = 5$, the Sampling Reduction took 1928 s, of which 71 s were spent on the BKZ updates. With $\beta = 10$, the Sampling reduction ran only for 577 s but here 190 s were spent in the BKZ updates.

It is noticeable that the very first base vectors are much more improved than the remaining base vectors. Most of the time, the effect of the BKZ updates peters out quickly. In particular, $\|\widehat{\boldsymbol{b}}_i\|^2$ does not change significantly beyond base column 20. Since $E\left[\|\boldsymbol{v}\|^2\right]$ does not change that much if only few $\widehat{\boldsymbol{b}}_i$ become smaller it quickly becomes less likely that a sampled vector is shorter than $\boldsymbol{b}_1$. This is also reflected in our estimates of the success probability's logarithm, shown in the leftmost diagram in Fig. 1. The estimates decrease quite rapidly with every recursion.

The value of BestBound actually depends on the choice of $u_{\max}$: If one increments $u_{\max}$ then $E(\mathfrak{b}, k, q)$ grows by $\frac{1}{4}\|\widehat{\boldsymbol{b}}_{n-u_{\max}-1}\|^2$ which means that q_γ and therefore $\Pr\left[(\mathrm{S}_{q_\gamma,k})\right]$ become smaller. But the diagram makes apparent that this effect is negligible.

The experiments exhibit another fact, however: The bound on the success probability is way too pessimistic. Most of the time, the search space size computed by BestBound exceeds the actual number of samples before a sufficiently short vector is found by a factor between 2^9 and 2^{10}. We outline a potential strategy to overcomes this problem in Sect. 8.

6 Globally Improved Bases

SR replaces the first base vector by a significantly shorter vector. The remaining base vectors are much less improved. We address this by two variants of SR. Due to the lack of space we can only sketch the algorithms.

Pool Sampling Reduction. Our pool variant of SR takes advantage of the short vectors sampled before a "short enough" vector is found. In the recursion step the generated pool of short vectors is prepended to the basis. In the final result there is a block of vectors not much longer than $\boldsymbol{b}_1$ and the length of the remaining base vectors is also somewhat shorter than in the result of the plain SR algorithm. The length of $\boldsymbol{b}_1$, however, is not further improved by the pool variant.

Short Projection Reduction. SR decreases $\|\widehat{\boldsymbol{b}}_i\|^2$ for small i only. ShortProjectionSR tries to reduce explicitly selected $\widehat{\boldsymbol{b}}_i$, $i \in T \subseteq \{1, \ldots, n\}$.

ShortProjectionSR computes the success probability with respect to $\mathfrak{b}_t = (\|\widehat{\boldsymbol{b}}_t\|^2, \ldots, \|\widehat{\boldsymbol{b}}_n\|^2)$ for all $t \in T$. Then the algorithm proceeds like

SR except that it compares the length of $\pi_t(\boldsymbol{v})$ against $\widehat{\boldsymbol{b}}_t$, $t \in T$, whereas SR compares $\boldsymbol{v}$ and $\boldsymbol{b}_1$. If $\pi_t(\boldsymbol{v})$ is sufficiently short then $\boldsymbol{v}$ is inserted at column t and the algorithm recurses.

ShortProjectionSR with $T = \{1, \ldots, 10\}$ and $\gamma = 0.99$ is similar to Schnorr's ESHORT algorithm. The goal of ESHORT is to make the reduction continue even if the first ten base vectors violate GSA. In contrast, ShortProjectionSR aims at producing shorter vectors in the middle of the basis and at shaping $\mathfrak{b}$ so the chances of further reductions increase. We will demonstrate the potential of this technique in Sect. 7.

7 Reduction of NTRU Bases

We explain why under plausible assumptions the recommended security parameter $N = 251$ for NTRU [13] offer at most 74 Bit security and that 80 Bit security requires $N \geq 271$. We report how ShortProjectionSR performs when applied to NTRU lattice bases.

NTRU security parameters. Recovering a private key from a public NTRU key h is equivalent to finding short vectors in a certain $2N$-dimensional lattice L_h^{NT} [14]. Therefore, a successful attack against NTRU takes no longer than a reduction of the publicly known basis of L_h^{NT} that recovers a short vector in L_h^{NT}.

In order to assess the impact of sampling reduction on NTRU we need to estimate its runtime. Schnorr concluded that the asymptotic runtime of RSR required to achieve a given approximation of $\lambda_1(L)$ is about the 4th root compared to previous methods [1]. We do not expect that SR gives the speedup Schnorr stated for RSR because it was derived in an idealized setting. But we conjecture the runtime of SR is somewhere in between because SR is similar to RSR. That is, we expect that SR improves the asymptotic runtime of lattice reduction by a factor x in the exponent with $1/4 < x < 1$. We estimate that $x \leq 0.75$.

