NTRU Cryptosystems Technical Report

Report # 012, Version 1
Title: Estimated Breaking Times for NTRU Lattices
Author: Joseph H. Silverman
Release Date: March 9, 1999

Abstract. In this note we report on experiments with the lattices underlying the NTRU Public Key Cryptosystem. We present data for the time needed to find a small vector and use this data to extrapolate expected breaking times for the NTRU PKCS for various parameter values. In particular, we find that NTRU 167, NTRU 263, and NTRU 503 are at least as secure as RSA 512, RSA 1024, and RSA 2048 respectively.

In this note we report on experiments with the lattices underlying the NTRU Public Key Cryptosystem. These experiments extend those described in [1]. We will concentrate entirely on the underlying lattices. For details of the NTRU public key cryptosystem, see [1].

§1. The Standard NTRU Lattice.

Fix integers N, d_f, and d_g. (See Table 1 below for typical values of these parameters.) Let S_d be the set of N-tuples with d coordinates equal to each of 1 and -1 and with the remaining $N-2d$ coordinates equal to 0. Similarly, let S'_d be the set of N-tuples with d coordinates equal to 1, with $d-1$ coordinates equal to -1, and with the remaining $N-2d+1$ coordinates equal to 0.

The *Standard NTRU Lattice* L^{NT} is the lattice of dimension $2N$ generated by the row vectors of a matrix of the following form, where $(h_0,\dots,h_{N-1})$ is a known list of integers:

$$L^{\mathrm{NT}} = \left(\begin{array}{cccc|cccc} \lambda & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & \lambda & \cdots & 0 & h_1 & h_2 & \cdots & h_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

The constant λ is a balancing constant which is chosen to maximize the efficiency of the search for small vectors in the lattice.

An NTRU private key consists of a pair of vectors

$$f = (f_0, f_1, \dots, f_{N-1}) \in S'_{d_f} \quad \text{and} \quad g = (g_0, g_1, \dots, g_{N-1}) \in S_{d_g},$$

and the attacker knows that the lattice contains the relatively short vector[†]

$$\mathbf{v} = (\lambda f_0, \ldots, \lambda f_{N-1}, g_0, \ldots, g_{N-1}).$$

The attacker knows the vector h, which is the NTRU public key, so he knows the lattice L^{NT}, and his goal is to recover the unknown vector $\mathbf{v}$, or any other vector of approximately the same length, from L^{NT}. (For example, L^{NT} contains N vectors of the same length as $\mathbf{v}$ obtained by cyclically rotating the coordinates of f and g.)

We consider the following constants associated to a lattice L.

$$\begin{aligned}
\sigma(L) &= \text{shortest expected (non-zero) length of vectors in } L,\\
\tau(L) &= \text{length of actual shortest vector in } L,\\
\alpha(L) &= \frac{\tau(L)}{\sigma(L)},\\
\gamma(L) &= \frac{\tau(L)}{\sigma(L)}\sqrt{\dim(L)}.
\end{aligned}$$

The attacker chooses the balancing constant λ to make α as small as possible, since lattice reduction methods work best when the target vector is as small as possible compared to the many vectors in L of length approximately σ. For a general lattice L, the Gaussian heuristic says that the length of the shortest non-zero vector satisfies

$$\sigma(L) \approx \sqrt{\frac{\dim(L)}{2\pi e}}\,\mathrm{Disc}(L)^{1/\dim(L)}.$$

For the standard NTRU lattice L^{NT}, these constants have the values

$$\begin{aligned}
\dim(L^{\mathrm{NT}}) &= 2N, \qquad \mathrm{Disc}(L^{\mathrm{NT}}) = (\lambda q)^N,\\
\sigma(L^{\mathrm{NT}}) &= \sqrt{\frac{\lambda N q}{\pi e}},\\
\tau(L^{\mathrm{NT}}) &= \sqrt{\lambda^2\|f\|^2 + \|g\|^2}.
\end{aligned}$$

Hence the optimal choice for the balancing constant is $\lambda = \|f\|/\|g\|$, which leads to the lattice constants:

$$\begin{aligned}
\alpha(L^{\mathrm{NT}}) &= \sqrt{\frac{2\pi e\|f\|\cdot\|g\|}{Nq}},\\
\gamma(L^{\mathrm{NT}}) &= \sqrt{\frac{4\pi e\|f\|\cdot\|g\|}{q}}.
\end{aligned}$$

[†] As we have formulated the problem here, the precise NTRU key is the vector $(f_0, f_{N-1}, \cdots, f_2, f_1)$, but this rearrangement of coordinates is irrelevant in studying lattice attacks.

