Lattice Attacks on NTRU

Don Coppersmith* Adi Shamir**

Abstract. NTRU is a new public key cryptosystem proposed at Crypto 96 by Hoffstein, Pipher and Silverman from the Mathematics department of Brown University. It attracted considerable attention, and is being advertised over the Internet by NTRU Cryptosystems. Its security is based on the difficulty of analyzing the result of polynomial arithmetic modulo two unrelated moduli, and its correctness is based on clustering properties of the sums of random variables. In this paper, we apply new lattice basis reduction techniques to cryptanalyze the scheme, to discover either the original secret key, or an alternative secret key which is equally useful in decoding the ciphertexts.

1 Introduction

NTRU [1] was proposed at the rump session of Crypto 96, as a fast public-key encryption system. The authors explored several potential attacks against the scheme, but concluded that they are extremely unlikely to succeed. In particular, they considered the standard lattice-based attack and showed that the attackers could not expect to find the secret key by computing the shortest vector in this lattice with the LLL [3] algorithm, since the secret key was surrounded by a "cloud" of exponentially many unrelated lattice vectors.

In this paper we present another lattice-based attack, which should either find the original secret key $\mathbf{f}$ or an alternative key $\mathbf{f}'$ which can be used in place of $\mathbf{f}$ to decrypt ciphertexts with only slightly higher computational complexity. We construct a lattice L, each of whose elements corresponds to a potential decrypting key $\mathbf{f}'$; the effectiveness of $\mathbf{f}'$ for decrypting is directly related to the length of the corresponding lattice element. If we find any vector $\mathbf{f}'$ as short as $\mathbf{f}$, we can decrypt easily. If, instead, we find several vectors $\mathbf{f}'^{(i)}$, each being 2 or 3 times the length of $\mathbf{f}$, then we can obtain partial decryptions from each potential key $\mathbf{f}'^{(i)}$ and piece them together to form a total decryption.

The paper is organized as follows. Section 2 gives some notation, and introduces a norm which will be useful to our analysis. In Section 3 we sketch the NTRU cryptographic system. In Section 4 we describe the lattice L. Section 5 relates the probability of success to the lengths of the recovered short vectors in the lattice.

* IBM Research, Yorktown Heights, NY, USA; copper@watson.ibm.com

** Dept. Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel; shamir@wisdom.weizmann.ac.il

W. Fumy (Ed.): Advances in Cryptology - EUROCRYPT '97, LNCS 1233, pp. 52-61, 1997.

2 Notation

We denote the integers by $\mathbf{Z}$ and the integers modulo q by $\mathbf{Z}_q$. N is a positive integer. We will identify the vector space $\mathbf{Z}^N$ (respectively $\mathbf{Z}_q^N$) with the ring of polynomials $\mathbf{Z}[X]/(X^N - 1)$ (resp. $\mathbf{Z}_q[X]/(X^N - 1)$), by

$$\mathbf{f} = (f_0, f_1, \ldots, f_{N-1})^T = \sum f_i X^i.$$

A boldface letter $\mathbf{f}$ represents a vector. The convolution of two vectors is given by $\mathbf{f} * \mathbf{g}$ where

$$(\mathbf{f} * \mathbf{g})_k = \sum_{i+j=k \pmod N} f_i g_j;$$

this is the ordinary polynomial product in $\mathbf{Z}_q[X]/(X^N - 1)$, and is both commutative and associative. The vector of all 1's is denoted by $\mathbf{1}$. The matrix of all 1's is J, and the identity matrix is I. The symbol p_q^{-1} denotes a multiplicative inverse of p modulo q.

2.1 Approximations to a norm

For $\mathbf{x} \in \mathbf{Z}^N$ define

$$\begin{aligned} \bar{\mathbf{x}} &= \tfrac{1}{N} \textstyle\sum_{i=0}^{N-1} x_i \\ |\mathbf{x}|_\perp &= \left(\textstyle\sum_{i=0}^{N-1} (x_i - \bar{\mathbf{x}})^2 \right)^{1/2} \end{aligned}$$

So $|\mathbf{x}|_\perp$ is the standard deviation of the entries of $\mathbf{x}$, scaled by $\sqrt{N}$. This norm is invariant under the operation of adding $t\mathbf{1}$ to $\mathbf{x}$, that is, adding t to each entry x_i. It is the L^2 norm of the projection of $\mathbf{x}$ orthogonal to the vector $\mathbf{1}$, hence the $\perp$ symbol.

