

Parallel Symmetric Attack on NTRU using Non-Deterministic Lattice Reduction

Tanya E. Seidel, Daniel Socek, and Michal Sramka

Tanya E. Seidel

*Department of Mathematical Sciences, Florida Atlantic University,
777 Glades Road, Boca Raton, FL 33431*
teseidel@aol.com

Daniel Socek

*Department of Computer Science and Engineering, Florida Atlantic University,
777 Glades Road, Boca Raton, FL 33431*
dsocek@brain.math.fau.edu

Michal Sramka

*Department of Mathematical Sciences, Florida Atlantic University,
777 Glades Road, Boca Raton, FL 33431*
sramka@math.fau.edu

Keywords NTRU, lattice reduction, hill-descending

Abstract

Currently, the most efficient attack on the NTRU public-key cryptosystem, proposed by Coppersmith and Shamir[4], is based on finding a short enough vector in an integral lattice. An NTRU lattice possesses a cyclic automorphism group whose symmetry may be exploited. We have designed methods for reducing bases of NTRU integral lattices based on this symmetry. In addition to these methods, we use hill-descending techniques to combine new and proposed lattice-reduction algorithms. This approach includes deterministic and non-deterministic components which may be efficiently parallelized.

1 Introduction

The NTRU cryptosystem was originally proposed by Hoffstein, Pipher, and Silverman[2]. Since its origin, it has undergone several improvements and has become the fastest public-key encryption scheme available. Consequently, NTRU is suitable for applications where processor and memory requirements are limited, such as Smart-Cards, mobile devices, and embedded technologies. NTRU has been accepted to IEEE P1363 standards and is currently being considered for standard by the Consortium for Efficient Embedded Security (CEES).

The security of NTRU is not necessarily based on the difficulty of reducing the NTRU lattice, but lattice reduction is currently the best known attack. In addition, it is necessary to carefully choose NTRU parameters in order to avoid certain specialized attacks such as using decryption failures to recover the private key[5], among others.

Even though many changes to the NTRU encryption scheme have been made, our attacks are based on the scheme introduced in [2]. The most recently proposed enhancement[6] affects implementation efficiency and may affect security.

2 A Brief Outline of NTRU Algorithm

The NTRU public-key encryption scheme[2] is specified with the following parameters: N, p, q, d_f, d_g, d_r with the standard parameters $p = 3, q = 2^k, N = 1 + 2r$ with r a prime and d_f, d_g, d_r specifying properties of polynomials to be chosen.

- **Key Generation**

Given $\mathcal{R} = \mathbb{Z}[x]/(x^N - 1)$, the ring of truncated polynomials modulo the ideal generated by $x^N - 1$, Bob chooses $f \in \mathcal{R}\{d_f, d_f - 1\}$. That is, Bob chooses a polynomial in the truncated ring with d_f coefficients equal to 1, $d_f - 1$ coefficients equal to -1, and all other coefficients equal to 0. Similarly, Bob chooses $g \in \mathcal{R}\{d_g, d_g\}$. He then computes the multiplicative inverse f_p^{-1} of f in \mathcal{R}_p , the truncated ring of polynomials modulo p , and similarly, the inverse f_q^{-1} of f in \mathcal{R}_q . Finally, Bob computes his public key

$$h \equiv p \cdot f_q^{-1} \cdot g \pmod{q} \quad (2.1)$$

The public information is now

$$\{N, p, q, d_f, d_g, d_r, h\}$$

and the private key is f .

- **Encryption**

Alice selects a random polynomial $r \in \mathcal{R}\{d_r, d_r\}$ and encrypts a message m by

$$e \equiv r \cdot h + m \pmod{q}$$

- **Decryption**

In order to decrypt the received ciphertext e , Bob first computes

$$a \equiv f \cdot e \pmod{q}$$

and then finds

$$b \equiv a \pmod{p}$$

Finally, Bob computes

$$c \equiv f_p^{-1} \cdot b \pmod{p}$$

Now c should be Alice's original message m .