N	$\gamma(L)$	q	d_f	d_g	d_ϕ
107	2.658	64	15	12	5
167	3.049	128	61	20	18
263	3.032	128	50	24	16
503	4.081	256	216	72	55

Table 1. Sample NTRU Parameter Sets

Experimental evidence described below suggests that if we hold $\gamma(L^{\rm NT})$ constant, then the log time to find the target vector (or a vector of approximately equivalent length) grows at least linearly with the dimension. In other words, if we let $T = T(L^{\rm NT})$ denote the amount of time it takes for LLL to find a target vector in the lattice $L^{\rm NT}$ of dimension $2N$, then

$$\log T \geq AN + B$$

for constants A and B.

§2. Experimental Results I: NTRU 167 and NTRU 263.

Four sets of NTRU parameters are given in Table 1. For these parameter sets, the value of the lattice constant

$$\gamma(L) = \frac{\tau(L)}{\sigma(L)}\sqrt{\dim(L)}$$

is equal to

$$\gamma(L^{\rm NT}) = \sqrt{\frac{4\pi e\|f\|\cdot\|g\|}{q}} = \begin{cases} 2.676 & \text{if } N = 107, \\ 3.053 & \text{if } N = 167, \\ 3.040 & \text{if } N = 263, \\ 8.801 & \text{if } N = 503. \end{cases}$$

All of the experiments in this note were run on 400 MHz Celeron machines running the Linux operating system. The software used was Victor Shoup's implementation of the LLL algorithm with improvements due to Schnorr, Euchner and Hoerner. Shoup's NTL package is available at [2]. Most of the general remarks in [1, Appendix] concerning lattice reduction algorithms apply to the experiments in this note. In particular, we set Schnorr's pruning constant to equal 0, because we did not find that setting it to be positive improved the running time. We also set the LLL constant $\delta = 0.99$, and we ran the program using increasing block sizes until it found the target vector (or a vector slightly longer than the target vector).

The experiments described in this note confirm the observation made in [1] that (at least for the NTRU lattices) the algorithm generally either finds a vector

of the exact correct length, or it finds one that is considerably too long to be useful for decryption. Thus the idea of Coppersmith and Shamir [3] to use vectors a little longer than the target vector to attack NTRU, while very interesting as a theoretical remark, does not appear to be of practical significance. In practice, LLL generally seems to terminate with a q-vector (i.e., a vector with one coordinate equal to q and the rest 0) until a sufficiently large block size is used, at which point it finds the target vector. The necessary block size increases with the dimension, and as one knows, the running time of LLL increases exponentially with the block size.

In this section we concentrate our attention on NTRU lattices with $q = 128$ and with lattice constant

$$\gamma(L^{\mathrm{NT}}) \approx 3.05,$$

since this covers the two sets of NTRU parameters NTRU 167 and NTRU 263. The results of our experiments are given in Table 2. The first column gives the NTRU parameter N, which we recall means that the NTRU lattice L^{NT} has dimension $2N$. The third column gives the time needed (in seconds) to find a target vector for one or more experiments at the given dimension. The time listed is the time required on the final run; that is, using the block size which actually found a target vector. The middle column gives the average amount of time needed to find a target vector for the given value of N.

N	T_{avg}	Experimental Times T
68	691	691
70	889	764, 934, 969
72	1082	1029, 1135
74	1222	1150, 1294
76	1208	1208
78	1776	1487, 1741, 2101
80	2390	2390
82	5296	3656, 4656, 6087, 6785
84	9831	9808, 9853
86	12505	7325, 17684
88	39652	15605, 40882, 62468
90	54453	30466, 78440

Table 2. Breaking Time (secs) for Lattices with $q = 128$ and $\gamma \approx 3.05$

The graph of $\log(T_{\mathrm{avg}})$ versus N shows that the growth is reasonably linear, although there is a noticable upward concavity. This upward concavity will only

help us as we linearly extrapolate breaking times at higher dimension, so we compute the linear regression line for the average times in Table 2:

$$\log(T) \approx 0.2002 \cdot N - 7.608.$$

Using this formula to estimate the time needed to find a target vector for NTRU 167 and NTRU 263 gives the values in Table 3.

N	T (seconds)	T (MIPS-years)
167	$1.638 \cdot 10^{11}$	$2.077 \cdot 10^{6}$
263	$3.634 \cdot 10^{19}$	$4.607 \cdot 10^{14}$

Table 3. Estimated Breaking Times for NTRU 167 and NTRU 263

Note that all of the timing figures in Table 2 are in seconds. Since the experiments were run on 400 MHz Celeron machines, we have converted the time in seconds to the time in MIPS-years by first multipying by 400 (to account for the 400 MHz machines) and then dividing by 31557600, which is the number of seconds in a year.