In some circumstances we will use the *approximation*

$$|\mathbf{x} * \mathbf{y}|_\perp \approx |\mathbf{x}|_\perp \, |\mathbf{y}|_\perp \, . \tag{1}$$

Indeed, letting $x_i = \bar{\mathbf{x}} + w_i$ and $y_j = \bar{\mathbf{y}} + z_j$ we find

$$\begin{aligned} |\mathbf{x} * \mathbf{y}|_\perp^2 &= \textstyle\sum_k [(\mathbf{x} * \mathbf{y})_k - \overline{\mathbf{x} * \mathbf{y}}]^2 \\ &= \textstyle\sum_k (\mathbf{w} * \mathbf{z})_k^2 \\ &= \textstyle\sum_k \left(\sum_i w_i z_{k-i}\right) \left(\sum_j w_j z_{k-j}\right), \end{aligned}$$

with indices being considered modulo N. For each product $w_i z_{k-i} w_j z_{k-j}$ counted in the sum, the difference $j - i$ between the $\mathbf{w}$-indices is the same as the difference $(k - i) - (k - j)$ between the $\mathbf{z}$-indices. Letting $d = i - j$ denote this common difference, and setting $\ell = k - j$, we rearrange the sum as:

$$\begin{aligned} |\mathbf{x} * \mathbf{y}|_\perp^2 &= \textstyle\sum_d \left(\sum_i w_i w_{i+d}\right) \left(\sum_\ell z_\ell z_{\ell+d}\right) \\ &= \left(\textstyle\sum_i w_i^2\right) \left(\sum_\ell z_\ell^2\right) + \sum_{d \neq 0} \left(\sum_i w_i w_{i+d}\right) \left(\sum_\ell z_\ell z_{\ell+d}\right) \\ &= |\mathbf{x}|_\perp^2 \, |\mathbf{y}|_\perp^2 + \textstyle\sum_{d \neq 0} \left(\sum_i w_i w_{i+d}\right) \left(\sum_\ell z_\ell z_{\ell+d}\right) \end{aligned}$$

Now let us assume that $\mathbf{w}$ and $\mathbf{z}$ behave like random vectors. For each of the $N-1$ terms corresponding to nonzero d, the autocorrelation coefficient $\sum_i w_i w_{i+d}$ should be smaller than the corresponding sum with $d = 0$, namely $\sum_i w_i^2$, by a factor of about $1/\sqrt{N}$, and similarly with the autocorrelation coefficient $\sum_\ell z_\ell z_{\ell+d}$, so that the product should be smaller by a factor of $1/N$. Further, these terms come in with random sign, so that some cancellation should occur. So, in the random case, we can assume that the second sum (over nonzero values of d) is much smaller than the first term, corresponding to $d = 0$. This leads us to the approximation (1):

$$|\mathbf{x} * \mathbf{y}|_\perp^2 = |\mathbf{x}|_\perp^2 \, |\mathbf{y}|_\perp^2 + \text{smaller terms}$$

$$|\mathbf{x} * \mathbf{y}|_\perp = |\mathbf{x}|_\perp \, |\mathbf{y}|_\perp + \text{smaller terms}$$

3 The NTRU system

We sketch the NTRU system, as developed in [1]. We give sample parameters, based on the authors' original recommendations, to aid the reader's intuition, but with the caution that these parameters can be modified in future versions of the NTRU system.

Public parameters include three positive integers, (N, p, q), with p and q relatively prime. For example we might have $N = 167$, $p = 15$, $q = 1024$. Part of the public key is a vector $\mathbf{h} \in \mathbf{Z}_q^N$. The space of allowable plaintext messages $\mathbf{m}$ is $S_{\mathbf{m}} = \{0, 1, \ldots, p-1\}^N$.