Based on experiments with NTRU lattices for $N \leq 122$, Hoffstein, Silverman and Whyte extrapolated the runtime T_{BKZ} of successful attacks with BKZ reduction. They found $T_{\mathrm{BKZ}} \geq 10^{A_{\mathrm{BKZ}}N + B_{\mathrm{BKZ}}}$ MIPS-years with $A_{\mathrm{BKZ}} = 0.1095$ and $B_{\mathrm{BKZ}} = -12.6401$. Since SR improves the asymptotic reduction time by the factor x in the exponent we expect the runtime $T_{\mathrm{SR},x} \geq 10^{A_{\mathrm{SR}}(x)N + B_{\mathrm{SR}}(x)}$ with $A_{\mathrm{SR}}(x) = xA_{\mathrm{BKZ}}$ for attacks with sampling reduction. From the experiments described below we know that the runtime of an sampling attack for $N = 109$ is about the same as a BKZ attack. Thus $B_{\mathrm{SR}}(x) \approx -11.9x - 0.7$.

N	$T_{1.0}$	$T_{0.75}$	$T_{0.50}$	$T_{0.25}$
167	4.9×10^5	1.2×10^4	3.1×10^2	7.9×10^0
251	7.7×10^{14}	9.7×10^{10}	1.2×10^7	1.6×10^3
271	1.2×10^{17}	4.3×10^{12}	1.5×10^8	5.5×10^3
300	1.8×10^{20}	1.0×10^{15}	6.0×10^9	3.4×10^4

Table 1. Breaking times in MIPS-years for various N and x

Tab. 1 shows the resulting breaking times for various N and x. We estimated $x \leq 0.75$ for sampling attacks, so for $N = 251$ a sampling attack is expected to require $< 10^{11}$ MIPS-years corresponding to 74 Bit security [15]. Ignoring refined attacks by zero-forcing [16] we find that 80 Bit security requires $N \geq 271$.

Sampling Reduction of L^{NT}. An NTRU instance defines a pair (a, c) of characteristic quantities. The larger these values the longer takes a successful attack with BKZ. The experiments in [17] were done for lattices with $(a, c) \approx (0.535, 1.73)$. We constructed NTRU instances for the same values and BKZ reduced the corresponding lattices with successively incremented parameter β until the private keys were recovered.

Consider an example with $N = 109$. BKZ recovered the actual private key not before $\beta = 10$. But for $\beta \geq 6$ the reduced bases contain vectors that are about 6.5 times as long as the private key. Coppersmith and Shamir [14] have shown that such almost shortest vectors enable an attacker to recover at least partial information about the plaintext whence we consider the reductions with $6 \leq \beta \leq 9$ successful attacks as well.

It is evident from the right diagram in Fig. 2 that a plain Sampling Reduction of the NTRU lattice reduced with $\beta = 5$ must fail: SR tries to find a vector $\boldsymbol{v}$ with $\|\boldsymbol{v}\|^2 < 41000$ but the probability to sample vectors shorter than 100000 is very small.

This is an almost perfect setup for the Short Projection variant. ShortProjectionSR with $T = \{15\}$ returned immediately a vector with $\|\pi_{15}(\boldsymbol{v})\|^2 < 120000$. The subsequent BKZ update with $\beta = 5$ broke the NTRU instance within 341 s. The resulting base vectors were about the same length as after BKZ reduction with $\beta = 6$, i. e. $700 < \|\boldsymbol{b}_i\|^2 < 900$ for many $1 \leq i \leq N$.

Even smaller BKZ parameter are possible: The left diagram in Fig. 2 shows the progress of a Short Projection Reduction with $\beta = 3$. The target set $T = \{t, t+1\}$ was in each recursion step chosen such that t was the first column with $\|\widehat{\boldsymbol{b}}_t\|^2 \geq 42000$. $\|\widehat{\boldsymbol{b}}_{15}\|^2$ was gradually decreased over 32 recur-