3 Lattice Reduction Attack on NTRU

The following attack was introduced by Coppersmith and Shamir[4]. Let L be the following $2N \times 2N$ matrix

$$L = \left[\begin{array}{c|c} I & O \\ \hline \text{cir}(h) & qI \end{array} \right] \quad (3.1)$$

where I is the $N \times N$ identity matrix, O the $N \times N$ zero matrix, and

$$\text{cir}(h) = \begin{bmatrix} h_0 & h_{N-1} & \dots & h_2 & h_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N-2} & h_{N-3} & \dots & h_0 & h_{N-1} \\ h_{N-1} & h_{N-2} & \dots & h_1 & h_0 \end{bmatrix}$$

is the circulant matrix of the public polynomial $h = h_0 + h_1x + \dots + h_{N-1}x^{N-1}$, defined in (2.1).

The columns of L span a unique integral lattice \mathcal{L} . By the definition of L , it is easy to show that the concatenated vector $(f, pg) \in \mathcal{L}$.

With the standard selection of polynomials f and g , it turns out that (f, pg) is a short vector in \mathcal{L} whose norm can be calculated from the public information. Hence, by reducing the basis of \mathcal{L} , an attacker may obtain the target vector (f, pg) if a vector of equal norm is found.

A basis that contains the target vector is referred to as a **resolution basis**. The well-known LLL algorithm[1] and its various improvements due to Schnorr and others, is currently the fastest method for general basis reduction. Schnorr also introduced a BKZ method[3] which, at the expense of polynomial time, produces a further reduced basis. In this approach, a larger set of vectors is processed simultaneously. The cardinality of this set of vectors is called the **block size**.

4 Symmetry of NTRU Lattice

The NTRU lattice possesses a non-trivial cyclic automorphism group. This symmetry leads to a new cryptanalytic approach.

Let $v = (v_1, v_2, \dots, v_{2N})$. Define **birotation** of v by k positions, denoted $\text{birotate}_k(v)$ as the function $\mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$ defined by $(v_1, v_2, \dots, v_{2N}) \mapsto (v_{1+k \bmod N}, v_{2+k \bmod N}, \dots, v_{N+k \bmod N}, v_{N+(1+k \bmod N)}, \dots, v_{N+(N+k \bmod N)})$.

The following Theorem shows some of the nice properties of such birotations.

Theorem 4.1 *Let \mathcal{L} be an NTRU lattice. Then $v \in \mathcal{L}$ if and only if $\text{birotate}_k(v) \in \mathcal{L}$ for any integer k .*

Proof. Let P_k be the $N \times N$ permutation matrix that performs a cyclic shift by k positions, and let P be the following block matrix

$$P = \left[\begin{array}{c|c} P_k & O \\ \hline O & P_k \end{array} \right]$$

where O is the $N \times N$ zero matrix. It is easy to see that

$$\text{birotate}_k(v) = Pv \tag{4.1}$$

Since $v \in \mathcal{L}$, it can be expressed as a linear combination of columns of L ; i.e., there exists a vector $x = (x_1, \dots, x_{2N})$, with all integer coefficients, such that

$$v = Lx.$$

Multiplying both sides by P yields

$$Pv = PLx = PLP^{-1}Px.$$

Use (4.1) to obtain

$$\text{birotate}_k(v) = PLP^{-1} \cdot \text{birotate}_k(x) \tag{4.2}$$

On the other hand, since

$$P_k \cdot \text{cir}(h) = \text{cir}(h) \cdot P_k$$

the following equality holds

$$\begin{aligned} PLP^{-1} &= \left[\begin{array}{c|c} P_k & O \\ \hline O & P_k \end{array} \right] \left[\begin{array}{c|c} I & O \\ \hline \text{cir}(h) & qI \end{array} \right] \left[\begin{array}{c|c} P_k^{-1} & O \\ \hline O & P_k^{-1} \end{array} \right] = \\ &= \left[\begin{array}{c|c} P_k & O \\ \hline P_k \cdot \text{cir}(h) & qP_k \end{array} \right] \left[\begin{array}{c|c} P_k^{-1} & O \\ \hline O & P_k^{-1} \end{array} \right] = \\ &= \left[\begin{array}{c|c} P_k P_k^{-1} & O \\ \hline P_k \cdot \text{cir}(h) \cdot P_k^{-1} & qP_k P_k^{-1} \end{array} \right] = \left[\begin{array}{c|c} I & O \\ \hline \text{cir}(h) & qI \end{array} \right] = L. \end{aligned}$$

Finally, equation (4.2) states that $\text{birotate}_k(v) = L \cdot \text{birotate}_k(x)$, that is, $\text{birotate}_k(v) \in \mathcal{L}$, since it can be written as a linear combination of columns of L .