There are additionally spaces

$$S_\phi, S_{\mathbf{f}}, S_{\mathbf{g}} \subseteq \mathbf{Z}_q^N$$

of allowable values of vectors ϕ, $\mathbf{f}$ and $\mathbf{g}$, to be described in the next few paragraphs. For example, we might have each of $S_\phi = S_{\mathbf{f}} = S_{\mathbf{g}}$ being the collection of all $\binom{N}{d}$ N-vectors with $d = 71$ entries of 1 and $N - d = 96$ entries of 0.

The private key contains vectors $\mathbf{f} \in S_{\mathbf{f}}$ and $\mathbf{g} \in S_{\mathbf{g}}$ related to the public key $\mathbf{h}$, and integers s, t, t_1, t_2 which need not be kept secret. The values $\mathbf{f}$ and $\mathbf{g}$ satisfy

$$p\mathbf{g} = \mathbf{f} * \mathbf{h} \pmod{q}. \tag{2}$$

The private key also includes a vector $\mathbf{f}_p^{-1}$, calculated from $\mathbf{f}$, satisfying

$$\mathbf{f} * \mathbf{f}_p^{-1} = (1, 0, 0, \ldots, 0)^T \pmod{p}.$$

This product corresponds to the polynomial 1.

Encryption: To encrypt the plaintext $\mathbf{m}$, the encryptor randomly selects $\phi \in S_\phi$ and computes the ciphertext

$$\mathbf{e} = \phi * \mathbf{h} + \mathbf{m} \pmod{q}.$$

A different random choice of ϕ is made for each plaintext $\mathbf{m}$.

Decryption: The decryptor computes

$$\mathbf{a} = \mathbf{f} * \mathbf{e} = (\mathbf{f} * \mathbf{h}) * \boldsymbol{\phi} + \mathbf{f} * \mathbf{m} \pmod{q},$$

and adjusts the entries by

$$b_k = a_k + t - \begin{cases} 0 & \text{if } a_k < s \\ q & \text{if } a_k \geq s. \end{cases}$$

Notice that

$$(\mathbf{f} * \mathbf{h}) * \boldsymbol{\phi} = p\mathbf{g} * \boldsymbol{\phi} \pmod{q}.$$

Parameters are chosen so that both $t_1\mathbf{1} + p\mathbf{g} * \boldsymbol{\phi}$ and $t_2\mathbf{1} + \mathbf{f} * \mathbf{m}$ are "small enough": the entries of the non-modular expression

$$\mathbf{b} = t\mathbf{1} + p\mathbf{g} * \boldsymbol{\phi} + \mathbf{f} * \mathbf{m}$$

are guaranteed to lie between $-q/2$ and $q/2$ most of the time. If in fact all entries lie in that range, the decryptor can switch from computation modulo q to computation modulo p, and calculate

$$\mathbf{b} * \mathbf{f}_p^{-1} \pmod{p}.$$

This removes dependence on the unknown $\boldsymbol{\phi}$, and recovers $\mathbf{m}$.

We estimate the bound on the elements of $\mathbf{b}$ which still make this computation possible. Using the approximation (1) we can say that

$$|t_1\mathbf{1} + p\mathbf{g} * \boldsymbol{\phi}|_\perp \approx p\,|\mathbf{g}|_\perp\,|\boldsymbol{\phi}|_\perp$$

where $|\boldsymbol{\phi}|_\perp$ is the norm of a typical element of S_ϕ. (We can arrange things so that all such $\boldsymbol{\phi}$ have the same norm.) Similarly

$$|t_2\mathbf{1} + \mathbf{f} * \mathbf{m}|_\perp \approx |\mathbf{f}|_\perp\,|\mathbf{m}|_\perp\,.$$