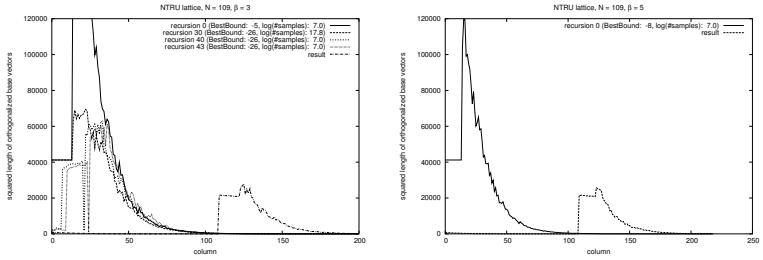

Fig. 2. Reduction of NTRU Lattice ($N = 109$)
$\|\widehat{\boldsymbol{b}}_i\|^2$ during the Short Projection Reduction of an NTRU lattice basis.

sion steps when it suddenly fell from 63670 to 1905. After that t could be incremented in at least every second recursion step. The NTRU instance was broken in the 44th recursion step after 805 s with $\|\boldsymbol{b}_1\|^2 = 812$. Further Sampling Reduction of the subbasis $B' = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{109}]$ improved the attackers key in 107 s to $\|\boldsymbol{b}_1\| = 364$, i. e. to about 3 - 4 times the length of the private key whence Coppersmith's and Shamir's method will reveal much more if not all information about the plaintext. The attack with BKZ only behaved differently: Here the first basis vector did not improve beyond $\|\boldsymbol{b}_1\|^2 = 650$ ($\beta = 9$) before the actual private key was recovered.

Our sampling attacks took about as long as the BKZ reductions. But for larger N the BKZ parameter β required for a successful attack is much larger so the reduction of β by sampling reduction will have significant impact on the runtime.

8 Conclusion and Further Work

We proposed a practical lattice basis reduction by sampling that avoids any dependence on Schnorr's Geometric Series Assumption. It generalizes Schnorr's RSR algorithm but is also well defined for bases where Schnorr's algorithm is not applicable. We demonstrated that the Sampling Reduction can significantly reduce the length of the base vectors. We also proposed two generalizations that further reduce the overall length of the base vectors and that allow the Sampling Reduction to proceed even if jumps in the length of the orthogonalized base vectors disrupt the plain Sampling Reduction.

We observed that our estimates of the success probability are too pessimistic. We plan to test whether it is numerically feasible to calculate the distribution of the length of the sampled vectors directly by convoluting (via FFT) the distributions of the coefficients ν_i. The so derived exact

success probability could open the way for search spaces better tailored to specific classes of lattice bases.

We demonstrated that the Sampling Reduction makes it possible to break NTRU instances with smaller BKZ parameters than before. We put forth arguments that in the light of Sampling Reduction the recommended NTRU parameters [8] offer less security than previously assumed. This result needs to be verified by further experiments in higher dimensions.

References

1. 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
2. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261** (1982) 515–534
3. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology **10** (1997) 233–260
4. Odlyzko, A.M.: The rise and fall of knapsack crytosystems. In: Cryptology and Computational Number Theory. Volume 42 of Proc. Symp. Appl. Math., AMS (1990) 75–88
5. Ajtai, M., Dwork, C.: A public-key cryptosystem with worstcase / average-case equivalence. In: Proceedings of the 29th Annual Symposium on Theory of Computing (STOC), ACM Press (1997) 284–293
6. 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
7. Micciancio, D.: The shortest vector problem is NP-hard to approximate to within some constant. SIAM Journal on Computing **30** (2001) 2008–2035
8. Consortium for Efficient Embedded Security: EESS #1: Implementation aspects of NTRUEncrypt and NTRUSign. `http://www.ceesstandards.org/documents/EESS1_v2.pdf` (2003) Version 2.0.
9. NTRU Cryptosystems, Inc.: Website. `http://www.ntru.com` (2004)
10. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Math. Programming **66** (1994) 181–199
11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. 2nd edn. Cambridge University Press (1992)
12. Shoup, V.: NTL – a library for doing number theory. URL `http://www.shoup.net/ntl/index.html` (2004) Release 5.3.2.
13. 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)
14. Coppersmith, D., Shamir, A.: Lattice attacks on NTRU. In: Advances in Cryptology – Eurocrypt'97. Volume 1233 of LNCS., Springer (1997) 52–61
15. Lenstra, A.K., Verheul, E.R.: Selecting cryptographic key sizes. J. Cryptology **14** (2001) 255–293
16. May, A., Silverman, J.H.: Dimension reduction methods for convolution modular lattices. [18] 110–125

17. Hoffstein, J., Silverman, J.H., Whyte, W.: Estimated breaking times for NTRU lattices. Technical Report 12, version 2, NTRU Cryptosystems (2003) http://www.ntru.com/cryptolab/tech_notes.htm#012.
18. Silverman, J.H., ed.: Cryptography and Lattices. Volume 2146 of LNCS., Springer-Verlag (2001)