For the converse, assume $\text{birotate}_k(v) \in \mathcal{L}$ and observe that

$$\text{birotate}_{N-k}(\text{birotate}_k(v)) = \text{birotate}_N(v) = v,$$

and so by the direct statement of this proposition it follows that $v \in \mathcal{L}$.

■

An elementary result is summarized in the following remark.

Remark 4.2 *Let \mathcal{L} be any integral lattice, and suppose that vectors $V = \{v_1, v_2, \dots, v_n\}$ form a basis for \mathcal{L} . If $w \in \mathcal{L}$ then, for any i , $1 \leq i \leq n$, the set of vectors $V' = \{v_1, v_2, \dots, v_{i-1}, w, v_{i+1}, \dots, v_n\}$ forms a spanning set for a sublattice \mathcal{L}' of \mathcal{L} . If vectors in V' are linearly dependent over \mathbb{Z} , then $\det V' = 0$ and $\dim \mathcal{L}' < \dim \mathcal{L}$. When the vectors in V' are linearly independent over \mathbb{Z} , then the sublattice \mathcal{L}' is of the same dimension as \mathcal{L} and V' forms a basis of \mathcal{L}' .*

The implementation of the previous Theorem is a core component of the proposed NTRU lattice reduction technique. That is, the vector of largest norm in the basis B can be replaced by a birotation of any vector of smaller norm. If the new basis spans the same lattice, then this basis is reduced relative to B . The algorithm in Table 1 illustrates this process.

Input: $2N \times 2N$ matrix L (a basis of an NTRU lattice), and $m, n \in \mathbb{Z}$, $1 \leq m < n \leq 2N$

Output: $2N \times 2N$ matrix L' with weight less than the weight of L

(* BIROT)

$L'' \leftarrow L$

$i \leftarrow 1$

$a \leftarrow \det(L)$

START:

n^{th} column of $L'' \leftarrow \text{birotate}_i(m^{\text{th}}$ column of L'')

$b \leftarrow \det(L'')$

if $\frac{b}{a} = \pm 1$ goto END

$i \leftarrow i + 1$

if $i < N$ goto START

$L' \leftarrow L$

TERMINATE

END:

$L' \leftarrow L''$

TERMINATE

Table 1: BIROT Algorithm

5 Hill-Descending Approach

It is a known fact that the LLL algorithm is sensitive to the order of the basis vectors. A systematic selection of permutations of the basis vectors, along

with the birotation reduction are the building blocks of the hill descending approach.

There are infinitely many bases of a lattice $\mathcal{L} \in \mathbb{R}^n$. However, there is no basis which is obviously better than the others. A basis B' is said to be **reduced** relative to a basis B , if $\text{wt}(B') \leq \text{wt}(B)$. The hill-descending approach implements a walk $B_0 \rightarrow B_1 \rightarrow \dots \rightarrow B_k$ in the space of bases of the lattice \mathcal{L} , where $\text{wt}(B_i) \leq \text{wt}(B_{i+1})$. Here, B_0 is the basis L defined in (3.1), and success is achieved if B_k is a resolution basis.

5.1 Ordering of Basis Vectors

Let $B = \{b_0, \dots, b_{2N}\}$ be a basis for NTRU lattice \mathcal{L} , and let $\pi \in S_{2N}$ be the identity permutation on $2N$ letters.