Making the second assumption that the two vectors $t_1\mathbf{1}+p\mathbf{g}*\boldsymbol{\phi}$ and $t_2\mathbf{1}+\mathbf{f}*\mathbf{m}$ are nearly orthogonal, we would obtain

$$\begin{aligned} |\mathbf{b}|_\perp^2 &= |t\mathbf{1} + p\mathbf{g} * \boldsymbol{\phi} + \mathbf{f} * \mathbf{m}|_\perp^2 \approx |t_1\mathbf{1} + p\mathbf{g} * \boldsymbol{\phi}|_\perp^2 + |t_2\mathbf{1} + \mathbf{f} * \mathbf{m}|_\perp^2 \\ &\approx p^2\,|\mathbf{g}|_\perp^2\,|\boldsymbol{\phi}|_\perp^2 + |\mathbf{f}|_\perp^2\,|\mathbf{m}|_\perp^2\,, \end{aligned}$$

which we choose to write as

$$|\mathbf{b}|_\perp^2 \approx \left(p^2\,|\boldsymbol{\phi}|_\perp^2\right)|\mathbf{g}|_\perp^2 + \left(|\mathbf{m}|_\perp^2\right)|\mathbf{f}|_\perp^2\,. \tag{3}$$

Make the third assumption that the entries of $\mathbf{b}$ are *normally distributed*, with mean near 0 (this governed our choice of t) and standard deviation $\sigma \approx |\mathbf{b}|_\perp/\sqrt{N}$. The decoding procedure will fail if any of the N entries b_i exceeds $q/2$ in absolute value.

In the table below, we see the effect of letting $q/2$ be a reasonable multiple (3,4,5 or 6) of the standard deviation σ. The second column gives the probability that an individual term $|b_i|$ will exceed $q/2$ (and hence be misinterpreted), and the third column gives the probability that at least one of the $N = 167$ terms $|b_i|$ exceeds $q/2$ (and hence the decryption is incorrect).

	Individual	Failure among
	failure	167 entries
$(q/2)/\sigma$	$\rho = \text{Prob}\{\lvert b_i\rvert > q/2\}$	$1-(1-\rho)^N$
3	2.70×10^{-3}	3.63×10^{-1}
4	6.33×10^{-5}	1.05×10^{-2}
5	5.73×10^{-7}	9.57×10^{-5}
6	1.97×10^{-9}	3.30×10^{-7}

So if $q/2 = 5\sigma$ (that is, $\sigma = q/10$) the procedure will correctly decode most messages. We would want to arrange parameters so that $\sigma < q/10$, and a smaller value of σ would ensure higher reliability.

Remark: We are essentially using an estimate on the L^2 norm of $\mathbf{b}$ to produce an estimate of its L^∞ norm; the L^2 bound is relatively easy to estimate, but the L^∞ bound is what is required for error-free decoding.

4 The lattice

We have seen that reliability of decoding is directly related to the ratio of $\sigma \approx |\mathbf{b}|_\perp / \sqrt{N}$ to q. In turn, equation (3) gives an estimate of $|\mathbf{b}|_\perp$ in terms of $|\mathbf{f}|_\perp$ and $|\mathbf{g}|_\perp$, where $p\mathbf{g} = \mathbf{f} * \mathbf{h} \pmod q$.

Let us consider an alternate N-vector $\mathbf{f}'$ which the cryptanalyst can use in place of the correct value $\mathbf{f}$. Calculate from equation (2) a value $\mathbf{g}'$, and from equation (3) an estimate of $|\mathbf{b}'|_\perp$. If this value $|\mathbf{b}'|_\perp$ is comparable to $|\mathbf{b}|_\perp$ (smaller or not much larger), then the cryptanalyst will be able to mimic the legitimate decoder, using $\mathbf{f}'$ and $\mathbf{g}'$, to decode the message.