As mentioned above, the extrapolated breaking times given in Table 3 are very conservative for the following reason. Examining the graph of the data in Table 2 (i.e., the graph of $\log(T)$ versus N), there is a clear upward concavity to the plotted points. For example, if we remove the first three points, that is remove the points with $N = 68$, 70, and 72, then the linear regression line for the remaining points is

$$\log(T) = 0.2582 \cdot N - 12.484.$$

This increased slope then leads to an extrapolated breaking time of $T = 2.537 \cdot 10^8$ MIPS-years for NTRU 167 and an extrapolated breaking time of $T = 1.470 \cdot 10^{19}$ MIPS-years for NTRU 263.

We are currently conducting experiments for NTRU 503, which has $q = 256$ and lattice constant $\gamma(L^{\mathrm{NT}}) = 8.8$. However, even our preliminary data makes it clear that the regression line for NTRU 503 will be considerably steeper than the regression line for the lower values of N. So if we use the data from Table 2 to extrapolate the breaking time for NTRU 503, the resulting time is almost certainly much lower than the truth; yet even this conservative estimate says that it would take about $2.663 \cdot 10^{40}$ seconds (which equals $3.375 \cdot 10^{35}$ MIPS-years) to break NTRU 503.

§3. Experimental Results II: NTRU 107.

In the original formulation of the NTRU public key cryptosystem, it was suggested that one could use $N = 107$ with quite small values of q, d_f and d_g, to create a cryptosystem with at least moderate security. Our experiments have shown that such a system can probably be broken in between 12 and 24 hours on a single 400 MHz machine, and an idea of Alexander May [4] to guess a lattice of slightly lower dimension might cut these times in half. (See also [5] for a detailed discussion of May's ideas, generalizations, and an analysis of their effect on the security of the NTRU public key cryptosystem.)

We thus do not recommend NTRU 107 for most practical applications, although its blinding speed could make it useful in some situations where speed is essential and each message has only a very small intrinsic value. In any case, it is still interesting to study the effectiveness of lattice reduction methods on smaller lattices of this sort, since it helps to verify the general pattern of breaking time versus dimension for NTRU lattices. In Table 4 we give the results of our experiments. The machines and algorithms used are as described in Section 2.

N	T	N	T	N	T
50	58	68	413	86	4766
52	76	70	526	88	3931
54	80	72	712	90	4306
56	104	74	735	92	14533
58	147	76	723	94	12117
60	192	78	1244	96	15692
62	224	80	1671	98	106683
64	251	82	2114	100	196083
66	294	84	2720	102	19674

Table 4. Breaking Time (secs) for Lattices with $q = 64$ and $\gamma \approx 2.676$

The linear regression line for the data in Table 4 is

$$\log(T) \approx 0.1339N - 2.9983,$$

and this leads to an estimated breaking time for NTRU 107 of

$$T(\text{NTRU 107}) \approx 8.34 \cdot 10^4 \text{ seconds} = 1.06 \text{ MIPS-years}.$$

An examination of Table 4 shows that the times for $N = 98$ and $N = 100$ may possibly be anomolous. To examine the robustness of our data, we recomputed the

regression line and estimated breaking times with the $N = 98$ and $N = 100$ data points removed. The results were

$$\log(T) \approx 0.1190N - 2.0182$$

with an estimated breaking time of

$$T(\text{NTRU 107}) \approx 4.48 \cdot 10^4 \text{ seconds} = 0.57 \text{ MIPS-years.}$$

Thus even removing these seemingly high values of T only cuts the estimated breaking time in half. We also note that the data in Table 4 exhibits the same small upward concavity already observed in Table 2.

References

[1] J. Hoffstein, J. Pipher, J.H. Silverman, NTRU: A new high speed public key cryptosystem, Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, Lecture Notes in Computer Science 1423, J.P. Buhler (ed.), Springer-Verlag, Berlin, 1998, 267–288

[2] NTL — A Number Theory Library, Victor Shoup, available at `http://www.cs.wisc.edu/~shoup/ntl/`

[3] D. Coppersmith, A. Shamir, Lattice attacks on NTRU, in W. Fumy, ed., *Proceedings fo EUROCRYPT 97*, Lecture Notes in Mathematics **1233**, Springer, 1997, 52–61

[4] A. May, Cryptanalysis of NTRU, preprint, February 1999

[5] J.H. Silverman, Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem, NTRU Cryptosystems Technical Report 013, March 2, 1999. ⟨`www.ntru.com`⟩

Comments and questions concerning this technical report should be addressed to

`techsupport@ntru.com`

Additional information concerning NTRU Cryptosystems and the NTRU Public Key Cryptosystem are available at

`www.ntru.com`