Define **distance** of two permutations $\alpha, \beta \in S_{2N}$ to be $d(\alpha, \beta) = k$, for some integer $0 \leq k \leq 2N$, if the two permutations differ at exactly k positions. Let

$$B(\alpha, k) = \{ \beta \in S_{2N} \mid d(\alpha, \beta) = k \}$$

denote the family of all permutations of distance k from a fixed permutation α . It follows that

$$|B(\alpha, k)| = \binom{n}{k} D_k$$

where D_k is number of derangements on k letters. It is easy to see that $B(\alpha, 0) = \{\alpha\}$ and $|B(\alpha, 1)| = 0$.

In the hill-descending algorithm, permutations from $B(\pi, k)$, for various values k are randomly selected. Such permutations are then applied to the ordering of basis vectors of B . In practice, a basis B is represented in a matrix form, and permuting the order of basis vectors means permuting the columns of the matrix.

5.2 The Algorithm

This section presents a description of the Las Vegas type method for resolving NTRU lattices. The algorithm is based on the combination of BKZ-LLL and BIROT primitives, and it can be implemented in parallel.

The algorithm requires input parameters L and τ , where L is an NTRU lattice basis and τ is the norm value of the target vector (f, pg) . In the first stage a BKZ-LLL with blocksize $s = 2$ is applied to L to obtain an initial reduction B . The blocksize 2 guarantees that the execution time will be polynomial. Next, basis B undergoes a loop of parallel M processes. At each PU_i (Processing Unit i) the columns of B are permuted according to a random permutation α which is of distance k from the identity permutation. Such permuted basis is supplied to local BKZ-LLL primitive, resulting in basis B'_i . At the main PU, the algorithm then examines the bases B'_i and selects the minimal weight basis B_{min} . If the basis B_{min} is reduced relative

to B_i , the algorithm loops back to the parallel stage, setting the input basis to B to B_{min} . In the case when $wt(B_{min}) > wt(B)$, the distance k is incremented by 1 before looping back to parallel stage. However, when k reaches the maximum value of $2N + 1$, BIROT routine is performed on B in order to escape the local minimum of hill-descending approach. Following the BIROT, the algorithm resets k value back to 2. In the case BIROT does not result in the further reduction, blocksize s is increased by 1.

The algorithm is expected to run until it produces resolved basis of L . It is non-deterministic in nature (Las Vegas type), and in some instances it does not produce desired results.

The hill-descending (Las Vegas type) algorithm is in Table 2 and in Table 3, and its schematic diagram is shown in Figure 1.