Thus we find the system of equations

$$p\mathbf{g}' = \mathbf{f}' * \mathbf{h} \pmod q \tag{4}$$

$$|\mathbf{b}'|_\perp^2 \approx \left(p^2 |\phi|_\perp^2\right) |\mathbf{g}'|_\perp^2 + \left(|\mathbf{m}|_\perp^2\right) |\mathbf{f}'|_\perp^2 . \tag{5}$$

Consider $|\phi|_\perp$ and $|\mathbf{m}|_\perp$ to be held constant at their "typical" values. Setting

$$\lambda = \frac{|\mathbf{m}|_\perp}{p\,|\phi|_\perp},$$

we are left with

$$\sigma'^2 = \frac{|\mathbf{b}'|_\perp^2}{N} \approx \left(\frac{p^2\,|\phi|_\perp^2}{N}\right) \left(|\mathbf{g}'|_\perp^2 + \lambda^2\,|\mathbf{f}'|_\perp^2\right)$$

It is a simple matter to build a lattice L, whose elements correspond to choices of $\mathbf{f}'$ and corresponding $\mathbf{g}'$, and with the squared norm of the elements being

$$|\mathbf{g}'|_\perp^2 + \lambda^2\,|\mathbf{f}'|_\perp^2 .$$

Start with a $2n \times 2n$ matrix L':

$$L' = \begin{bmatrix} \lambda I & 0 \\ H & qI \end{bmatrix}.$$

Here I is the $N \times N$ identity matrix, and H is the circulant matrix whose columns are circularly shifted versions of the vector $\mathbf{h}p_q^{-1} \pmod q$; recall p_q^{-1} is an integer satisfying $p_q^{-1}p = 1 \pmod q$.

A vector in the column span of L' will be of the form

$$\mathbf{v}'_{\mathbf{f}',\mathbf{x}} = L' \begin{bmatrix} \mathbf{f}' \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \lambda \mathbf{f}' \\ \mathbf{g}' \end{bmatrix},$$

where $\mathbf{g}'$ satisfies $p\mathbf{g}' = \mathbf{f}' * \mathbf{h} \pmod q$, and $\mathbf{x}$ is an arbitrary integer vector representing multiples of q.

The presence of H in the lower left of L' insures the relation $\mathbf{g}' = p_q^{-1}\mathbf{h} * \mathbf{f}'$ $(\text{mod } q)$, and the block qI serves to perform the reduction modulo q.

The vectors $\lambda\mathbf{f}'$ and $\mathbf{g}'$ will generally have nonzero mean, but we are interested in the orthogonal norms $|\mathbf{f}'|_\perp$ and $|\mathbf{g}'|_\perp$. To this end, subtract from each column vector $\mathbf{v}$ in the top half of L' the constant vector $(\bar{\mathbf{v}})\mathbf{1}$ so that the result has zero mean; similarly each vector $\mathbf{w}$ in the bottom half of L' is replaced by $\mathbf{w} - (\bar{\mathbf{w}})\mathbf{1}$. Our new matrix L is then

$$L = \begin{bmatrix} \lambda I - (\lambda/N)J & 0 \\ H - \alpha J & qI - (q/N)J \end{bmatrix},$$

where J is the matrix of all 1's, and α is a suitably chosen scalar.

Remark: L has only $2N - 2$ independent vectors, because

$$L \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} = L \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \mathbf{0}.$$

Now a typical vector is

$$\mathbf{v}_{\mathbf{f}',\mathbf{x}} = L \begin{bmatrix} \mathbf{f}' \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \lambda(\mathbf{f}' - (\bar{\mathbf{f}'})\mathbf{1}) \\ \mathbf{g}' - (\bar{\mathbf{g}'})\mathbf{1} \end{bmatrix},$$

and the square of its L^2 norm is

$$\begin{aligned} |\mathbf{v}_{\mathbf{f}',\mathbf{x}}|^2 &= \lambda^2 |\mathbf{f}'|_\perp^2 + |\mathbf{g}'|_\perp^2 = \left(\tfrac{1}{p^2 |\phi|_\perp^2}\right) \left[|\mathbf{m}|_\perp^2 |\mathbf{f}'|_\perp^2 + p^2 |\phi|_\perp^2 |\mathbf{g}'|_\perp^2 \right] \\ &= \left(\tfrac{1}{p^2 |\phi|_\perp^2}\right) |\mathbf{b}'|^2 \end{aligned}$$

Thus the norm of the lattice element $\mathbf{v}_{\mathbf{f}',\mathbf{x}}$ is directly related to the suitability of $\mathbf{f}'$ as a decrypting key.

Remark: We also need $\mathbf{f}'$ to be invertible modulo p, so that ${\mathbf{f}'_p}^{-1}$ can be used in the decrypting process. This seems to be a weak requirement.

For a given vector $\mathbf{f}'$, select $\mathbf{x}$ to minimize this norm, and define

$$n_{\mathbf{f}'} = (p\,|\phi|_\perp) \min_{\mathbf{x}} |\mathbf{v}_{\mathbf{f}',\mathbf{x}}| = |\mathbf{b}'|.$$

5 Lengths of suitable vectors

We have seen that the correct key $\mathbf{f}$ should have

$$n_{\mathbf{f}} < q/10$$

in order to insure that messages are decoded correctly at least 0.9999 of the time.

If the lattice basis reduction finds a vector $\mathbf{f}'$ with, say, $n_{\mathbf{f}} = q/4$, then the cryptanalyst can still gain much useful information. The entries b'_k of the recovered vector $\mathbf{b}$ are likely to be contained in the interval

$$[-3\sigma, +3\sigma] = [-3q/4, 3q/4],$$

since there are only 167 entries and the probability of any given entry lying outside the 3σ interval is about 0.0026. Any entry b'_k in the interval $[q/4, 3q/4]$ (mod q) is unreliable, because it could represent either b'_k or $b'_k - q$ and still lie within the range $[-3\sigma, 3\sigma]$. But entries b'_k in the intervals $[0, q/4) \cup (3q/4, q)$ (mod q) are reliable; one can assume that they represent integers in the range $(-q/4, q/4)$ with no aliasing. We expect a fraction 0.68 of all b'_k to lie in this reliable range. Each represents knowledge of a linear relation among the message components m_i (mod p), namely

$$b'_k = \sum_i m_i f'_{k-i} \pmod{p}.$$

If we find two such vectors $\mathbf{f}'^{(1)}$ and $\mathbf{f}'^{(2)}$, each yielding about $0.68N$ linear relations (modulo p) among the N entries m_i, then we can solve the resulting system of linear equations to recover the message $\mathbf{m}$.

If the recovered vectors $\mathbf{f}'$ are somewhat longer, say

$$n_{\mathbf{f}'^{(i)}} \approx 4 \times n_{\mathbf{f}}$$

then we may have to work with faulty partial information: a few of the estimate integers b'_k might be incorrect, leading to a few incorrect linear equations among a collection of mostly correct ones. Then we will have to resort to techniques from error-correcting codes to discover the incorrect equations among the correct ones.

So our success depends on the success of lattice basis reduction methods in finding relatively short vectors in the lattice. If we find a vector as short as $\mathbf{f}$:

$$n_{\mathbf{f}'} \leq n_{\mathbf{f}}$$

then clearly we can use $\mathbf{f}'$ as a decrypting key. If we find two vectors not much longer than $\mathbf{f}$:

$$n_{\mathbf{f}'^{(1)}} = n_{\mathbf{f}'^{(2)}} \leq 2.5 \times n_{\mathbf{f}}$$

then each will give us partial information, and we can combine this information via linear algebra to recover $\mathbf{m}$. If we find several vectors somewhat longer yet,

$$n_{\mathbf{f}'^{(i)}} \approx 4 \times n_{\mathbf{f}}$$

then we still have a chance, if error-correcting techniques can be applied.

The Lovasz lattice basis reduction methods [3] are only guaranteed to find a vector whose length satisfies

$$n_{\mathbf{f}'} < 2^{N/2} n_{\mathbf{f}}$$

which is clearly insufficient. Schnorr [4], [5] has improved the original methods by using block techniques; he can find shorter vectors, at a higher computational price, than LLL. But it is still not guaranteed to find vectors as short as

$$n_{\mathbf{f}'} \approx 4 n_{\mathbf{f}}.$$

To summarize: if there are many vectors $\mathbf{f}'$ with $n_{\mathbf{f}'} \leq n_{\mathbf{f}}$ then we are likely to stumble across one and be able to decrypt. If $\mathbf{f}$ is much shorter than all other vectors, then we are likely to find $\mathbf{f}$. The only hope for the scheme to remain secure is for many vectors to satisfy, say, $n_{\mathbf{f}'} = 10 \times n_{\mathbf{f}}$ and hope that the lattice basis reduction methods fail to find $\mathbf{f}$ among the sea of $\mathbf{f}'$. With any improvements in the technology of lattice basis reduction, this temporary security would vanish.