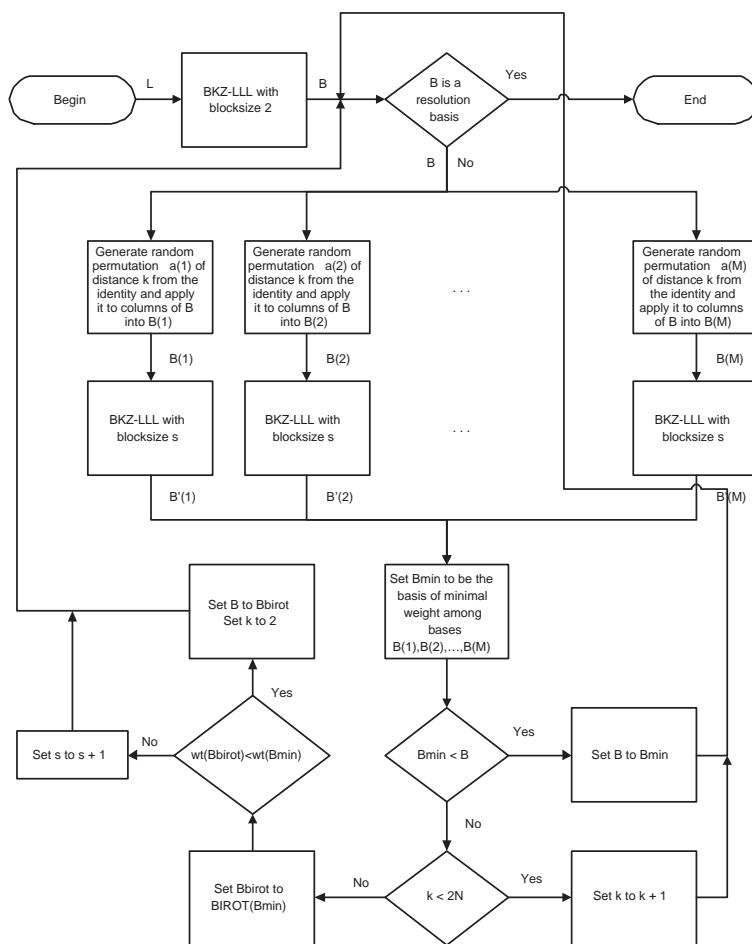


Figure 1: Scheme for the parallel hill-descending algorithm

Input: L an NTRU lattice basis, τ the norm of target vector
Output: resolution basis of L

(* MASTER)

$s \leftarrow 2$ (blocksize)
 $k \leftarrow 2$ (distance)
 $B \leftarrow L$
START:
for each processor j , $1 \leq j \leq M$, perform (* SLAVE j)
select B_{min} such that $\text{wt}(B_{min}) = \min\{\text{wt}(B_1), \dots, \text{wt}(B_M)\}$
if $v \in B_{min}$ such that $\|v\| = \tau$ then
 TERMINATE
if $\text{wt}(B_{min}) < \text{wt}(B)$ then
 $B \leftarrow B_{min}$
 goto START
 $k \leftarrow k + 1$
if $k \leq 2N$ then
 goto START
 $B_{birot} \leftarrow \text{BIROT}(B_{min})$
if $\text{wt}(B_{birot}) < \text{wt}(B_{min})$ then
 $B \leftarrow B_{birot}$
 $k \leftarrow 2$
 goto START
 $s \leftarrow s + 1$
goto START

Table 2: Parallel hill-descending algorithm - MASTER

6 Conclusion

A method for exploiting the symmetry of the NTRU lattice was introduced with the replacement of large vectors in the basis with smaller norm, birotated vectors. This method alone led to reduced bases, but was significantly improved through parallel processing. By using existing well-known lattice reduction techniques, in conjunction with the birotation method, the hill-descending approach introduces a walk through the space of bases which ultimately leads to faster resolution when compared to existing lattice reduction methods. Furthermore, performance is enhanced through the parallelization of critical components, as introduced in the proposed algorithm. While experimental results support the effectiveness of this hill-descending approach, its non-deterministic nature implies variable performance and does not guarantee resolution.

Input: B_j a lattice basis, distance k , block size s

Output: reduced basis of B_j

(* SLAVE j)

randomly select $\alpha \in B(\pi, k)$ where $\pi \in S_{2N}$ is the identity permutation

apply α to the order of vectors in basis B_j

$B'_j \leftarrow \text{BKZ} - \text{LLL}_s(B_j)$

return B'_j

Table 3: Parallel hill-descending algorithm - SLAVE

Acknowledgements

We would like to thank the organizers of the 3rd Pythagorean Conference, and specifically, Spyros Magliveras for his unwavering support and guidance.

References

- [1] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring Polynomials with Rational Coefficients*, *Mathematische Annalen* **261** (1982), pp 515-534.
- [2] J. Hoffstein, J. Pipher, J.H. Silverman, *NTRU: A Ring-Based Public Key Cryptosystem*, Preprint (1998).
- [3] C.P. Schnorr *Block Korkin-Zolotarev Bases and Successive Minima*, Technical Report TR-92-063 (1992).
- [4] D. Coppersmith, A. Shamir, *Lattice attacks on NTRU*, *Advances in Cryptology - EUROCRYPT '97*, Walter Fumy (Ed.), Springer LNCS volume **1233** (1997), pp 52-61.
- [5] N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, W. Whyte, *The Impact of Decryption Failures on the Security of NTRU Encryption*, in *Proc. Crypto 2003*, Santa Barbara, USA, (2003)
- [6] J. Hoffstein, J. Silverman, *Optimizations for NTRU*, in *Proceedings of Public Key Cryptography and Computational Number Theory*, de Gruyter, Warsaw, September 2000