6 Other comments

The lattice used in our main attack contains linear combinations of the columns of the circulant matrix H and appropriate multiples of the identity matrix I. An alternative lattice attack is to consider the dual lattice which characterizes all the integral solutions of the following homogeneous equation $H * \mathbf{f} = p\mathbf{g} + q\mathbf{k}$, where $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{k}$ are three vectors with integral unknowns, and p, q are the two moduli. This lattice is closely related to that described in Section 4, except for the difference between $\mathbf{x}$ and $|\mathbf{x}|_\perp$; it is hoped that this alternative description might help the reader's intuition.

We consider the set of all the column vectors $\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$ of $2n$ integers which make the n entries in $\mathbf{k}$ integral. It is easy to show that it forms a lattice since its discrete and closed under addition. This lattice has full dimension $2n$ (except in degenerate cases), and we can find the $2n$ basis vectors in two groups of n. In each group we combine the n column vectors into a matrix, and denote the resultant $n \times n$ matrices F G and K:

1. Find a basis for the homogeneous case in which $K = 0$. The resultant equation is $H * F = pG$, which can be solved by $F = pI$ and $G = H$ since $HpI = pH$.

2. Find a basis for the inhomogeneous case in which $K = I$. The resultant equation is $H * F = pG + qI$. To solve it, we assume that H is invertible modulo p, and find two integral matrices B, C satisfying $H * B = I + pC$ (that is, B is the inverse of H modulo p, and C is the matrix of multiples of the modulus p in the modular reductions.) Then $F = qB$ and $G = qC$ is a solution since $H * F = qH * B = qI + pqC$, and $pG + qI = pqC + qI$.

We now combine the two cases into a single $2n \times 2n$ matrix A whose columns generate the lattice of $\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$ vectors. The matrix A is:

$$A = \begin{bmatrix} pI & qB \\ H & qC \end{bmatrix}.$$

The small column vector we are looking for in this lattice has entries of zero and one in the top half, and around two or three in the bottom half. We believe that for the recommended parameters of the NTRU cryptosystem, the LLL algorithm will be able to find the original secret key $\mathbf{f}$ as the first half of such an unusually short lattice vector.

7 Extensions

We understand that the authors of NTRU, after learning the details of our attack, are continuing their research into related schemes [2].

One direction of their research involves schemes similar to NTRU but with larger parameters. The expense, for the designers of the system, comes with larger public keys and more time-consuming encryption. The added security comes from the notion that in a lattice of higher dimension (several hundred) it will be computationally harder for the opponent to find high-quality vectors. To maintain this security, one must keep ahead of advances in lattice basis reduction techniques.

Another direction of their research involves extensions to noncommutative groups. Instead of using a group algebra over $\mathbf{Z}_N$ (that is, the ring $\mathbf{Z}_q[X]/(X^N - 1)$), one would use a group algebra over a noncommutative group. At the time of this writing we have not had sufficient time to analyze these proposed extensions, but we hope to be able to comment on the noncommutative version in the final version of the paper.

8 Acknowledgments

We thank Claus Schnorr for insight into lattice basis reduction methods.

References

1. J. Hoffstein, J. Pipher and J. H. Silverman, "NTRU: A new high speed public key cryptosystem," Manuscript, August 30, 1996; presented at rump session of Crypto 96.
2. J. Hoffstein, J. Pipher and J. H. Silverman, private communications, October 1996 and January 1997.

3. A. K. Lenstra, H. W. Lenstra and L. Lovasz, "Factoring Polynomials with Integer Coefficients," Matematische Annalen **261** (1982), 513–534.
4. C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," Theoretical Computer Science **53** (1987), 201-224.
5. C. P. Schnorr, "Block reduced lattice bases and successive minima," Combinatorics, Probability and Computing **3** (1994), 507-522.